

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВОЛИНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ЛЕСІ УКРАЇНКИ**

Кафедра комп'ютерних наук та кібербезпеки

На правах рукопису

САХНЮК ВАСИЛЬ ЄВГЕНОВИЧ

РОЗРОБКА ЕЛЕКТРОННОГО КАБІНЕТУ ВИКЛАДАЧА

Спеціальність: 122 Комп'ютерні науки

Освітньо-професійна програма

«Комп'ютерні науки та інформаційні технології»

Робота на здобуття освітнього ступеня «Магістр»

Науковий керівник:

Головін Микола Борисович,

кандидат фізико-математичних наук,

доцент кафедри комп'ютерних наук

та кібербезпеки

РЕКОМЕНДОВАНО ДО ЗАХИСТУ

Протокол № _____

засідання кафедри комп'ютерних наук та кібербезпеки

від _____ 20__ р.

Завідувач кафедри

(_____) _____

(підпис)

ПІБ

ЛУЦЬК–2024

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1 ОСОБЛИВОСТІ АВТОМАТИЗАЦІЇ РОБОТИ ПЕДАГОГА З ВИКОРИСТАННЯМ АВТОМАТИЗОВАНОГО РОБОЧОГО МІСЦЯ	6
1.1 Робочий процес викладача та його особливості	6
1.2 Автоматизоване робоче місце як основа ефективної роботи педагога	13
1.3 Типові елементи онлайн робочого місця викладача	20
1.4. Огляд та порівняльний аналіз існуючих розробок для онлайн робочого місця викладача	27
РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ОНЛАЙН РОБОЧОГО МІСЦЯ ВИКЛАДАЧА	34
2.1 Постановка задачі, призначення та вимоги до програмного засобу «ARMlect»	34
2.2 Вибір моделі розробки програмного засобу «ARMlect»	39
2.3 Опис проекту автоматизованого робочого місця «ARMlect»	42
2.4 Обґрунтування вибору інструментальних засобів розробки програмного засобу «ARMlect»	47
2.5 Основні режими функціонування програмного засобу «ARMlect»	56
2.6 Організація тестування та налагодження програмного засобу «ARMlect»	61
2.7 Рекомендації по використанню та впровадженню програмного засобу «ARMlect»	64
ВИСНОВКИ	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	70
ДОДАТКИ	74

ВСТУП

Актуальність теми. Сучасний навчальний процес у закладах вищої освіти складний та багатогранний. У нього, зокрема, входять: власне процес навчання, комунікація між суб'єктами цього процесу, система оцінювання, облік відвідуваності, навчальні матеріали та їх поширення, створення та використання конспектів занять, форм контролю знань, лекцій, практичних занять, планування різних видів навчальної діяльності та багато інших пунктів. Водночас активно розвиваються інформаційні технології, зокрема, мережеві та вебтехнології, зростає масштаб їх використання для оптимізації та автоматизації робочого процесу різних галузей. Рішення та розробки на основі таких технологій широкого використовуються державними та приватними установами, які прагнуть забезпечувати стабільно високу та конкурентну якість послуг й продуктів. Такі тенденції дійсні і для сфери освіти. Освітяни приватних та державних навчальних закладів здійснюють великі обсяги роботи, а технічно-інформаційні засоби покликані спростити їх робочих процес та підвищити ефективність праці. Значний інтерес науковців викликає вивчення проблеми створення та оптимізації віртуального освітнього простору і, зокрема, робочих місць освітян, чому присвячено значну кількість робіт, в тому числі праць вітчизняних науковців. Зокрема, актуальні роботи належать таким авторам як Кантар І. Л. [12], Ганашок А. І. [31], Гроза П. М. [15] та ін. [1, 3, 4–7, 10–11, 15, 17–18], у яких висвітлено різносторонні погляди на автоматизацію робочих місць у сфері освіти.

Дослідники пропонують різні підходи до вирішення даної проблеми. Одним із варіантів є використання універсальних віртуальних робочих місць, запозичених з інших сфер, у освітній галузі [4, 12]. Популярними є рішення на основі табличних та текстових редакторів із офісних пакетів, зокрема Microsoft Office, Libre Office та Google Cloud. Такі рішення часто використовуються у школах, але можуть бути адаптовані і для роботи викладачів у вищих навчальних закладах [1, 31]. Крім того існують і більш традиційні погляди, за якими застосунки для викладачів не повинні орієнтуватися на мережу Інтернет, а працювати виключно локально [4–6, 15].

На нашу думку, така ширина існуючих підходів та різноплановість застосування інформаційних технологій свідчить про актуальність та неповне вирішення даної проблеми. У свою чергу ми вважаємо, що необхідно робити акцент на підходах з використанням мережі Інтернет для забезпечення максимальної гнучкості робочого місця. Такий підхід відповідає сучасним вимогам до навчального процесу, а зокрема його мобільності та можливості проведення у дистанційному форматі, коли у викладача нема доступу до його стаціонарного робочого місця у закладі вищої освіти. Вебресурс повинен бути адаптованим для різних носіїв даних, зокрема стаціонарних комп'ютерів, ноутбуків, планшетів та мобільних телефонів, підключених до мережі Інтернет. Доступ до ресурсу повинен бути захищеним, а вхід у систему можливим лише після проходження ідентифікації. Щодо структури та функціоналу системи, то вони повинні бути такими, що стандартизують та спрощують рутинні обов'язки викладача.

Університети у наш час ще не повністю забезпечені простими та ефективними технологіями, які оптимізують ці процеси, тому проблема досі не вирішена остаточно.

Мета роботи. Метою роботи є реалізація онлайн робочого місця викладача.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- описати особливості робочого процесу викладача на основі актуальних теоретичних досліджень;
- проаналізувати поточні підходи до вирішення проблеми реалізації онлайн робочого місця викладача;
- здійснити огляд та аналіз аналогічних програмних розробок;
- сформулювати вимоги до програмного продукту;
- обрати модель розробки та визначити структуру програмного засобу;
- обрати середовище та технології розробки;
- розробити онлайн робоче місце викладача за допомогою обраних інструментальних засобів;
- виконати тестування розробленого програмного засобу;

- сформулювати вимоги до впровадження та використання програмного засобу.

Об'єкт дослідження – автоматизоване робоче місце викладача і технології його розробки.

Предмет дослідження – процес проектування та розробки автоматизованого робочого місця викладача з використанням вебтехнологій.

Практичне значення одержаних результатів полягає у реалізації багатофункціонального та простого у використанні робочого місця викладача, з можливістю використання у мережі Інтернет. Створений програмний продукт може бути використаний у закладах вищої освіти, які ще не мають ефективних механізмів для автоматизації роботи викладача. Матеріали магістерського дослідження можуть слугувати відправною точкою для подальших теоретичних досліджень та технологічних розробок.

РОЗДІЛ 1

ОСОБЛИВОСТІ АВТОМАТИЗАЦІЇ РОБОТИ ПЕДАГОГА З ВИКОРИСТАННЯМ АВТОМАТИЗОВАНОГО РОБОЧОГО МІСЦЯ

1.1 Робочий процес викладача та його особливості

Викладання – діяльність викладача по вибору необхідних суб'єкту навчання елементів об'єктивного досвіду людства і управління процесом їх засвоєння суб'єктами навчання в навчанні. Якість освіти – поняття складне і його рівень безпосередньо пов'язаний з якістю діяльності викладача, яка сьогодні повинна відповідати багатьом вимогам. Одним з елементів, що забезпечують якість освіти використання інформаційних технологій, які поступово та послідовно впроваджуються в навчальний процес задля його спрощення та автоматизації. Це необхідно, оскільки навчальний процес складається з багатьох видів діяльності, які не завжди легко здійснювати та контролювати без використання таких технологій.

Зокрема, викладачеві ВНЗ необхідно здійснювати облік успішності, відвідуваності, дотримуватися термінів здачі робіт, формувати нові екзаменаційні квитки і питання. Важливою проблемою є також те, що всі відомості про оцінки і відвідуваності студентів містяться в індивідуальному журналі викладача або на аркуші і, крім того, такі дані часто зберігаються з використанням системи позначень повністю зрозумілих тільки самому викладачу – це значно ускладнює роботу та комунікацію між учасниками навчального процесу. Для збільшення ефективності необхідне використання централізованого сховища для такої інформації. Це дозволяє краще організувати елементи, необхідні для забезпечення навчального процесу.

Успіх будь-якої діяльності багато в чому залежить від уміння її організувати, тим більше за умови реалізації кількох видів діяльності. Праця викладача ускладнюється ще й тим, що треба вміти організувати не тільки свою працю, а й роботу студентів. Змістова сторона освітнього процесу – важлива його складова, але без ефективної організації забезпечити ефективність роботи педагога неможливо.

Структуру педагогічної діяльності вивчали А. І. Артюхіна [26], Н. М. Мирончук [22, 28], Резнік С. Д. [25]. Дослідження А. І. Артюхіною педагогічної діяльності як складної динамічної системи, що має свою специфічну структуру, дозволили їй виділити основні її елементи [26, с. 7]:

- конструктивний;
- організаторський;
- комунікативний;
- інформаційний;
- розвивальний;
- орієнтаційний;
- мобілізаційний.

Докладно характеризує структуру педагогічної діяльності І.Ф. Харламов. Він підкреслює, що для надбання і вдосконалення професійної вмілості і вироблення педагогічного майстерності педагога необхідно глибоко розуміти структуру педагогічної діяльності і пов'язану з нею систему теоретичних знань, практичних умінь і навичок [23]. На підставі психолого-педагогічних досліджень він виділяє наступні взаємопов'язані види педагогічної діяльності :

- діагностична діяльність (пов'язана з вивченням учнів і встановленням рівня їх успішності, особистісних характеристик);
- орієнтаційно-прогностична діяльність (виражається в умінні вчителя визначати лінію навчально-виховної діяльності, її конкретні цілі і завдання на кожному етапі роботи вчителя, прогнозувати її результати);
- конструктивно-проектувальна діяльність (органічно пов'язана з орієнтаційно-прогностичною діяльністю: якщо вчитель прогнозує той чи інший результат в своїй роботі, йому необхідно вміти конструювати, проектувати її зміст, спланувати конкретну навчальну і виховну діяльність);
- організаторська діяльність (пов'язана з залученням здобувачів освіти в планову навчально-виховну роботу та стимулюванням активності в її процесі);
- інформаційно-пояснювальна діяльність (виконуючи цей вид діяльності, викладач виступає не тільки як організатор навчально-виховного процесу,

а й як джерело наукової, світоглядної і морально-естетичної інформації. Від того, як сам педагог володіє навчальним матеріалом, залежить якість його пояснення, його змістовність, логічна стрункість, насиченість яскравими деталями і фактами. Варто зазначити, що не всі викладачі на достатньому рівні володіють цим видом діяльності внаслідок недостатнього рівня знання свого предмета, що негативно позначається на навчанні і вихованні студентів в цілому);

- комунікативно-стимулююча діяльність (пов'язана з тим впливом викладача, який ґрунтується на його особистісній харизмі, моральній культурі, вмінні організувати істинно педагогічне спілкування зі здобувачами освіти, спонукати їх своїм позитивним прикладом до активної різноманітної діяльності з оволодіння суспільним досвідом. Цей вид діяльності ґрунтується на любові до студентів, душевному відношенні до них, гуманістичній спрямованості особистості педагога);
- аналітично-оцінна діяльність (проявляється в умінні викладача аналізувати хід навчального процесу, виявляти в ньому позитивні сторони, бачити свої і чужі прорахунки і недоліки в організації роботи зі студентами, аналізувати чи досягаються результати діяльності по навчанню та соціальному розвитку, зіставляти їх з результатами роботи колег. Цей вид педагогічної діяльності сприяє пошукам шляхів удосконалення навчально-виховної роботи викладача і його професійного росту);

Така структура педагогічної діяльності, сутність і система умінь і навичок вчителя, оволодіння якими лежить в основі становлення і розвитку його професійної майстерності [33].

В процесі навчання викладач вирішує завдання навчання і розвитку студентів, що і визначає основні функції його діяльності:

- отримання і накопичення нових знань як в області предмета викладання, так і форм, прийомів і методів викладання (реалізація цієї функції передбачає наявність здатності опановувати спеціальні знання, знання з педагогіки, психології і т.д., а також творчо використовувати накопичені знання. Варто зазначити, що рівень спеціальних знань викладача вузу

значно вищий, ніж рівень знань методики викладання, і це велика проблема вищої школи);

- проектування процесу навчання (виходячи з цілей вивчення дисципліни, викладач розробляє програму, структуру курсу, вибирає методику і технологію викладання. Уміння планувати вивчення цілого курсу і кожне окреме заняття, співвідносити зміст окремих тем з загальними цілями навчальної дисципліни, знання вікових та індивідуально-психологічних особливостей учнів – необхідні складові діяльності викладача. При проектуванні навчального процесу педагог творчо підходить до використання вже розроблених технологій, методів, прийомів навчання, а також пропонує власні оригінальні підходи);
- відбір і структурування змісту навчального заняття (виходячи зі знання особливостей аудиторії, власної підготовки, викладач конструює кожне заняття. Відзначимо, що одна тема може бути подана різними групами студентів по-різному: відібраний інший матеріал, інакше розподілено час на вивчення окремих питань, використані інші методи і прийоми викладання);
- організація навчально-виховного процесу (мова йде не тільки про організацію діяльності студентів, а й про вміння викладача організувати себе, свою роботу з досягнення цілей навчання);
- встановлення комунікативних зв'язків (без таких зв'язків неможливий особистісний розвиток. Важливе значення в професійній діяльності викладача мають його взаємовідносини з колегами, зі студентами, тому що вони стають джерелом і каталізатором процесу збагачення знаннями, умовою вдосконалення професіоналізму, засобом виховного впливу на студентів. Ефективність комунікативних зв'язків обумовлюється рядом позицій: об'єктивне сприйняття людини в процесі спілкування з нею, взаємна довіра під час спільної діяльності, взаємообмін інформацією, уникнення конфліктів, справедлива, але тактовна критика, зміна, за необхідності, власної поведінки по відношенню до партнера.

В поточних час особливої важливості набуває і дистанційна освіта. Парадигма освіти змінюється. Тепер важливим аспектом є самостійне набуття

знань. Логіка освіти в таких випадках спрямована на самостійну роботу особистості, де особистість переходить на новий рівень творчого розвитку. Формування парадигми освіти з орієнтацією на критерії Болонського процесу, висувають нові вимоги до якості освітньої системи, розвитку професійної компетентності особистості [10]. Компетентність визначається як володіння людиною відповідною компетенцією, що включає її особистісне ставлення до неї і предмету діяльності, розуміння відповідальності за свої дії.

Завдання підвищення якості робочого процесу викладача пов'язане з розширенням кордонів освітнього простору, забезпеченням вільного доступу до знань. Додання системі освіти якостей відкритої системи створює кардинальну зміну її властивостей і властивостей її компонентів [1, с. 3]. Зокрема, нова роль викладача в сучасних умовах обумовлюється покладанням на нього функцій координування пізнавального процесу, коригування змісту дисципліни, консультування при складанні індивідуального навчального плану, складання нових екзаменаційних квитків для контролю отриманих знань студентом, керування навчальними проектами за допомогою комп'ютерних і мережевих технологій і т.д. При цьому, підвищення інтенсивності праці викладача ВНЗ, особливо викладача інформаційних технологій, обумовлюється також необхідністю забезпечення навчального процесу певної якості в заданий час в умовах постійно зростаючого обсягу інформації. Для вирішення цих завдань створюються комп'ютерні засоби підтримки навчального процесу відомі як «автоматизовані робочі місця», призначені для спрощення роботи. Вони активно створюються і впроваджуються на виробництві, в освіті і т.д. [2].

Важливо зазначити, що діяльність викладача вищого навчального закладу є системною та характеризується динамічністю, креативністю, інтелектуальною й емоційною напруженістю, наявністю як планових так і спонтанних елементів. Як наслідок викладач ВНЗ повинен володіти високою здатністю до самоорганізації та планування професійного робочого процесу.

Важливим елементом організації праці викладача ВНЗ є планування робочого процесу як механізм запобігання дефіциту робочого часу та спосіб досягнення продуктивності діяльності.

Організаторські здібності проявляються в умінні людини налагоджувати успішну групову діяльність над розв'язанням спільних проблем та завдань, власній ефективній діяльності, що досягається вмінням планувати дії, раціонально розподіляти час, зусилля та інші ресурси.

Робочий час викладача формується на основі його навчальних, методичних, наукових і організаційних обов'язків у поточному навчальному році, відображених в індивідуальному плані викладача. Обсяг робочого часу викладача з повними обов'язками не перевищує 1548 годин на один академічний навчальний рік (відповідно – 36 годин на тиждень) [22]. Кількість занять, доручених для проведення певному викладачеві, виміряна в академічних годинах, відповідає навчальному навантаженню викладача. Мінімальна й максимальна величина навчального навантаження встановлюється наказом в межах конкретного університету. При додатковій потребі у ВНЗ III і IV рівня акредитації викладач може бути залучений до організації занять понад визначений обов'язковий обсяг навчального навантаження, сформований індивідуальним робочим планом, у межах свого робочого часу. Графік робочого часу викладача узгоджується з розкладом аудиторних навчальних занять, консультацій та іншими видами робіт, що передбачені індивідуальним робочим планом викладача. Викладач повинен слідувати встановленому йому графіку робочого часу.

Важливим елементом для забезпечення якості роботи викладача, є глибоке розуміння процесу навчання та його складових. Процес навчання відповідно є формою пізнання об'єктивної дійсності, оволодіння досвідом людства, взаємодією викладача й студента. Поняття “навчальний процес” містить всебічні елементи навчання: викладач, форми, методи та засоби навчання, які він використовує; студент, що працює під керівництвом викладача на занятті та самостійно вдома; забезпечення навчального процесу наочністю й необхідними технічними засобами. Структурний процес навчання формують деякі елементи:

- цільовий (формування мети та цілей вивчення матеріалу на заняттях, навчального предмета та освітньої мети навчального закладу);
- стимулююче-мотиваційний (спонукання здобувачів освіти до активної пізнавальної діяльності, формування належної мотивації);

- змістовий (ефективний та раціональний підбір предметів навчальної програми, змістовність конкретних навчальних програм і підручників, а також оптимальність змісту кожного заняття);
- операційно-дієвий (підбір методів, форм, прийомів і засобів навчання);
- контроль-регулюючий (контроль засвоєння знань, сформованості умінь та навичок студентів);
- оцінно-результативний (оцінка володіння матеріалом, умінь і навичок кожного студента, потенційних та реальних причин неуспішності і пошук шляхів їх вирішення або попередження).

Втілення кожного елемента та їх поєднань спирається на свідому, цілеспрямовану спільну роботу викладача та студента [28].

Для реалізації такого структурного процесу використовується широкий набір освітніх технологій. Освітні технології – систематичний метод планування, застосування та оцінювання всього процесу навчання і засвоєння знань шляхом обліку людських і технічних ресурсів і взаємодії між ними для досягнення більш ефективної форми освіти. Вони реалізують досягнення з області досліджень теорії і практики в межах системи освіти, що має зв'язки з усіма сторонами організації педагогічної системи для досягнення специфічних і потенційно-відтворюваних педагогічних результатів [33]. Також вони пов'язані з такими поняттями як наукове планування, організація, оцінювання та корекція педагогічного процесу з метою підвищення його ефективності, що дозволяє отримати необхідний гарантований результат [31, с. 8].

Для забезпечення якісного навчального процесу викладач також повинен використовувати інтерактивне навчання – спеціальну форму організації пізнавальної діяльності, спосіб пізнання, здійснюваний у формі спільної діяльності студентів, при якій всі учасники взаємодіють один з одним, обмінюються інформацією, спільно вирішують проблеми, моделюють ситуації, оцінюють дії інших і свою власну поведінку, занурюються в реальну атмосферу ділового співробітництва для розв'язання проблеми.

Процес навчання потребує якісної формалізації та автоматизації, використання різних технологій навчання. Технологія навчання – вибудована на

основі закономірностей логічна послідовність операцій, що відображає шлях гарантованого досягнення дидактичної мети.

Професійний викладач повинен не лише ефективно працювати з поточними завданнями, а й планувати подальшу діяльність, реалізовувати елементи педагогічного проектування. Педагогічне проектування – це попередня розробка основних деталей майбутньої діяльності учнів і педагогів. Педагогічне проектування полягає в тому, щоб створювати приблизні варіанти майбутньої діяльності і прогнозувати її результати. Об'єктами педагогічного проектування можуть бути: педагогічні системи, педагогічний процес, педагогічні ситуації. Формами педагогічного проектування є документи, в яких описується з різним ступенем точності створення і дію педагогічних систем, процесів і ситуацій. Він, за потреби, реалізовує практико-орієнтовану діяльність, метою якої є розробка нових, неіснуючих в практиці освітніх систем і видів педагогічної діяльності (наприклад, предметом проектування можуть стати зразки майбутніх програм, підручників і т.д.). При цьому необхідно дотримуватись різносторонніх принципів педагогічного проектування, зокрема: принципу людських пріоритетів (принцип орієнтації на людину) та принципу саморозвитку проєктованих систем, процесів, ситуацій (динамічність, гнучкість, здатність до перебудови, ускладнення, спрощення).

Після виконання вищезгаданих функцій та реалізації відповідних їм компонентів діяльності, викладач також здійснює контроль в педагогічному процесі, процедуру оціночної діяльності, що включає в себе дії з використанням різноманітних педагогічних заходів і методів вимірювань для отримання інформації про хід та результати навчання [26, с. 17].

1.2 Автоматизоване робоче місце як основа ефективної роботи педагога

В процесі аналізу поняття «автоматизовані робочі місця», фахівці, як правило, позначають їх як професійно-орієнтовані невеликі обчислювальні системи, розташовані безпосередньо на робочих місцях фахівців і призначені для автоматизації їх роботи [15].

Перш за все варто зазначити, що автоматизоване робоче місце входить, як програмно-технічний елемент, до автоматизованої системи і поза даною конкретною системою не існувати не може.

Автоматизовані робочі місця в автоматизованій системі можуть бути різних видів, в залежності від специфіки праці та професії відповідного спеціаліста (АРМ оператора-технолога, АРМ проектувальника, АРМ викладача, АРМ лікаря та ін.)

Кожен вид АРМ призначається для одного виду користувачів АС. Якщо, наприклад, на кафедрі один завідувач, але чотири викладачі, тоді в АС кафедри буде два види АРМ: один для завідувача, другий для викладача. Проте фізично на кафедрі буде використовуватись п'ять елементів АРМ.

АРМ, складова автоматизованої системи складається, як правило, з трьох складових: програмної, технічної та інформаційної.

Програмна частина АРМ формується на основі завдань АС та виду діяльності на відповідному робочому місці [32].

Важливо розуміти, що персонал є частиною АС, проте, з іншого боку, не є частиною АРМ. Кожне АРМ охоплює тільки програмне, інформаційне та технічне забезпечення [8, с. 3]. Оскільки кожна автоматизована система певним чином прив'язана до організації, для якої її було створено, то жодне АРМ не можна механічно відділити з цієї системи і аналогічним чином перенести в іншу організацію. АРМ існує тільки всередині АС. Іншими словами, персональний комп'ютер можна придбати та встановити як серійний виріб; з іншої сторони, серійних АРМ не буває, відповідно не буває серійних АС. Кожна з таких систем повинна бути адаптована програмним чином до особливостей роботи організації, наприклад закладу освіти [14].

Для використання у освітній сфері АРМ має також володіти деякими характеристиками експертних систем для автоматизації процесу аналізу даних та прийняття певних рішень. Такі системи включають знання про деяку частково формалізовану предметну область, вони здатні пропонувати та пояснювати користувачеві вирішення завдань в межах цієї області, а також емулювати здатність людини (експерта) міркувати і приймати рішення. Експертні системи призначені для вирішення складних завдань на основі міркувань з

використанням знань, представлених, головним чином, у вигляді умовних правил та операторів, а не за допомогою звичайного процедурного коду. Такий модуль часто входить до складу АРМ. Експертна система складається з наступних елементів:

- база знань – семантична модель, що описує предметну область і дозволяє відповідати на такі питання з неї, відповіді на які в явному вигляді відсутні в базі. Вона є основним компонентом інтелектуальних та експертних систем та дозволяє зробити механізм роботи з даними ефективнішим;
- машина логічного висновку – система, що моделює механізм міркувань і оперує знаннями і даними з метою отримання нових відомостей з знань і інших даних. Зазвичай така машина використовує програмно реалізований механізм дедуктивного логічного висновку або механізм пошуку рішення у внутрішній мережі;
- підсистема пояснень – система, що дозволяє користувачеві отримувати відповіді на питання: «Як було отримано те чи інше рішення?»;
- підсистема спілкування – забезпечує ведення діалогу з користувачем, в ході якого система запитує в нього необхідні факти для процесу міркування. Також підсистема спілкування надає можливість людині в певній мірі контролювати і коригувати хід міркувань експертної системи.

Підхід з використанням таких підсистем може допомогти викладачеві при вирішенні завдань, важких для швидкого та точного аналізу експертом-людиною, та отримати результати, що не поступаються за якістю і ефективністю рішенням, що отримуються таким експертом. Такі підходи часто позначають також терміном «інженерія знань», введеним Е. Фейгенбаумом.

Програмні засоби, що базуються на технології експертних систем, або інженерії знань, набули значного поширення в світі. Важливість експертних систем полягає в наступному:

- технологія експертних систем істотно розширює коло практично значущих завдань, що вирішуються на комп'ютерах, рішення яких приносить значний економічний ефект;

- технологія ЕС є найважливішим засобом у вирішенні глобальних проблем традиційного програмування: тривалість і, отже, висока вартість розробки складних додатків;
- висока вартість супроводу складних систем, яка часто в кілька разів перевершує вартість їх розробки; низький рівень повторної використовуваного програм і т.д.
- об'єднання технології ЕС з технологією традиційного програмування додає нові якості до програмних продуктів за рахунок: забезпечення динамічної модифікації додатків користувачем, а не програмістом; більшої "прозорості" додатка (наприклад, знання зберігаються на обмеженому просторі, що не вимагає коментарів до знань, спрощує навчання і супровід); кращої графіки; інтерфейсу і взаємодії [30].

Крім реалізації деяких характерних елементів експертних систем, створення і використання АРМ повинно ґрунтуватися на ряді загальних принципів проектування систем обробки даних.

Обґрунтування конкретних характеристик всіх засобів АРМ відбувається на етапі параметризації. Параметризація полягає у виділенні і вивченні властивостей технічних, програмних, інших формаційних засобів, які відповідають вимогам і обмеженням на різних етапах структуризації АРМ. Для безперебійної роботи АРМ використовуються спеціальне методичне, інформаційне, математичне, програмне, лінгвістичне, технологічне, організаційне, ергономічне і правове забезпечення [8].

Інформаційне забезпечення включає в себе організацію інформаційної бази. Розробники визначають зв'язки між інформаційними потоками, формують пакет вхідної документації в електронному вигляді, визначають використання нормативно-довідкової документації і рівні доступу до інформації користувачів різних категорій залежно від суті вирішуваних ними завдань. Користувачі поділяються за службовим становищем, за спеціальностями, за рівнем підготовленості і частоті роботи з обчислювальною технікою, за рівнем доступу до даних.

Математичне забезпечення АРМ становить сукупність алгоритмів, які забезпечують введення, контроль, зберігання і обробку інформації, формування

її у вигляді графіків, таблиць і діаграм, а також забезпечення достовірності і захисту інформації. Математичне забезпечення служить основою для розробки комплексу програмних засобів і має узгоджуватися з потенційними вимогами користувача АРМ.

Програмне забезпечення АРМ визначає інтелектуальні і професійні можливості користувача, широту і повноту виконуваних функцій, можливість використання додаткових пристроїв [9]. Прикладне програмне забезпечення зазвичай налаштоване на роботу за заданим алгоритмом з інформацією з певної предметної області. Тому введення та обробка даних супроводжуються суворим контролем, що значно знижує ймовірність помилок і приводить до підвищення ефективності роботи користувача. В організації АРМ, програмне забезпечення повинно використовувати зручний, призначений для користувача, інтерфейс і включати в себе алгоритми для вирішення задач, необхідних користувачеві. Видача результатів здійснюється в зручному наочному вигляді (таблиці, графіки, діаграми). Програмне забезпечення надає можливість оперативної роботи користувача з БД і можливість спільного використання ресурсів в мережевих технологіях.

Лінгвістичне забезпечення АРМ включає в себе мови спілкування технології АРМ з користувачами (російська, англійська), мови-запити (SQL), інформаційно-правові мови і мови-посередники в мережах. Мовні засоби АРМ необхідні для однозначного змістового співвідношення дій користувача і ПЕОМ. Чим вище інтелектуальність АРМ, тим більше можливостей необхідно передбачати в мовних засобах АРМ. Мови АРМ повинні бути орієнтовані на користувача і професійні особливості АРМ.

Технологічне забезпечення АРМ являє собою встановлену сумісність проектних рішень, що визначають послідовність операції, процедур та етапів в реалізації завдань за допомогою АРМ. Можна ви ділити системне і прикладне технологічне забезпечення АРМ, яке включає в себе основні операції, які реалізуються інформаційною технологією:

- введення інформації з паперових документів за допомогою клавіатури;
- контроль введеної інформації за допомогою засобів контролю і візуально;

- введення та контроль даних з магнітних носіїв, набраних на інших ПЕОМ;
- прийом даних у вигляді повідомлень по каталогах зв'язку з використанням електронної пошти, локальних або глобальних мереж, контроль інформації, що приймається [20];
- редагування даних;
- накопичення і зберігання інформації;
- пошук і обробка даних за запитами користувачів;
- висновок на екран або на друк результатів пошуку і обробки;
- захист інформації від різних категорій загроз;
- формування та передача даних на інші комп'ютери на магнітних носіях і по мережі.

Операційне забезпечення включає в себе комплекс документів, що регламентують дії фахівця при використанні АРМ. Документація включає в себе:

- посадові інструкції фахівців АРМ;
- перелік рівнів доступу та паролів до різних категорій даних;
- інструкцію з експлуатації локальних і мережевих АРМ;
- інструкцію з набору і введення інформації;
- перелік завдань, що вирішуються за допомогою АРМ алгоритмів їх вирішення, функціональних можливостей технології;
- перелік вихідних таблиць, графіків, діаграм, що надаються на виході сеансу роботи з АРМ.

Ергономічне забезпечення АРМ включає в себе нормативи, що враховують комплекс фізіологічних, психологічних, антропометричних параметрів людини для безпечної і комфортної організації праці користувача АРМ. До них можна віднести вимоги до розташування і колірної гами використовуваних екранних форм і розташуванню технічних засобів, до освітленості та інших характеристик приміщення, до якості і зручності інтер'єру. Всі ці та багато інших чинників визначають умови охорони та зручності праці головного учасника АРМ – людини.

Для забезпечення ефективності роботи, автоматизовані робочі місця повинні бути реалізовані та адаптовані таким чином, щоб відповідати функціональному призначенню конкретного випадку. Проте загальні вимоги до

реалізації АРМ зазвичай типові: системність, гнучкість, стійкість та ефективність [21].

Згідно з принципом системності АРМ варто розглядати як системи, структура яких узгоджується з відповідним функціональним призначенням. Принцип гнучкості передбачає адаптивність комплексу до потенційних змін та реструктуризації завдяки модульності архітектури всіх підсистем та уніфікації будови відповідних елементів.

Принцип стійкості означає, що АРМ реалізовує основні функції незалежно від впливу на неї як внутрішніх, так і зовнішніх потенційних чинників. Іншими словами, несправності в певних частинах системи повинні бути легко переборні, а робочий та продуктивний стан системи – швидко відновлений.

Ефективність АРМ можна означити як комплексну характеристику рівня втілення вищезгаданих елементів, отриману відносно витрат по створенню й підтримці системи під час використання.

Функціонування автоматизованих робочих місць може принести бажаний ефект лише у випадку раціонального розподілу функцій та навантаження між людиною і програмними засобами обробки інформації, ядром яких є ЕОМ. Тільки за такої умови АРМ може стати засобом підвищення не лише ефективності праці та якості управління, а також і соціальної комфортності фахівців.

Реалізація АРМ передбачає, що головні операції з внесення, збереження та обробки інформації виконуються обчислювальною технікою та програмним забезпеченням, а користувач системи здійснює такі операції, що потребують використання нестандартних підходів у процесі аналізу даних та підготовці відповідних управлінських рішень [13].

За умови реалізації вищезгаданих принципів, використання автоматизованих робочих місць може значно підвищити ефективність праці викладачів. Деякі схожі системи уже використовуються у закладах вищої освіти. Проте їх встановлення, адаптація та підтримка часто вимагає активної роботи професіоналів, що впливає на їх загальну ціну використання. З огляду на це, Міністерство освіти не може забезпечити такими інформаційними комплексами всі навчальні заклади, які цього потребують.

Вибір необхідних освітніх рішень на ринку комп'ютерних послуг досить обмежений і не достатньо різноманітний у питаннях врахування точкових особливостей відповідних навчальних закладів.

Проте основна проблема для реалізації АРМ викладача у конкретному закладі вищої освіти часто полягає у тому, що рішення на ринку зазвичай не враховують особливості навчального закладу, занадто комплексні та пропонують багато функцій, котрі не знаходять застосування, також такі системи часто не мають необхідних саме для викладача шаблонів [10]. З огляду на такі типові проблеми бачимо, що автоматизовані робочі місця для закладів вищої освіти повинні мати гнучке програмне забезпечення та архітектуру, які можливо адаптувати з урахуванням характеристик та особливостей відповідного навчального закладу. Така можливість часто передбачена деякими освітніми системами АРМ, проте реалізація таких можливостей передбачає додаткової плати.

З огляду на це бачимо, що для ефективного та широкомасштабного використання необхідна система з можливістю швидкої та легкої адаптації до необхідних умов.

1.3 Типові елементи онлайн робочого місця викладача

Використання автоматизованих робочих місць є одним з важливих елементів для реалізації онлайн робочого місця викладача. Для забезпечення роботи такого простору, він повинен складатися з деяких типових елементів та відповідати ряду вимог.

Автоматизація такого робочого місця повинна передбачати зокрема:

- операції по пошуку необхідного довідкового матеріалу;
- проведення обчислювальних робіт при мінімальному втручанні людини;
- пошук детальних відомостей (в тому числі необхідних показників) в інформаційному просторі;
- редагування та оформлення результатів роботи в потрібній формі на відповідних носіях або форматах; фонове виконання локальних розрахункових завдань.

Для ефективного функціонування онлайн робочого місця періодично необхідна оперативна обробка великих масивів інформації, включаючи підготовку даних для прийняття рішень на всіх рівнях управління. У зв'язку з цим виникла концепція розподілених інформаційних систем, що передбачає закінчену автоматизовану обробку інформації на різних рівнях ієрархії управління з подальшою передачею необхідних агрегованих даних знизу вгору. Реалізація даної концепції висунула проблему створення на кожному рівні управління точок обробки інформації, що реалізуються у вигляді АРМ. Таким чином, автоматизоване робоче місце фахівця стає найважливішим ланкою в області обробки інформації та новим елементом інформаційних технологій.

Розробка і широке впровадження АРМ та онлайн робочих місць стають можливими також завдяки появі інтерактивних інструментальних засобів.

Інформаційний простір предметної області, включно з нормативно-довідковою інформацією та системами управління базами даних, засобами візуалізації, системою інтерфейсу, агреговані для забезпечення взаємодії та інтеграції цих елементів у межах комп'ютеризованого робочого місця в єдину «лінійку» автоматизованої обробки інформації формують типові елементи для такого онлайн робочого місця.

Для продуктивного використання та адаптації відповідної системи викладачеві, поряд з основами педагогіки і психології, необхідно: знати можливості комп'ютера в своїй предметній області і володіти навичками роботи з ним; мати навички управління пізнавальною діяльністю студентів як під час стаціонарного заняття, так і в період їх самостійної роботи з системою в дистанційному режимі; вміти підбирати і відповідним чином компоувати навчальний матеріал для його реалізації у межах свого АРМ; у співробітництві з програмістами або самостійно розробляти та редагувати елементи формалізованого комплексу інформаційного забезпечення навчальної дисципліни і т. д. Таким чином, професійна діяльність викладача все в більшому масштабі набуває творчого характеру і вимагає від нього підвищення педагогічного майстерності, значних інтелектуальних і часових витрат. У зв'язку з вищесказаним, актуально та важливо позбавити педагога від необхідності виконувати ряд нетворчих, трудомістких процесів і операцій, що займають

сьогодні значну частину його робочого часу. Рішення цієї задачі можливе за допомогою розробки і застосування в вузі автоматизованого робочого місця викладача, доступного для використання в онлайн режимі.

В даний час практично кожне підприємство або організація мають одне або кілька автоматизованих робочих місць, а в багатьох АРМи переросли в системи, що охоплюють кілька десятків або сотень співробітників. Уже кілька десятиліть автоматизується робота бухгалтерів, продавців, менеджерів і т.д. В останні роки розробники програмного забезпечення звернули увагу і на освітні установи. В даний час будь-якого викладача можна порівняти з інженером. Він володіє мобільними знаннями, гнучким методичним і критичним мисленням, потрібними йому для цілеспрямованого аналізу, проектування, конструювання та результативного використання дидактичних об'єктів в навчанні. Основне завдання викладача - управління процесом навчання. Для вирішення цього завдання розробляється безліч спеціальних програм, що полегшують і автоматизують виконання деяких операцій або дій, які виконуються вчителем. Для автоматизації таких процесів доцільно використовувати автоматизоване робоче місце – у даному випадку робоче місце викладача певного ЗВО, обладнане засобами, необхідними для автоматизації виконання ним своїх професійних обов'язків. При цьому інструментарієм є, як правило, персональний комп'ютер, що доповнюється в міру необхідності різними периферійними пристроями та додатковими сервісами. Ми пропонуємо комбінувати такі підходи з використанням актуальних вебтехнологій для забезпечення гнучкості АРМ, та переведення робочого місця для збільшення його ефективності в онлайн формат.

До спеціалізованого онлайн АРМу викладача можна пред'явити ряд загальних вимог, які повинні забезпечуватися при його створенні, а саме:

- наявність засобів обробки інформації;
- можливість роботи в діалоговому (інтерактивному) режимі;
- виконання основних вимог ергономіки;
- достатній рівень продуктивності і надійності комп'ютера та інших технічних засобів, що працюють в системі АРМ;
- раціональність характеру вирішуваних завдань програмним забезпеченням;

- максимальний ступінь автоматизації рутинних процесів;
- достатній рівень сервісу користувачів;
- інші чинники, що забезпечують максимальну комфортність і задоволеність фахівця використанням АРМ як робочого інструменту.

Також одним з типових елементів такої системи повинен бути механізм реалізації конструктивно-оцінної функції роботи викладача: для ефективного контролю та оцінки навчальної діяльності студентів. Такий модуль системи, зокрема, може бути реалізовано у вигляді онлайн-журналу.

Крім цього, важливо також враховувати погодинне навантаження викладача [5]. На початку кожного навчального року викладач отримує погодинне навантаження, список дисциплін, які він повинен «прочитати» протягом навчального року. Щорічно викладач створює або змінює робочу програму з дисциплін, переробляє курси лекцій, практичних занять та контрольні завдання з дисциплін. З роками матеріалу накопичується все більше, і зберігати його стає все важче. Крім того, в процесі навчання студентів постійно приходиться накопичення інформації, яку треба десь тимчасово зберігати і систематизувати. Наприклад, можна зберігати інформацію щодо кожного проведеного заняття, де перераховуються присутні студенти, що тему заняття та кількість годин. Така інформація повинна бути доступна постійно, тому важливо забезпечити викладачеві доступ до неї з використанням баз даних та бекенд-технологій та створити механізми її відображення у інтерфейсних формах вебресурсу.

Такий ресурс може допомогти для забезпечення активної участі викладача у створенні нових екзаменаційних квитків і питань, а також у веденні електронного журналу відвідуваності і успішності по своїй дисципліні [2].

Висока динаміка змін зовнішнього соціально-економічного середовища, яка диктувала вимоги до майбутнього фахівця, визначає необхідність створення засобів інформаційної підтримки викладача на основі принципів адаптивного управління, використання різноманітних освітніх технологій, можливості швидкого зміни цілей і змісту навчання, індивідуального підходу до окремих студентів [27].

Дослідники пропонують різні підходи до вирішення даної проблеми. Одним із варіантів є використання універсальних віртуальних робочих місць,

запозичених з інших сфер, у освітній галузі [12]. Популярними є рішення на основі табличних та текстових редакторів із офісних пакетів, зокрема Microsoft Office, Libre Office, та Google Cloud. Такі рішення часто використовуються у школах, але можуть бути адаптовані і для роботи викладачів в університетах [31]. Крім того існують і більш традиційні погляди, за якими застосунки для викладачів не повинні орієнтуватися на мережу Інтернет, а працювати виключно локально [15, с. 8].

На нашу думку, така ширина існуючих підходів та різноплановість застосування інформаційних технологій свідчить про неповне вирішення даної проблеми. У свою чергу ми вважаємо, що необхідно робити акцент на підходах з використанням мережі Інтернет для забезпечення максимальної гнучкості робочого місця. Такий підхід відповідає сучасним вимогам до навчального процесу, а зокрема його мобільності та можливості проведення у дистанційному форматі, коли у викладача нема доступу до його стаціонарного робочого місця у закладі вищої освіти. Вебресурс повинен бути адаптованим для різних носіїв даних, зокрема стаціонарних комп'ютерів, ноутбуків, планшетів та мобільних телефонів, підключених до мережі Інтернет. Доступ до ресурсу повинен бути захищеним, а вхід у систему можливим лише після проходження ідентифікації. Щодо структури та функціоналу системи, то вони повинні бути такими, що стандартизують та спрощують рутинні обов'язки викладача.

Для реалізації такої системи необхідно пройти такі етапи: виявлення та формалізація першочергових завдань викладача, виявлення особливостей навчального закладу та їх урахування у подальшій розробці системи, аналіз взаємозв'язків завдань викладача, побудова функціональної схеми інформаційного та програмного забезпечення АРМ викладача як комплексної системи, яка передбачає наявність інструментальних засобів для інформаційної підтримки діяльності викладача, забезпечення роботи ресурсу в онлайн режимі. Відповідно до цього, необхідно за допомогою актуальних технологій веброзробки, зокрема бекенд-технологій, формалізувати складові компоненти викладацької діяльності з точки зору її автоматизації.

Необхідно також представити концептуальну модель знань, яка повинна включати в себе предметні і організаційно-методичні знання викладача і модель

студента [4, с. 4]. На основі всього вищесказаного, слід зазначити, що онлайн АРМ викладача повинне бути саме тим інструментом, який дозволить автоматизувати рутинні процеси в його навчальній діяльності та дасть можливість мати постійний доступ до необхідної інформації через інтерфейсні форми вебресурсу [2].

Іншою важливою особливістю онлайн робочого місця є можливість синтезу навчальної програми на основі усталених стандартів та з урахуванням індивідуальних особливостей сприйняття та успішності студентів.

Необхідно також описати типовий навчальний формат, який загалом відповідає різним закладам вищої освіти. У роботі кожного викладача з різних університетів можна виділити загальні процеси, які веде кожен викладач (для прикладу – ведення журналу успішності студентів, журналу відвідуваності, структури курсу). Виділивши всі ці спільні обов'язки та врахувавши особливості конкретного навчального закладу, можна побудувати автоматизоване робоче місце на базі онлайн-ресурсу, яке не тільки автоматизує діяльність викладача, але також може бути тим інструментом, який полегшить документообіг, як в межах кафедри, так і в межах вузу, дозволивши тим самим керівникам кафедри і вузу мати повні звіти про діяльність викладачів і студентів.

Якісна онлайн система для інформаційної підтримки управління якістю навчання на рівні викладача повинна ефективно втілювати ряд функцій:

- реалізувати автоматизацію всіх етапів викладацької діяльності;
- передбачити широкі можливості роботи з різними інформаційними;
- ресурсами;
- дозволити збирати і аналізувати статистичний матеріал за якістю навчання;
- своєчасно виявляти недоліки методичної роботи викладача за рахунок аналізу як структури цілей навчання і змісту навчальних елементів, так і результатів, отриманих в ході навчання;
- адаптувати навчальний процес до постійно мінливих умов;
- формувати екзаменаційні квитки.

Функціонування онлайн АРМ активізує творчу активність, інтенсифікує діяльність, сприяє підвищенню виконавської дисципліни фахівців всіх рівнів.

Основними цілями створення такої системи є:

- вдосконалення техніки і технології управління робочим процесом;
- скорочення термінів підготовки та поліпшення якості прийнятих рішень;
- підвищення рівня інформаційної підтримки процесу виконання професійної діяльності;
- перенесення акценту на творчу діяльність за рахунок вивільнення від рутинної обробки інформації.

Для їх досягнення необхідно рішення наступних завдань [24, с. 10]:

- збір, контроль, фіксація, передача, обробка, зберігання та візуалізація різних по формі, типу і структурованості відомостей;
- інтегрована переробка інформації, необхідної для ефективного ведення робочого процесу;
- накопичення, зберігання і організація доступу до інформації у вигляді баз даних і/або баз знань;
- транспортування інформації з використанням засобів, забезпечених належним захистом;
- оперативний та безперебійний доступ до інформаційних ресурсів з використанням різних пристроїв у мережі Інтернет.

Для реалізації цілей та завдань онлайн робочого місця доцільно компонувати систему на основі модулів, що відповідають за реалізацію одного завдання або їх групи. Певною перевагою запропонованого підходу є те, що модулі можуть працювати незалежно один від одного. Це буде сприяти розширюваності і гнучкості написання як додаткових модулів, так і програмного інтерфейсу для різних випадків.

Завдяки компоненту автентифікації користувача, на одній і тій же базі даних можна створити онлайн робоче місце для безлічі викладачів, а також забезпечувати перегляд результатів навчальної діяльності вищим керівництвом.

1.4. Огляд та порівняльний аналіз існуючих розробок для онлайн робочого місця викладача

В час розвитку інформаційних, мережевих, вебтехнологій виникає все більша потреба в автоматизації та перенесенні в інформаційне середовище ресурсів та систем, які забезпечують покращене та полегшене функціонування та організацію робочих процесів, в яких задіяні зокрема викладачі та студенти закладів вищої освіти. На даний момент більшість рішень на ринку є комплексними та інтегрують у собі багато функцій. Системи не обмежуються автоматизацією робочого місця власне викладача, а переважно формують великі та багатосторонні ресурси для організації процесу навчання, надаючи, в тому числі, певні можливості для полегшення роботи викладача. Тому найбільш схожими системами, які пропонують дотичні до нашої системи автоматизації робочого місця можна вважати модульні системи управління навчанням (learning management systems або LMO) та системи дистанційного навчання.

Сучасні LMS відрізняються зручним та гнучким інтерфейсом, широкими функціональними можливостями і дозволяють вивести дистанційне навчання на якісно новий рівень [19]. Такі системи дозволяють організувати навчальний процес «з нуля», керувати ним від імені викладача, відстежувати успішність учнів за допомогою створення онлайн-курсів або віртуальних класів, доступних в будь-який час і в будь-якій точці світу, де є Інтернет [29]. Всі навчальні матеріали при цьому зберігаються в одному місці, їх зручно адаптувати і переглядати в залежності від цілей навчання і сфери діяльності компанії або організації.

Варто розглянути популярні системи з підтримки, організації та управління навчальним процесом.

Moodle – безкоштовна LMS з відкритим вихідним кодом, орієнтована, перш за все, на організацію взаємодії між викладачем і учнями, хоча також підходить для організації дистанційних курсів та підтримки очного навчання (рис. 1.1).

Moodle – модульна система. Її особливостями можна вважати масштабне співтовариство (яке і розвиває платформу, створюючи нові модулі), широкий

функціонал, можливості адаптації та локалізації для більш ніж 100 мов. На сьогодні є найпопулярнішою безкоштовною системою навчання.

Moodle є безкоштовною лише умовно, оскільки для її підтримки потрібні технічні фахівці, власний або орендований сервер. Якщо кількість студентів не перевищує 500, то в таких випадках вигідніше розглянути комерційні платформи.

The screenshot shows the Moodle course interface. At the top left is the Moodle logo with 'ЦАПУ' below it. A breadcrumb trail reads: 'На головну > Мої курси > Загальноуніверситетські курси > Ознайомлення з можливостями Moodle'. A 'Редагувати' button is in the top right. On the left is a 'НАВИГАЦІЯ' sidebar with a tree structure: 'На головну', 'Моя домашня', 'Сторінки сайту', 'Мій профіль', 'Поточний курс', and 'Ознайомлення з можливостями Moodle' (expanded to show 'Учасники', 'Відзнаки', 'Загальне', '1. Поняття про Moodle', '2. Ресурси Moodle', '2.1. Ресурси "Книга" і "Сторінка"', '2.2. Ресурси "URL", "Файл" та "Тека"', '3. Діяльності Moodle', and '3.1. Форум "Новини"'). The main content area has a red heading 'Вітаємо на курсі "Ознайомлення з можливостями Moodle"'. Below it is introductory text about the course's purpose and activity types. A section titled '1. Поняття про Moodle' is highlighted in blue, with 'Файл: 1 Тести: 2' and 'Прогрес: 1 / 3' on the right. Below that, '2. Ресурси Moodle' is also highlighted in blue.

Рисунок 1.1 - Ознайомлення з можливостями Moodle

Moodle є веборієнтованим середовищем, яке можна масштабувати і налаштовувати. Воно характеризується високим ступенем безпеки і пропонує великий набір інструментів для комп'ютеризованого і дистанційного навчання. Moodle підтримується мережею сертифікованих партнерів, а також має активне співтовариство користувачів і розробників по всьому світу (понад 129 мільйонів активних користувачів). Основною навчальною одиницею системи є навчальний курс.

До переваг системи можна віднести такі елементи: має вбудований конструктор курсів, підтримує асинхронне і змішане навчання, гейміфікацію, мобільне навчання, стандарти SCORM, синхронне навчання, відеоконференції,

для розширення функціоналу доступна система плагінів, відкритий вихідний код дозволяє гнучко налаштувати систему.

Серед недоліків системи варто виділити те, що архітектура та бекенд-частина системи дуже складна, тому самостійне допрацювання Moodle потребує професійних навичок програмування. Це, у свою чергу, вимагає додаткових витрат на підвищення кваліфікації персоналу до необхідного рівня, зокрема вивчення мови програмування PHP.

Moodle – потужна модульна система, яку можна довести до рівня платних систем дистанційної освіти за допомогою готових модулів. Однак впровадження та налаштування до такого рівня може зайняти кілька місяців. Moodle користується популярністю серед освітніх установ, невеликих компаній та ентузіастів електронного навчання. Програма має відкритий код та безкоштовна [19].

Схожою альтернативою є комплексна система Atutor – модульна вебсистема підтримки та контролю навчального процесу (рис. 1.2).

Як і Moodle, Atutor – це відкрита вебсистема для електронного навчання. Серед особливостей можна виділити вбудований редактор курсів, що працює на межі фронтенд та бекенд частин, адаптивний інтерфейс, можливість розширення функціоналу за рахунок модулів [25, с. 297].

ATutor легший для використання новими користувачами, в мережі доступні текстові посібники від активних користувачів системи, а на сайті є докладна документація.

Варто зазначити, що гнучке налаштування бекенд-частини проекту також потребує значних навичок програмування. Установка може викликати складності у користувачів, не знайомих з веброботикою. Адміністратору доведеться розбиратися у всіх тонкощах платформи і вбудованого конструктора.

Систему можна частково налаштувати та пристосувати до конкретних вимог. Функціонал залежить від встановлених модулів та власної розробки. Старі модулі можуть не працювати з останніми релізами ATutor, а підтримуваних модулів не так багато. Тому якщо користувачеві потрібен широкий готовий функціонал, то краще використовувати платні СДО.

За допомогою ATutor можна створювати курси, проводити тести та збирати статистичну інформацію. Модуль BigBlueButton дозволяє проводити вебінари. Є підтримка форумів і функція відстеження активності користувачів.

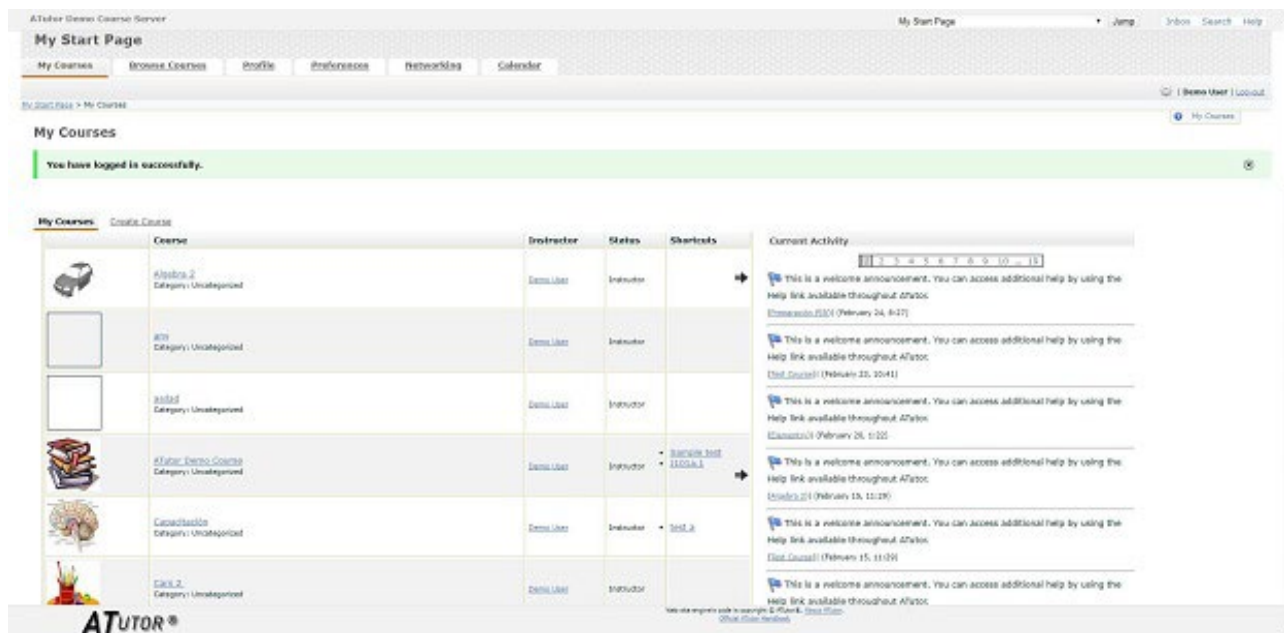


Рисунок 1.2 - Курси в ATutor

У ATutor є вбудований конструктор тестів. Всього доступні 8 типів питань: єдиний вибір, множинний вибір, кілька відповідей, відкрита відповідь, сортування, відповідність, відповідність графіка, шкала Лікерта.

Первинна система звітності ATutor досить обмежена, для більш докладної статистики варто використовувати модуль Test Analysis. Статистика вивантажується в форматі CSV.

У ATutor можна додавати користувачів вручну, імпортом файлу, запрошувати їх по email або дозволяти їм реєструватися самим.

Система має схожі недоліки як Moodle, її гнучке налаштування потребує професійного програмування, крім того кількість готових плагінів досить обмежена. З переваг системи варто виділити наявність конструктора курсів, велике співтовариство користувачів на Github, де можна отримати необхідну допомогу при виникненні проблем з ресурсом. Також систему можна використовувати у різних середовищах, оскільки вона перекладена та локалізована для понад 50 мов.

Наступною популярною альтернативою, що частково покриває необхідні для викладача закладів вищої освіти функції та завдання, є Google Classroom – безкоштовний вебсервіс, розроблений Google для шкіл, який покликаний спростити створення, поширення та оцінку завдань цифровим способом у мережі Інтернет. На цьому ресурсі викладач організовує роботу з декількома курсами, дає завдання, в тому числі індивідуальні, перевіряє їх та ставить оцінки (рис. 1.3).

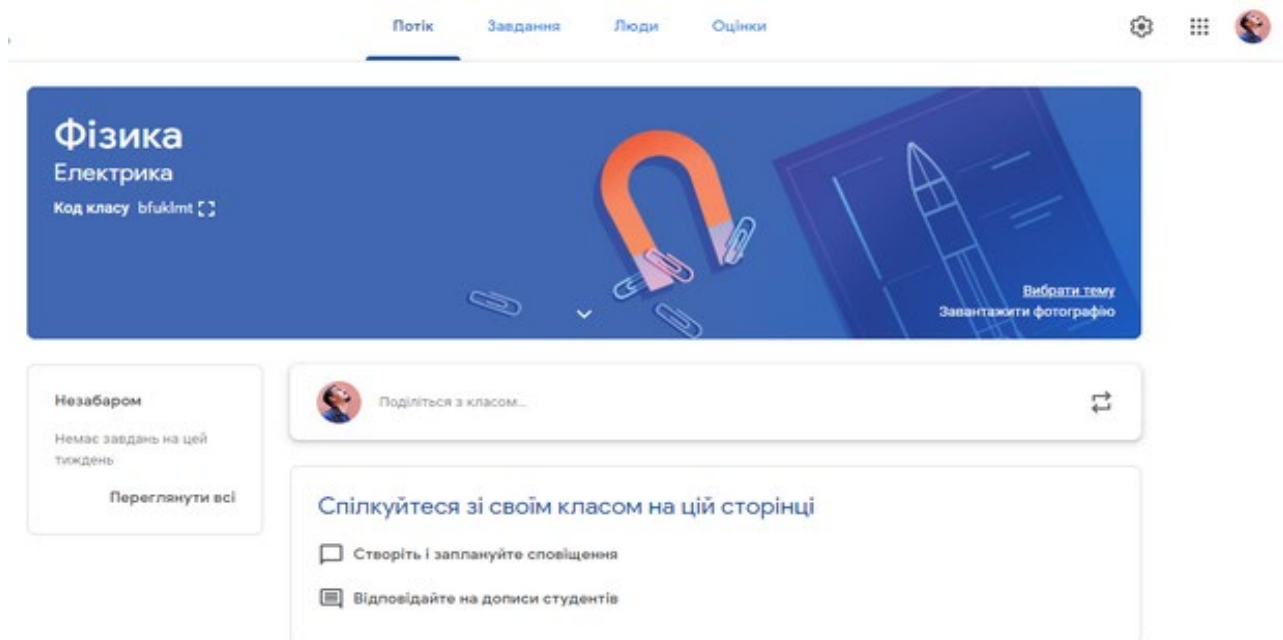


Рисунок 1.3 - Сторінка курсу у Google Classroom

До особливостей середовища можна віднести те, що для кожного класу створюється свій код, який учні можуть використовувати для приєднання до спільноти. Цей процес усуває необхідність створення попередніх реєстрів. Також ресурс інтегрований з Google Drive. Коли викладач використовує Google Classroom, папка «Клас» автоматично створюється на його диску Google з новими вкладеннями для кожного створюваного класу. Аналогічним чином, коли учні використовують Google Classroom, папка «Клас» створюється на сторінці їх Google-диска з вкладеними папками для кожного класу, до якого вони приєднуються. При створенні завдання у вигляді Google-документа, платформа буде створювати і поширювати індивідуальні копії документа для кожного учня в групі. Також при створенні завдання вчитель вказує термін виконання роботи. Коли учень надає завдання до початку терміну, на його документі з'являється

статус «Перегляд», що дозволяє вчителям робити сортування. Коли учні приступили до своєї роботи, вчитель може забезпечити зворотний зв'язок в той момент, коли учень знаходиться в статусі «Перегляд» («Viewing»). Коли робота повертається учневі, школяр знову перемикається в статус «Редакція» («Edit») і продовжує роботу над документом.

Важливо відзначити, що і вчителі, і студенти можуть бачити всі завдання на головному екрані Google Classroom. Це дозволяє контролювати роботу відразу в декількох класах.

Завдяки поєднанню класних оголошень, створених учителем, і інтегрованих можливостей коментування завдань, у викладачів і студентів завжди є можливість підтримувати зв'язок слідкувати за статусом виконання кожного завдання.

До недоліків системи можна віднести те, що у безкоштовній версії сервісу Google Classroom немає електронного журналу. Така можливість доступна тільки для корпоративних користувачів Google Classroom. Крім цього код системи не є відкритим, тому її бекенд-частину не можна проаналізувати та відповідним чином адаптувати до необхідних вимог.

Серед платформ орієнтованих для використання в університетах також варто згадати Ilias (рис 1.4). Ilias – німецька вебплатформа, найбільш поширена в університетах Німеччини. Особливістю є спосіб реалізації, схожий на соціальну мережу, простота використання і налаштування системи, широкий функціонал та потужний редактор тестів. Система найбільше підходить закладам освіти та зокрема університетам. Архітектура та функціонал системи Ilias розроблялися спеціально для ЗВО [33].

На офіційному сайті доступна докладна документація на англійською та німецькою мовами. У мережі є текстові та відео інструкції з використання платформи, проте українською та російською мовами інформації майже немає.

Ilias схожий на соціальну мережу. Так як платформа розрахована під ВНЗ, студентам буде особливо зручно нею користуватися. Процес адміністрування також простий і зрозумілий, однак для установки плагінів потрібно прочитати документацію.

Установкою і налаштуванням платформи повинен займатися фахівець. Інтерфейс і функціонал можна змінювати тільки за допомогою плагінів. Власні функції до бекенд-системи додати неможливо.

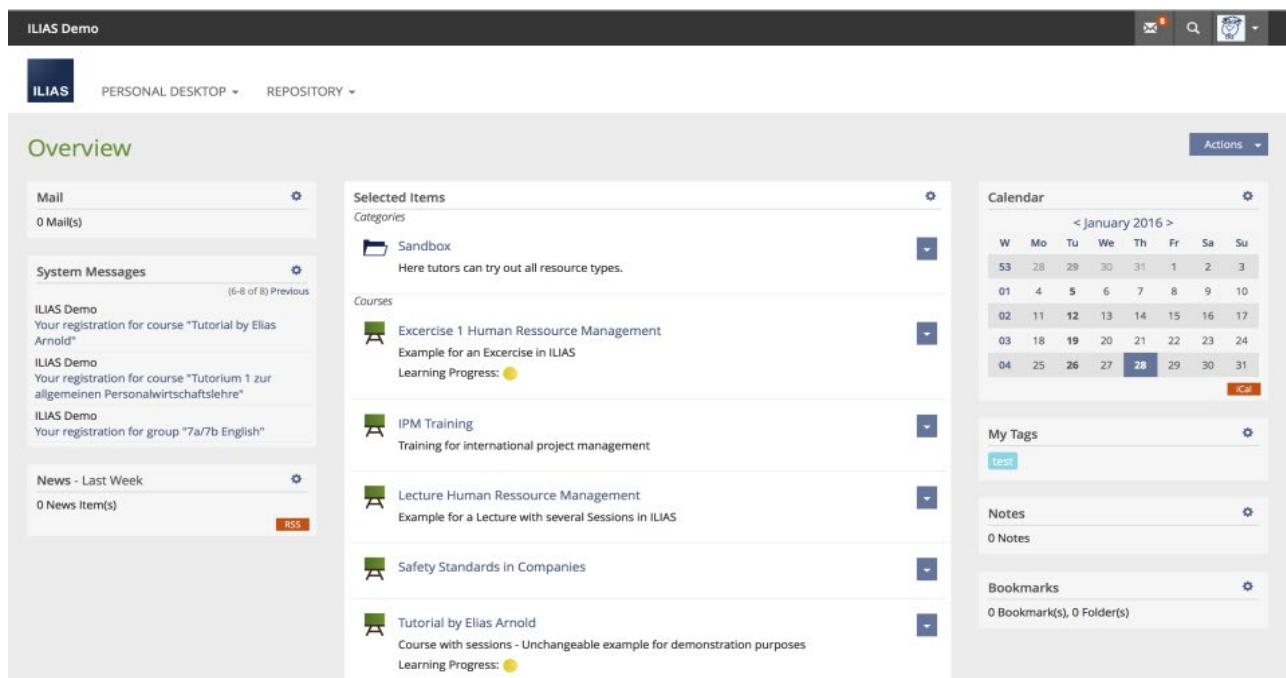


Рисунок 1.4 - Система Іліас

Функціонал платформи досить широкий. У Іліас можна завантажувати, створювати і призначати навчальні матеріали. Є інструменти оповіщення і спілкування: форуми, обговорення, розсилки, списки друзів. У Іліас є вбудований конструктор контенту. З його допомогою можна створювати тести, вправи та опитування. У вбудованому конструкторі тестів можна створювати 17 видів питань. Також доступні елементи звітності. Статистика в Іліас вивантажується в форматах XML і CSV. Колонки в звітах налаштовуються за допомогою плагінів.

У Іліас можна додавати користувачів вручну, імпортом файлу, запрошувати їх по email або дозволяти їм реєструватися самим.

Іліас – система навчання, що нагадує соціальну мережу. Нею легко користуватися для кінцевих користувачів, проте, оскільки система закрита, то налаштування бекенд-частини та функціоналу можливе лише за допомогою готових плагінів. Система розроблялася для німецьких університетів, однак зараз застосовується в навчальних закладах по всьому світу.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ОНЛАЙН РОБОЧОГО МІСЦЯ ВИКЛАДАЧА

2.1 Постановка задачі, призначення та вимоги до програмного засобу «ARMIect»

Автоматизоване робоче місце викладача “ARMIect” призначене для використання у закладах вищої освіти. Система покликана спростити та автоматизувати робочий процес викладача таких закладів. Більшість функцій призначені для реалізації цієї мети, але система в цілому може використовуватися усіма учасниками навчального процесу. Для здобувачів передбачений ряд функцій, в основному інформативних.

За допомогою цього програмного забезпечення можна вносити, отримувати та редагувати інформацію стосовно предметів та занять навчальної програми, працювати з даними про викладачів та студентів, вести облік та контроль навчального процесу, вносити та редагувати дані про успішність та відвідуваність студентів у електронний журнал.

Програмний засіб повинен забезпечувати виконання наступних вимог:

- постійний та безперебійний доступ до системи за допомогою браузера;
- адаптивність для ефективного використання пристроями на основі різних ОС та з різною роздільною здатністю екрана;
- реалізований механізм реєстрації та авторизації користувачів, автентифікації в процесі роботи з системою;
- реалізація чотирьох груп користувачів та розмежування прав;
- збереження формалізованої інформації у базі даних;
- забезпечення безпеки та обмеження доступу до даних користувачів;
- реалізація основних функцій робочого процесу викладача;
- низький поріг входу та легкість роботи з системою.

Належність до певної групи системи задається під час реєстрації користувача. Поки користувач не увійшов у систему вважається, що він гість системи.

Під час використання продукту різні групи користувачів повинні мати доступ лише до набору типових функцій, пов'язаних з їхньою діяльністю.

Викладачі повинні мати можливість:

- читати загальнодоступну інформацію про портал розміщувати та редагувати інформацію про себе на порталі;
- переглядати інформацію про студентів та інших викладачів;
- створювати академічні групи студентів;
- створювати формалізовані сторінки для предметів навчальної програми;
- додавати заняття та їх опис;
- переглядати, вносити дані про відвідуваність та успішність у електронний журнал.

Під час використання продукту здобувачі мають доступ до певного набору функцій, зокрема:

- читати загальнодоступну інформацію про портал;
- розміщувати та редагувати інформацію про себе на порталі;
- переглядати інформацію про студентів та викладачів;
- переглядати дані про відвідуваність та успішність у електронному журналі.

Доступ до системи мають і незареєстровані користувачі, гості системи. Проте набір їх привілеїв обмежений. Вони можуть лише читати загальнодоступну інформацію про портал. Для отримання розширеного доступу та повноважень такі користувачі повинні пройти реєстрацію.

Адміністратор системи має повний набір функцій. Він може переглядати інформацію про всіх користувачів, редагувати її та видаляти користувачів із системи. Представники групи викладачів можуть повністю стати учасниками системи тільки за допомогою додаткового підтвердження від адміністратора.

Користувачі мають деякі опції для персоналізації використовуваного середовища. Зокрема, їм доступний власний профіль, де вони можуть редагувати інформацію про себе, завантажувати профільний малюнок та оновлювати його пізніше.

В кінці роботи користувачі можуть завершити свій робочий сеанс та вийти з системи.

Повинен бути реалізований електронний журнал системи, де ведеться облік інформації про студентів та показники, якими характеризується їх навчальний процес. Зокрема, там можна отримати інформацію про учасників академічних груп, їх відвідуваність та успішність.

Вимоги до інформаційної і програмної сумісності. Серверна частина програмної розробки повинна підтримувати повноцінний режим розгортання та роботи у популярних дистрибутивах на основі Linux, зокрема в Ubuntu, операційних системами групи Windows, починаючи з Windows 7, та операційних системах MacOS. Для розробки програмного забезпечення повинно використовуватись середовище вебпрограмування з підтримкою перевірки синтаксису мов PHP та JavaScript. Код програми повинен бути відкритим та доступним для перегляду. Таким чином, програма має підтримувати подальшу розробку та оптимізацію, адаптацію під умови конкретного закладу вищої освіти, написання додаткових модулів та плагінів користувачами системи. Також на серверній машині повинні бути встановлені вебсервер (Nginx, Apache або ін.), система баз даних, система управління базами даних, реалізована підтримка PHP, бібліотеки Bootstrap 5, jquery.

Система початково орієнтована на автоматизацію роботи викладачів закладів вищої освіти, проте повинна бути можливість її адаптувати до особливостей роботи навчальних закладів різної форми власності та різних ступенів, зокрема шкіл, коледжів та приватних курсів. Повинна бути можливість від системи програмним чином вилучити одну з стандартних функцій, якщо дана функція не відповідає набору необхідних функцій такого закладу. Можливо зробити і зворотню дію та додати функції, котрі не входять до ядра системи, але необхідні для конкретного закладу. Базова система API повинна бути гнучкою та дозволяти додавати нові елементи, котрі відповідатимуть за виконання необхідних завдань. Таким чином програмний засіб відкритий до змін та може адаптуватися до різних вимог та завдань.

Більш детальні вимоги до програмного продукту представлені та описані в Додатку А.

2.2 Вибір моделі розробки програмного засобу «ARMlect»

Життєвий цикл розробки системи (Systems Development Life Cycle) (також життєвий цикл розробки програмного забезпечення) – це процес, в ході якого створюється нова або змінюється стара система програмного забезпечення, а також моделі і методи, які розробники використовують при розробці таких систем. Програмне забезпечення є кінцевим продуктом процесу розробки системи. Процес розробки системи складається як з проектування продукту (дизайну), так і з виготовлення продукту, що відповідає потребам і очікуванням користувачів [35].

Таким чином, мета нашого процесу розробки – виготовити високоякісний продукт, тобто програмне забезпечення, яке може бути виготовлене до встановленого терміну, входити в межі обумовленого бюджету, працювати ефективно та результативно в поточній та плановій ІТ-інфраструктурі та не є дорогим в супроводі і розгортанні.

Коли розглядається програмне забезпечення, як і будь-який інший продукт, його життєвий цикл (процес розробки) можна розділити на фази. Найменування фаз і деталізація поділу на фази змінюється в залежності від конкретного випадку розробки. Проте, перелік основних елементів розробки зазвичай містить деякі стандартні елементи. Типовими частинами життєвого циклу можна вважати: аналіз (analysis), проектування/планування (design), реалізація (implementation) та супровід (maintenance).

Процес життя будь-якої системи або програмного продукту може бути описаний за допомогою моделі життєвого циклу, що складається зі стадій. Моделі можуть використовуватися для подання всього життєвого циклу від задуму до припинення застосування або для представлення частини життєвого циклу, що відповідає поточному проекту. Модель життєвого циклу представляється у вигляді послідовності стадій, які можуть перекриватися та у деяких випадках повторюватися циклічно відповідно до галузі застосування, розміру, складності, потреб у змінах та можливостях. Кожна стадія описується формулюванням мети і очікуваних результатів. Процеси та елементи життєвого циклу відбираються і виконуються на цих стадіях для повного задоволення мети

і результатів кожної стадії. Різні проекти можуть використовувати різні стадії в межах конкретного життєвого циклу.

Для розробки програмного засобу «ARMlect» було вирішено обрати за основу інкрементну модель, а також, де доцільно використовувати елементи інших моделей. На рисунку зображено схему інкрементної моделі розробки програмного забезпечення.

Ідея, що лежить в основі інкрементної моделі, полягає в тому, що програмну систему слід розробляти за принципом збільшень, так, щоб розробник міг використовувати дані, отримані при розробці більш ранніх версій ПЗ (рис. 2.1). Нові дані виходять як під час розробки ПЗ, так і в процесі його подальшого використання. Таким чином ключові етапи цього процесу – проста реалізація підмножини вимог до програми і вдосконалення моделі в серії послідовних релізів до тих пір, поки не буде реалізовано ПЗ у всій повноті. В ході кожної ітерації організація моделі змінюється, і до неї додаються нові функціональні можливості [16].

Одне з ключових питань розробки програмного забезпечення – як працювати зі змінами, оскільки у випадку створення гнучких програмних проектів наявність змін неминуча. Програмний засіб може використовуватись у різних середовищах та закладах освіти, що веде до зміни вимог, створюються нові технології, які доцільно застосовувати для удосконалення програмних модулів, і змінюються самі платформи, для яких створювалася система. Зазначені зміни вимагають переробки та потребують, як правило, повторного аналізу вимог разом з впровадженням, так і реалізації нових функціональних можливостей. Вартість змін повинна бути настільки низькою, наскільки це можливо. Таким чином, в процес розробки необхідно додати такі елементи, які допоможуть допомогти реалізувати потенційні зміни до того моменту, коли їх внесення потребувати уже суттєвого обсягу робіт. Наприклад, створення прототипу допоможе заздалегідь показати клієнту основні характеристики системи, проаналізувати її на відповідність вимогам. Зміни найкраще робити тоді, коли їх внесення по можливості дешеве та не потребує значних часових ресурсів. На підставі цього, виправданий підхід, де передбачена поступова

(інкрементна) розробка і адаптація продукту. Таким чином, зміни можна вносити і в тих частинах, які ще не почали розробляти.



Рисунок 2.1 - Інкрементна модель розробки програмного забезпечення

Інкрементна модель розробки може бути як планова, так і гнучка. Модель забезпечує побудову спочатку невеликої частини системи з подальшим її розширенням в кілька етапів. Поетапний підхід дозволяє розробникам і майбутнім користувачам системи випробувати її починаючи з ранніх ітерацій, отримувати зворотний зв'язок ще тоді, коли можливо зробити зміни, наприклад, в архітектурі системи, без перепису всього коду.

Специфікація програмного забезпечення, проект і його реалізація діляться на частини (increment), які розробляються послідовно одна за одною. Таким чином, зменшується кількість частин системи, що потребують переробки і клієнти отримують можливість протягом більш тривалого часу уточнювати свої вимоги. До типових ознак підходу треба віднести ще те, що компоненти системи (частини), що лише виготовляються уже доступні для використання, що допомагає клієнту випробувати їх в процес розробки, з метою уточнення своїх подальших вимогах, пропонованих до продукту.

Основні етапи розробки, передбачені інкрементною моделлю, можна описати наступним чином. Перш за все, визначаються вимоги в більш загальній

формі і додатково поділяються на більш та менш важливі. Потім визначаються частини програми, тобто, в якій кількості і з чого складаються частини, які почне отримувати клієнт в якості свого програмного забезпечення. Такі частини, які додаються до системи та доставляються клієнту позначають терміном інкремент (increment). Кожна наступна частина повинна додавати до системи певну функціональність. При цьому випуск починають з компонентів (частин) з найвищим пріоритетом. Коли частини системи визначені, беруть першу частину і починають її деталізувати, використовуючи для цього найбільш підходящий процес (зокрема ефективно використання каскадної моделі для такої деталізації). У той же час можна уточнювати вимоги і для інших частин, які в поточній реалізації інкремента не розробляються. Якщо інкремент готовий, то він поставляється клієнту (замовнику), який може використовувати його в роботі (або проводити додаткові точкові тести). Це дозволить клієнтові уточнити вимоги для таких компонентів (або для наступних версій того ж самого компонента). У випадку необхідності, можна згодом повернутися до поточної частини. Після цього займаються наступною частиною системи. Нові частини стикаються з існуючою системою. Різні частини системи не повинні розроблятися в межах одного інкременту.

Переваги інкрементної (поетапної) розробки:

- витрати, які виникають у зв'язку зі зміною вимог користувачів, зменшуються, повторний аналіз і сукупність документації значно скорочуються в порівнянні з каскадною моделлю;
- легше отримати відгуки від клієнта про виконану роботу – клієнти можуть озвучити свої коментарі щодо готових частин і можуть ефективніше контролювати виконання робіт. Таким чином, перші частини системи є своєрідним прототипом всієї системи в цілому;
- у клієнта є можливість швидко отримати і освоїти створене програмне забезпечення – клієнти можуть отримати реальні переваги від системи раніше, ніж це було б можливо з використанням інших моделей.

Для організації інкрементної розробки зазвичай вибирається характерний часовий інтервал, наприклад, тиждень. Згодом, протягом цього інтервалу

відбувається оновлення проекту: додається нова документація як текстова, так і графічна, розширюється набір тестів, додаються нові програмні коди і т.д. Теоретично кроки розробки можуть виконуватися і паралельно, але такий процес дуже складно скоординувати. Інкрементна розробка проходить краще за все, якщо наступна ітерація починається після того, як оновлення всіх артефактів в попередній ітерації закінчено, і істотно гірше, якщо час, необхідний на оновлення артефактів, значно перевищує обраний інтервал.

В результаті кожної ітерації отримується робоче, але не повнофункціональне ПЗ, яке ще не є програмним продуктом і не підлягає поширенню. В результаті кожної ітерації створюється версія деякої частини ПЗ. Необхідно зауважити, що як правило на кожній ітерації визначаються і реалізуються нові вимоги, деякі ітерації можуть бути цілком присвячені удосконаленню існуючої програми, наприклад, з метою підвищення її продуктивності.

З точки зору структури життєвого циклу ми використовуємо інкрементну модель з елементами еволюційної, ітеративної та каскадної моделей. Варто зазначити, що неможливо розглядати кожен з цих підходів ізольовано. Найчастіше таку змішану еволюційну модель називають просто ітеративною (кажучи про процес) або інкрементною (коли йдеться про нарощування функціональності продукту). Значимість такого підходу полягає в організації ітерацій та інкрементів, що особливо корисно для зниження невизначеності із завершенням кожної ітерації.

Розбиттю на елементи може бути підданий не тільки життєвий цикл в цілому, що включає пов'язані стадії — формування вимог, проектування, конструювання тощо, а й кожен інкремент може, в свою чергу, розбиватися на уточнюючі ітерації, пов'язані, наприклад, з деталізацією структури декомпозиції проекту — наприклад, архітектури модулів систем [34].

Даний підхід особливо ефективний та продуктивний для нашого програмного продукту оскільки дозволяє зробити процес розробки більш гнучким, дозволити адаптувати систему до змінних вимог різних навчальних закладів, тестувати прототипи системи уже на ранніх етапах розробки, враховувати та реалізовувати уточнення до неї та виправляти виявлені недоліки

без великих фінансових та часових затрат. В свою чергу, використання елементів інших моделей допомагає прискорити проміжні етапи розробки.

2.3 Опис проекту автоматизованого робочого місця "ARMlect"

Структуру продукту доцільно організувати за модульним принципом з використанням формалізованих наборів інформації у таблицях баз даних, та технологій, що забезпечують механізми введення, систематизації, зберігання, надання та зміни даних з використанням обчислювальної техніки педагогу та студенту [2].

Можна виділити такі основні глобальні модулі цієї системи:

- модуль реєстрації та авторизації;
- модуль збору та надання інформації;
- модуль навчання і перевірки знань;
- модуль оцінки знань.

Ці модулі можуть комбінуватися та взаємодіяти між собою, а також з іншими модулями, в тому числі і розробленими сторонніми організаціями. Таким чином, ці модулі покривають такі області освітнього процесу:

- організаційну, що полягає у веденні різноманітної звітності та журналів;
- інформаційну, що полягає у зборі, оновленні, отриманні та видаленні актуальної інформації про студентів, викладачів, та про предмети навчальної програми;
- оціночну, що дозволяє систематизувати та автоматизувати процес поточного та підсумкового контролю студентів;
- статистичну, що допомагає виявляти динаміку змін у кількісних показниках виміру прогресу студентів та їх відвідуваності, та дозволяє аналізувати ці дані навчальну, що допомагає організувати навчальний процес, зокрема дистанційний, у вигляді представлення інформації про навчальні предмети та окремі заняття, які входять до їх складу, контрольних завдань з метою їх подальшої перевірки, а також оцінний модуль, який по введеним критеріям здатний певною мірою оцінити отримані знання;

- планувальну, що дозволяє на основі статистичної інформації, отриманої в процесі обробки даних, формувати та уточнювати моделі учнів, а також аналізувати навчальну програму та навчальний процес, його ефективність та планувати можливі зміни для їх адаптації та оптимізації;

Перевагою запропонованого підходу є те, що модулі можуть працювати незалежно один від одного. Це буде сприяти розширюваності і гнучкості написання як додаткових модулів, так і адаптації програмного інтерфейсу для уже наявних частин системи.

Вони можуть включати до складу інформаційного масиву різну статистичну, текстову і графічну інформацію в необмеженому обсязі з обов'язковою її формалізацією (поданням, введенням і виведенням на комп'ютер в певній, характерній для даної системи формі). Дії по обробці інформації, що зберігається в базах даних, забезпечуються за допомогою набору функцій спеціального програмного забезпечення - системою управління базами даних (СУБД).

Одиниці інформації присвоюються певні атрибути (автор, ключові слова, регіон, клас інформації, і тому подібні), її пошук здійснюється з використанням структури відповідного підходу. При цьому СУБД передбачає наявність системи редагування для внесення змін до бази даних, систему формування форм звітності за вихідними результатами, систему довготривалого зберігання вкладеної інформації. Їх наявність дозволяє зберігати, отримувати та доповнювати інформацію на різних рівнях вкладеності, забезпечувати захист від несанкціонованого доступу та видалення інформації. Зокрема, в останньому випадку мова йде про доступ до інформації користувачів двох категорій - привілейованих, які мають право вносити зміни в бази даних, і непривілейованих, наприклад гостей системи.

Розглянемо детальніше модулі системи та особливості їх реалізації у програмному засобі.

Модуль реєстрації та авторизації доцільно реалізувати на основі технології з використанням токенів. В наш час вибір технологій та бібліотек, які дозволяють реалізувати таку систему досить широкий. Одним із найпопулярніших рішень є використання системи токенів JWT (JSON web token). Вони особливо корисні

для реалізації систем авторизації та автентифікації та можуть бути використані для проведення проміжних перевірок інформації та статусу активного користувача в різні етапи його роботи з програмою з моменту його авторизації — тоді користувача можна певним чином позначити символічним набором інформації з метаданими про нього, а згодом використовувати її під час виконання ним різних типових операцій. Також за допомогою технологій такого типу досить зручно реалізовувати принцип розмежування прав та створення користувачів різних категорій, які за допомогою використання проміжних перевірок інформаційної мітки мають повний, обмежений або закритий доступ до різних типів інформації або до виконання попередньо визначеного набору дій.

Важливим є також питання організації взаємодії фронтенд та бекенд частин системи. Кожна з частин відповідає за свої типові завдання. Фронтенд частина продукту, з типовими функціями нашої системи, не є статичною, незмінною у часі під час використання тих чи інших компонентів різними користувачами під час різного часу роботи з ними. Тому вона потребує наповнення своїх UI форм та елементів відповідним контентом з бекенд-частини. На цьому етапі виникає питання зберігання та отримання релевантної інформації про користувачів та такої, що має до них відношення, у певній формі та у певному місці. Зазвичай, для цього використовуються системи баз даних різних видів. Для отримання та внесення відповідної інформації необхідна взаємодія та комунікація з бекенд частиною, яка може проаналізувати отриману інформацію та дати релевантну відповідь. Наприклад, під час використання UI форми, куди користувач вносить певні дані, зокрема вносить бал студента, ми, в кінці роботи користувача з нею, надсилаємо цю інформацію на бекенд частину розробки, де така інформація проходить валідацію, попередню обробку та, у випадку успішної перевірки, вноситься у базу даних. Це можна досягти за допомогою методу POST, який використовується переважно для надсилання даних з фронтен на бекенд частину. Ми можемо вказати користувачеві на статус його операції, за допомогою відповіді, яка була отримана з серверної частини.

Доволі типовою є ситуація, коли виникає необхідність динамічно змінювати дані в певній частині інтерфейсу. В такому випадку також потрібно знайти оптимальну технологію для того, щоб посилати запити на отримання чи

змінення інформації у базі даних, а також отримання відповідей з серверної частини програмного засобу. Є група таких технологій, одними з найпопулярніших серед них вважаються AJAX та SOAP. Вони дозволяють налагодити взаємодію двох частин продукту в режимі реального часу, посилати запити у фоновому режимі, не використовуючи базовий тригер, такий як відправлення форм. Це дозволяє забезпечити системі більше гнучкості у генерації динамічного контенту, розширяє можливості вибору для надсилання запитів – запити в такому випадку можна посилати практично в будь-який момент та реагувати при цьому на широкий набір тригерів, які забезпечує фронтенд технологія Javascript. Такі рішення можуть бути корисними у різних випадках, наприклад, коли користувач редагує інформацію про себе і одразу після того бачить її у оновленому вигляді. Також аналогічні підходи можна використати при роботі вчителя з журналом — особливо в процесі його редагування — це дозволяє бачити зміни в режимі реального часу та дає можливість краще контролювати робочий процес, бачити проміжний результат, і, за потреби, його редагувати.

Важливим завданням перших етапів розробки продукту є планування системи баз даних, а саме її структури. Необхідно визначити кількість та найменування таблиць, які входять до бази даних, та обрати оптимальний тип даних, якому відповідають ті чи інші інформаційні рядки таблиці. Таким чином можна описати, формалізувати та структурувати попередньо визначену інформацію, що стосується роботи викладача, його робочого процесу, інформації про студентів та навчальну програму та розподілити її ефективним способом для зручного подальшого використання як при її виведенні, так і при внесенні, редагуванні та видаленні. Тому весь цей інформаційний масив необхідно розподілити за таблицями, рядками та визначити приналежність його елементів до різних типів даних. Ми пропонуємо загальну концептуальну структуру даних, відображену у табл. 2.1, для нашого проекту.

Таблиця 2.1

Концептуальна структура даних

Назва таблиці	№	Назва поля	Опис поля	Примітки
Користувачі	1	Код	Код ідентифікації користувача	Таблиця містить дані про вчителів та студентів
	2	Ім'я		
	3	Прізвище		
	4	Електронна пошта		
	5	Пароль		
	6	Створено	Час реєстрації користувача	
	7	Змінено	Час останнього оновлення інформації про користувача	
	8	Групи	До яких груп належить користувач	
	9	Предмети	З якими предметами навчальної програми працює користувач	
	10	Роль	Роль користувача у системі	
Профільні зображення	1	Код	Код ідентифікації зображення	Інформація про профільні зображення
	2	Назва файлу		
Групи	1	Код	Код ідентифікації групи	Інформація про академічні групи
	2	Найменування	Назва академічної групи	
Предмети	1	Код	Код ідентифікації предмету	Інформація про предмети навчальної програми
	2	Найменування	Назва предмету	
	3	Опис	Опис предмету	
Заняття	1	Код	Код ідентифікації зображення	Інформація про заняття
Заняття	2	Найменування	Тема заняття	

Бал	3	Опис	Опис заняття	Інформація про заняття Інформація про отримані бали
	4	Тип	Вид заняття	
	5	Предмет	Предмет, до якого належить заняття	
	6	Дата	Дата проведення заняття	
	1	Код	Код ідентифікації балу	
Бал	2	Значення	Величина балу	Інформація про отримані бали
	3	Студент	Студент, котрий отримав бал	
	4	Предмет	Предмет, на якому отримано бал	
	5	Заняття	Заняття, на якому отримано бал	
	6	Дата	Дата проведення заняття	

2.4 Обґрунтування вибору інструментальних засобів розробки програмного засобу «ARMlect»

Сучасні засоби розробки програмного забезпечення мають велике розмаїття критеріїв, які при використанні дають розробнику можливість автоматизації процесу розробки додатків. На даний момент за допомогою інструментальних засобів можна:

- створювати інтерфейс з використанням стандартних компонентів;
- передавати управління різним процесам, в залежності від стану системи;
- створювати як оболонки для баз даних, так і самі бази даних; розробляти більш надійне програмне забезпечення, за допомогою обробки виняткових ситуацій, які виникають через некоректну роботу програмного забезпечення.

Сучасні технології розробки характеризуються такими параметрами [35]:

- підтримка об'єктно-орієнтованого стилю програмування;

- можливість використання CASE-технологій, як для проектування розроблюваної системи, так і для розробки моделей реляційних баз даних;
- використання візуальних компонент для наочного проектування інтерфейсу;
- підтримка баз даних;

При виборі інструментальних засобів розробки онлайн АРМ викладача «ARMIest» використовувалися наступні критерії для пошуку необхідних технологій:

- швидкість розробки додатків;
- можливість використання у веброзробці;
- перспективність платформи, розробляється;
- можливість розмежування прав доступу;
- легкість роботи з базами даних, наявність відповідних розширень та функцій, що дозволяють швидко виконувати типові операції;
- простота і зручність, ефективність роботи при створенні форм представлення даних;
- надійність роботи середовища розробки;
- популярність технології та широке коло користувачів, важливо для вирішення можливих проблем.

Для реалізації фронт-енд частини було використано Bootstrap Devtool, який об'єднує три основні інструменти, що є невід'ємною частиною сучасного вебконтенту. HTML – це мова розмітки, яка використовується для структурування та наповнення веб-сторінок. CSS – інструмент для стилізації, за допомогою якого можна додати привабливий зовнішній вигляд до HTML-вмісту: встановити кольори фону та шрифтів, організувати контент у кілька стовпців тощо. JavaScript – мова сценаріїв, яка забезпечує динамічну взаємодію з вмістом: оновлення даних у реальному часі, керування мультимедіа, анімація елементів та багато іншого. Ці інструменти були використані для реалізації основної структури презентації товару, його стилізації та додавання анімаційних ефектів. Це допомогло зробити процес розробки більш простим, автоматизованим та ефективним.

JavaScript, як клієнтська мова програмування, відіграє важливу роль у створенні динамічних та інтерактивних веб-сторінок. Завдяки використанню JavaScript були реалізовані форми, кнопки та елементи інтерфейсу (наприклад, діалогові вікна, підказки, каруселі) з мінімальними витратами часу та без необхідності вручну прописувати кожен елемент окремо.

Для реалізації бекенд-частини «ARMlect» було обрано мову програмування PHP. PHP (Hypertext Preprocessor) – це мова сценаріїв з відкритим вихідним кодом для сторони сервера, що означає, що програми, написані на ній, працюють на вебсерверах і не залежать від веббраузера.

Ми використовуємо її саме для побудови серверної сторони онлайн робочого місця викладача. Серед переваг цієї мови, які важливі для нашого проекту, варто відзначити наступні елементи. Починаючи з версії 5, PHP підтримує об'єктно-орієнтоване програмування, пропонуючи цілком новий набір можливостей. Ми використовуємо елементи ООП для повторного використання коду, структуризації та полегшення подальшої підтримки проекту.

Також, використання PHP призводить до швидшого завантаження сторінок вебсайтів у порівнянні з багатьма іншими технологіями веброзробки. Наприклад, в даний час для більшості сценаріїв використання PHP приблизно втричі швидше, ніж Python. У свою чергу, менший час завантаження є важливим фактором рейтингу SEO, який допомагає надалі просувати вебсайт, забезпечуючи конкурентні переваги. Більш висока швидкість застосування забезпечує задоволення клієнтів і, в поєднанні з іншими перевагами, допомагає побудувати та зберегти клієнтську базу [42].

PHP є відкритим і безкоштовним, що допомагає розробникам швидко та легко встановити його для використання. Існує багато фреймворків PHP, і розробник може вибрати будь-який фреймворк для роботи та власної адаптації ресурсу. Також PHP пропонує вбудоване програмування HTML, що є причиною значної синергії між цими двома технологіями. PHP дозволяє створювати відповіді з системи API одразу в форматі розмітки HTML.

Однією з принципових вимог до продукту є легкість встановлення та використання у різних середовищах. Бекенд системи, написаний на PHP може бути розгорнутий у більшості відомих операційних систем, таких як Linux, Unix, Solaris, Microsoft Windows, MAC OS та багатьох інших. Мова також підтримує більшість вебсерверів, включаючи Nginx, Apache та IIS. Використання PHP надає свободу вибору операційної системи та вебсервера. Таким чином систему можна реалізувати практично у всіх середовищах закладів вищої освіти. Зокрема, найбільш поширеним сценарієм роботи вебсайту PHP є стек LAMP. Це означає, що вебсайт працює на вебсервері Apache HTTP, розгорнутому в системі Linux, і використовує MySQL як базу даних. Всі ці компоненти безкоштовні, а стек добре перевірений, що передбачає скорочення необхідного часу та коштів на розробку. Також добре працює у складі стеку XAMP, на основі Windows.

Мова також відповідає нашим вимогам щодо роботи з базами даних. PHP легко підключається до бази даних і надійно з'єднується з ними. Він має вбудований модуль, який використовується для легкого підключення до бази даних. Це також скорочує час підключення до систем управління базами даних. Кілька різних баз даних можна інтегрувати з PHP, зокрема пропонується розширена підтримка MySQL, яка активно використовується у нашому проекті.

Також PHP пропонує і можливість встановлення на основі хмарних середовищ, що дає ще більше варіативності для запуску системи для нового навчального закладу. У наш час багато сучасних продуктів, як правило, використовують хмарні обчислювальні рішення, такі як Amazon Web Services, для різних цілей. Додатки, написані на PHP, підтримуються різними хмарними сервісами, такими як AWS Lambda. Таким чином, PHP-програму можна розгорнути на хмарному сервері та досягти чудової масштабованості та інших корисних ефектів.

При виборі системи програмування було розглянуто різні середовища розробки додатків. Враховуючи те, що при реалізації бекенд-частини ми використовуємо мову PHP, було обрано середовище розробки PhpStorm, яке ми вважаємо оптимальним для роботи з нею та інтеграції з іншими інструментами, що використовуються у проекті.

PhpStorm, IDE на основі Java (інтегроване середовище розробки), – це інноваційний редактор коду, створений JetBrains для допомоги веброзробникам та, зокрема, розробникам PHP. Використовуючи IntelliJ IDEA, програмне забезпечення здатне надати розробникам платформу для написання коду на PHP а також мовами та стандартами, такими як Emmet, JavaScript, HTML5 та CSS3. В основному це полегшена версія IntelliJ, яка надає додаткові функції для підтримки PHP. PhpStorm надає першокласну допомогу в кодуванні, всебічне розуміння кодів та підтримку PHP-інструментів та основних середовищ IDE. Програмне забезпечення оснащене різними корисними функціями, які можна розширити для подальшої зручності та продуктивності завдяки підтримці плагінів. Він використовує передові технології інтерфейсу і здатний до надійного рефакторингу, налагодження та модульного тестування. Він може безперешкодно працювати з такими фреймворками та CMS як Joomla, Laravel, CakePHP, WordPress, Zend, Symfony, Drupal та Magento.

Нами було обрано цю систему через ряд переваг. По-перше IDE забезпечує розумне доповнення коду, виділення синтаксису, розширену конфігурацію форматування коду, перевірку помилок на льоту, компілювання коду, підтримує змішування мов тощо. Автоматизовані рефакторинги, непомітно супроводжують процес розробки, допомагаючи легко та безпечно робити глобальні налаштування проекту [39]. Також інструмент дуже корисний для виявлення помилок в процесі розробки. Сотні перевірок коду перевіряють код під час набору блоків програми та перевіряють весь проект на наявність можливих помилок або недоліків. Швидкі виправлення для більшості перевірок дозволяють легко виправити або вдосконалити код в режимі реального часу.

Крім цього середовище розробки забезпечує легку навігацію та пошук у коді. PhpStorm допомагає ефективніше здійснювати пошук по коду та заощадити час при роботі з складною та багатокomпонентною структурою проекту. Можна перейти до методу, функції або визначення змінної лише одним клацанням миші або знайдіть місця їх використання у всій структурі системи.

Важливо, що середовище також може функціонувати як редактор HTML і CSS. Підтримуються всі передові технології вебробки, включаючи HTML5, CSS, SASS, LESS, ECMAScript тощо. Редагування в реальному часі дає

можливість миттєво бачити всі зміни в браузері, не оновлюючи сторінку. Також доступна і розширення для роботи з Javascript, це дозволяє нам зручно та ефективно створювати модулі для взаємодії фронтенд та бекенд частин проекту в одному редакторі коду.

Для контролю над процесом розробки та життєвим циклом проекту важливо мати зручні механізми роботи з системами контролю версій. PhpStorm надає уніфікований інтерфейс для роботи з багатьма популярними системами керування версіями, забезпечуючи стабільну взаємодію із git, GitHub. Це дозволяє покращити командну роботу над проектом та забезпечує йому більше гнучкості для відновлення даних та тестування нових функцій.

Однією в вимог до середовища розробки ми визначаємо зручність роботи з базами даних. PhpStorm надає інструменти та функції доповнення коду для роботи з базами даних та SQL у проектах. Зокрема, можна налаштовувати підключення до баз даних, переглядати дані прямо у середовищі, редагувати схеми та табличні дані, запускати запити та аналізувати схеми за допомогою діаграм UML. Код SQL можна вводити з використанням бекенд мов програмування або редагувати безпосередньо в редакторі SQL, з підсвічуванням синтаксису, розумним доповненням коду, аналізом коду на помилки та неточності, форматуванням коду та навігацією.

Для роботи з формалізованою інформацією для нашого проекту було вирішено зберігати її у базах даних для забезпечення її структуризації, захисту та зручністю роботи.

База даних – це окрема програма, яка зберігає колекцію даних. Кожна база даних має один або кілька різних API для створення, доступу, управління, пошуку та тиражування даних, які вона містить.

Також ми розглядали інші види сховищ даних, такі як файли у файловій системі або великі хеш-таблиці в пам'яті, але отримання та запис даних не буде настільки швидким і простим із такими типами систем.

В даний час для проекту ми використовуємо реляційні системи управління базами даних (СУБД) для зберігання та управління нашим обсягом даних. Це називається реляційною базою даних, оскільки всі дані зберігаються в різних

таблицях, а зв'язки встановлюються за допомогою первинних ключів або інших ключів, відомих як зовнішні ключі [40].

Було вирішено використовувати реляційну систему управління базами даних (RDBMS), оскільки вона відповідає ряду вимог до системи:

- дозволяє реалізувати базу даних з таблицями, стовпцями та індексами;
- гарантує референтну цілісність між рядками різних таблиць;
- оновлює індекси автоматично;
- інтерпретує комбіновані запити SQL та дозволяє поєднувати інформацію з різних таблиць.

Конкретною такою системою для нашого проекту було обрано MySQL. Це безкоштовна система баз даних з відкритим кодом, яка полегшує ефективне управління базами даних та пропонує механізми підключення їх до програмного забезпечення та серверних мов програмування. Це стабільне, швидке, надійне та потужне рішення з розширеними функціями.

MySQL випускається за ліцензією з відкритим кодом. Це великий плюс для реалізації на основі державних закладів вищої освіти, оскільки це дозволяти зменшити затрати на реалізацію та подальшу роботу проекту [40].

Перевагою є і те, що MySQL використовує стандартну форму добре відомої мови даних SQL, що дозволяє використовувати гнучкий набір команд для роботи з інформацією. MySQL працює в різних популярних операційних системах та багатьох мовах бекенд-розробки, включно з PHP.

Також використання MySQL оптимальне для швидкої та багатофункціональної роботи навіть з великими наборами даних, що дозволяє прискорити загальну роботу нашої розробки.

MySQL добре підтримується PHP, мовою яку ми використовуємо для бекенд-розробки «ARMlect», що дозволяє спростити роботу з типовими операціями щодо підключення до баз даних та роботою з інформацією викладачів та студентів.

Також MySQL підтримує і великі бази даних, до 50 мільйонів рядків та більше в таблиці. Це важливо для потенційного створення системи для великих навчальних закладів або їх груп.

Очевидно, наше онлайн АРМ потребує легкої та зручної взаємодії з даними та можливостей управління ними без написання програмного коду. Для цієї мети ми обрали вебзастосунок phpMyAdmin, написаний на мові PHP. PhpMyAdmin – одна з найпопулярніших, безкоштовних програмних платформ з відкритим кодом для адміністрування даних MySQL та MariaDB у мережі Інтернету [38]. Цей інструмент зручно використовувати як під час розробки, так і під час подальшої підтримки програмного засобу адміністратором системи у вищому навчальному закладі.

Графічний інтерфейс керує багатьма операціями, пов'язаними з даними, такими як створення, модифікація, видалення даних, зміна структури таблиці, які знаходяться в базах даних. Він може одночасно керувати кількома базами даних. Він також виконує інші дії, такі як перегляд даних, виконання SQL-запитів, а також управління індексами, користувачами та дозволами. Це також забезпечує такі функції, як резервне копіювання, виконання пошукових операцій, імпорт та експорт даних у різних форматах, таких як SQL, CSV, XML, аркуші Excel.

Загалом це найбільш надійний та зручний менеджер баз даних, який в основному використовується для вебдодатків або програм. Щоб встановити програмне забезпечення phpMyAdmin потрібен сервер, на базі операційної системи Windows, Linux або MacOS. Для запуску інструменту потрібен інтерфейс веббраузера, там за допомогою графічного інтерфейсу можна виконувати типові операції SQL.

Для встановлення системи потрібен вебсервер для зберігання файлів phpMyAdmin, це можна зробити на основі стеку, де розгорнуто АРМ. Також застосунок може підключатися до MySQL для управління даними додатку, відображення їх у зручних інтерфейсних формах, що особливо зручно для підтримки онлайн робочого місця адміністраторами системи. Для ОС Windows вебзастосунок найлегше розгортати у складі пакету XAMPP – це найпростіший спосіб отримати інструмент phpMyAdmin [37].

Варто виділити деякі особливості інструменту phpMyAdmin, через які ми додали його до стеку інструментів:

- дозволяє керувати дозволами різних користувачів серверної ОС, що зручно для розподілення ролей між адміністраторами у навчальному закладі;

- виконує SQL-запити, пакетні запити, тригери, події, збережені процедури, функції для відображення відповідних результатів обробки даних;
- виконує всі операції, пов'язані з даними, такі як створення, читання, редагування, видалення, перегляд, зміна структури та схеми, пошук даних;
- надає інструменти для імпорту, експорту та завантаження даних, текстових файлів, що важливо для створення резервних копій та зміни сервера;
- екпортує дані у різні формати, такі як CSV, XML, PDF, документ Word, LaTeX, електронна таблиця, файл Excel та інші формати, це можна використати для формування звітів;
- підтримує складні запити SQL, таким чином можна сформулювати швидкі та ефективні механізми роботи з інформацією;
- підтримує InnoDB, зовнішні ключі та MySQLi;
- забезпечує зручний графічний інтерфейс доступу до даних вебсайту;
- дозволяє генерувати графічний макет бази даних;
- дозволяє генерувати SQL запити, а також перетворювати їх у формат PHP, що допомагає при створенні додаткових модулів АРМ;
- забезпечує адміністрування декількох серверів одночасно, що дозволяє в перспективі масштабувати систему;
- налаштувати інструмент просто, це не потребує професійних навичок програмування;
- елементи та функції графічного інтерфейсу зручні та доступні для розуміння розробникам та адміністраторам;
- оскільки це вебінструмент, до нього можна отримати доступ з будь-якої комп'ютерної системи через мережу Інтернет та адмініструвати бази даних «ARMlect» дистанційно;
- не вимагає підключення до мережі, оскільки встановлюється на серверній машині з підтримкою MySQL;
- вигідно у порівнянні з консоллю, оскільки багато завдань, котрі потребують від адміністратора знань консольних команд, доступні у графічному інтерфейсі користувача;
- забезпечує автоматичне резервне копіювання MySQL;

Для командної роботи над проектом, особливо під час реалізації та адаптації онлайн робочого місця в навчальному закладі зручно також використовувати популярну систему контролю версій Git. Вона дозволяє реалізувати набір важливих функцій:

- контролювати процес впровадження;
- тестувати систему без ризику втрати даних;
- слідкувати за історією змін у проекті;
- дає можливість доповнювати та вдосконалювати проект стороннім розробникам.

Використання цих технологій підвищує надійність та стабільність функціонування, легкість розробки та адаптації програмного засобу «ARMlect».

2.5 Основні режими функціонування програмного засобу «ARMlect»

Одним із базових та найважливіших елементів системи є модуль реєстрації та авторизації. Він відповідає за ідентифікацію користувачів, відповідну персоніфікацію контенту та захист інформації. Було вирішено, що просту систему реєстрації та авторизації варто доповнити сторонніми технологіями для забезпечення додаткового захисту та реалізації проміжних елементів перевірки. Для цієї мети було створено додатковий підмодуль системи, реалізований на основі технології з використанням токенів. В наш час вибір технологій та бібліотек, які дозволяють реалізувати таку систему досить широкий. Одним із найпопулярніших рішень є використання відкритого стандарту для створення токенів доступу JWT (JSON Web Token).

Цю технологію ми використовуємо для реалізації комплексних систем авторизації та автентифікації, що дозволяє генерувати спеціальний шифр на основі приватного ключа та здійснювати проміжні перевірки інформації та статусу активного користувача в різні етапи його роботи з програмою з моменту авторизації. На даному етапі серверна частина системи перевіряє авторизаційні дані користувача, та у випадку їх релевантності та успішного входу, надсилає на фронтенд частину системи спеціальний унікальний шифр, який буде ідентифікувати користувача під час його сеансу роботи з системою. Цей шифр

ми надалі зберігаємо у сховищі cookies користувача, тому подальші доступ до нього та використання є простими та зручними. Цей символічний набір інформації надалі перевіряється під час виконання користувачем різноманітних типових операцій. Таку технологію та загальний підхід ми використовуємо, зокрема, для реалізації розмежування прав та автентифікації користувачів, які за допомогою використання проміжних перевірок інформаційної мітки мають повний, обмежений або закритий доступ до різних типів інформації або до виконання певного попередньо визначеного набору дій. Наприклад, коли користувач системи надсилає запит на зміну профільного зображення, то такий запит відбудеться лише у випадку успішної перевірки токена користувача. Це дозволяє захищати дані користувачів на рівні усіх типових операцій з цими даними: внесенням, переглядом, редагуванням та видаленням — такі операції мають право здійснювати лише користувачі з релевантним токеном. Таким чином, система захищена від несанкціонованого доступу до виконання запитів на бекенд частину неавторизованими користувачами та ботами.

Також такий підхід використовується для керування доступом по різних сторінок платформи. Сторінки платформи поділяються на загальнодоступні та такі, що доступні лише авторизованим користувачам. Для першої групи сторінок ми не виконуємо жодних перевірок, тому їх можуть переглядати всі користувачі системи, зокрема і гостьові. Для доступу та перегляду другої групи сторінок ми здійснюємо перевірку валідності шифру користувача, і лише в успішному випадку показуємо йому відповідний контент.

Коли розроблено систему реєстрації та авторизації, формалізовано й додано інформацію у базу даних та створено основу інтерфейсних частин постає питання організації взаємодії між фронтенд та бекенд частинами системи. Кожна з частин відповідає за свої типові завдання. Проте без взаємодії між ними загальна система не буде функціонувати. Фронтенд вісь нашої системи «ARMіст» не є статичною та незмінною у часі під час використання різноманітних компонентів унікальними користувачами під час різних етапів роботи з ними. Тому вона потребує диверсифікованого наповнення своїх UI форм та елементів відповідним контентом. На даному етапі постає питання щодо механізму отримання та редагування релевантної інформації про користувачів та

такої, що має до них відношення. У нашому випадку ця інформація зберігається у базі даних MySQL. Для отримання та внесення відповідної інформації необхідна взаємодія та комунікація з бекенд частиною, яка аналізує дані, що отримуються з фронтенд частини та у результаті дає релевантну відповідь. Наприклад, під час використання UI форми, куди користувач вносить певні дані, зокрема під час проходження реєстрації у системі, вони надходять на бекенд частину розробки, де проходять валідацію, попередню обробку (приведення до уніфікованого формату), фільтрацію різними функціями PHP, та у випадку успішної перевірки, вносяться у базу даних завершуючи процес реєстрації користувача з такими даними в системі. Відправка форм здійснюється переважно за допомогою методу POST, який використовується для транспортування відповідних даних з фронтенд на бекенд частину. Також після здійснення POST-запиту ми можемо виконати певну дію, наприклад вказати користувачеві на статус успішності/неуспішності його реєстрації у системі. Схожий механізм використовується також при авторизації користувача, зміні профільних даних, та під час інших етапів роботи з онлайн робочим місцем.

Доволі важливими та типовими є такі ситуації, коли, навіть під час сеансу одного користувача, виникає необхідність динамічно змінювати дані в певній частині інтерфейсу без перезавантаження браузера. Наприклад, під час оновлення профільного зображення або редагування даних користувача. Після виконання цих дій відповідні інформаційні поля на сторінці оновлюються. Для вирішення таких завдань ми підібрали оптимальну технологію для того, щоб посилати запити на отримання чи змінення інформації у базі даних. Для деяких ситуацій нам достатньо відправки форм та послідовного методу POST, що використовується для них. Проте у випадку, коли необхідно динамічно оновлювати дані, ми використовуємо набір технік веброзробки AJAX, за допомогою яких ми генеруємо асинхронні запити на отримання та обробку даних бекенд частиною. Так ми налагоджуємо взаємодію клієнтної та серверної частин продукту в режимі реального часу, та надсилаємо запити у фоновому режимі, не використовуючи базовий тригер, такий як відправлення форм та при цьому не порушуючи процеси, які відбуваються на сторінці. Це дозволяє забезпечити системі більше гнучкості у генерації динамічного контенту, розширяє

можливості вибору для надсилання запитів, запити в такому випадку можна посилати практично в будь-який момент, реагувати при цьому на широкий набір тригерів, які задає фронтенд технологія Javascript.

Такі рішення ми використовуємо коли користувач виконує певні дії, як наслідок, без перезавантаження браузера у фоновому режимі відправляються запити, та, у деяких випадках інформація на сторінці видозмінюється та відображається в оновленому вигляді. Також аналогічні підходи ми використовуємо при роботі вчителя з журналом — особливо в процесі його редагування, наприклад, коли він виставляє бали — так користувач бачить зміни в режимі реального часу та має можливість краще контролювати свій робочий процес, відслідковувати проміжний результат, за потреби його редагувати і, як наслідок, уникати помилок.

Взаємодія частин продукту значним чином базується на системі API. API – це набір програмного коду, що забезпечує передачу даних між одним програмним продуктом та іншим або частинами одного продукту. Він також містить умови цього обміну даними. У питаннях взаємодії ми слідуємо принципам REST з метою реалізації RESTful API. REST – це скорочення від Representational State Transfer. Це архітектурний стиль, для якого характерна група вимог, дотримання дозволяє уніфікувати різні ресурси та розробку.

Зокрема, у нашому ресурсі ми слідуємо таким вимогам:

- клієнт-сервер – відокремлюючи проблеми, пов'язані з користувальницьким інтерфейсом, від проблем зберігання даних, ми покращуємо легкість та зручність використання одного користувацького інтерфейсу на декількох платформах та покращуємо масштабованість, спрощуючи та виділяючи компоненти сервера;
- відсутність стану – кожен запит від клієнта до сервера повинен містити всю інформацію, необхідну для розуміння запиту, і не може використовувати будь-який збережений контекст на сервері. Тому стан сеансу повністю зберігається на клієнтській частині;
- уніфікований інтерфейс елементів системи. Це спрощує та диверсифікує архітектуру, що дозволяє кожній частині розвиватися незалежно, але при цьому легко комбінуватися;

- індивідуальні ресурси ідентифікуються у запитах, наприклад, використовуючи URI у вебслужбах RESTful. Самі ресурси концептуально відокремлені від подань, що повертаються клієнту. Наприклад, сервер може надсилати дані зі своєї бази даних у форматі HTML, XML або як JSON – жоден з них не є внутрішнім представленням сервера.

Під час розробки груп користувачів та системи API ми використовували елементи об'єктно-орієнтованого програмування. Обрана мова PHP підтримує об'єктно-орієнтоване програмування, пропонуючи повний набір можливостей. Ми використовуємо елементи ООП для оптимізації повторного використання коду за допомогою механізмів наслідування, для загальної структуризації бекенд-частини проекту та полегшення подальшої підтримки програмного засобу. Зокрема, всі елементи API реалізовані за однаковим принципом у вигляді методів класів, що спрощує та стандартизує можливість додавання нових таких елементів у систему. Також елементи API описані в окремому файлі, де у формалізованому вигляді особливості та вимоги до використання конкретних одиниць.

Основні функції реалізації подано в додатку Б.

Важливий елемент роботи над онлайн робочим місцем викладача – робота зі збереженням та обробкою даних. Загальна структура бази даних та зв'язки між її елементами відображені нижче (рис. 2.1).

Для реалізації такої системи таблиць було здійснено ряд кроків. Спочатку описано та формалізовано елементи робочого процесу викладача. Інформаційні елементи цих процесів було виділено в окремі одиниці, класифіковано за типами даних та призначенням та, після завершення класифікації, додано у вигляді таблиць та відповідних рядків у реляційні бази даних MySQL, систему яких було розгорнуто на основі стеку LEMP в операційній системі Linux. Одиниці інформації було проаналізовано на предмет їхніх взаємозв'язків з іншими елементами робочого процесу та одиницями інформації. Відповідні зв'язки було реалізовано за допомогою системи ключів, доступної у базах даних MySQL – Primary Key (для позначення батьківських елементів) та Foreign Key, для позначення елементів, котрі наслідують ключі першого типу.

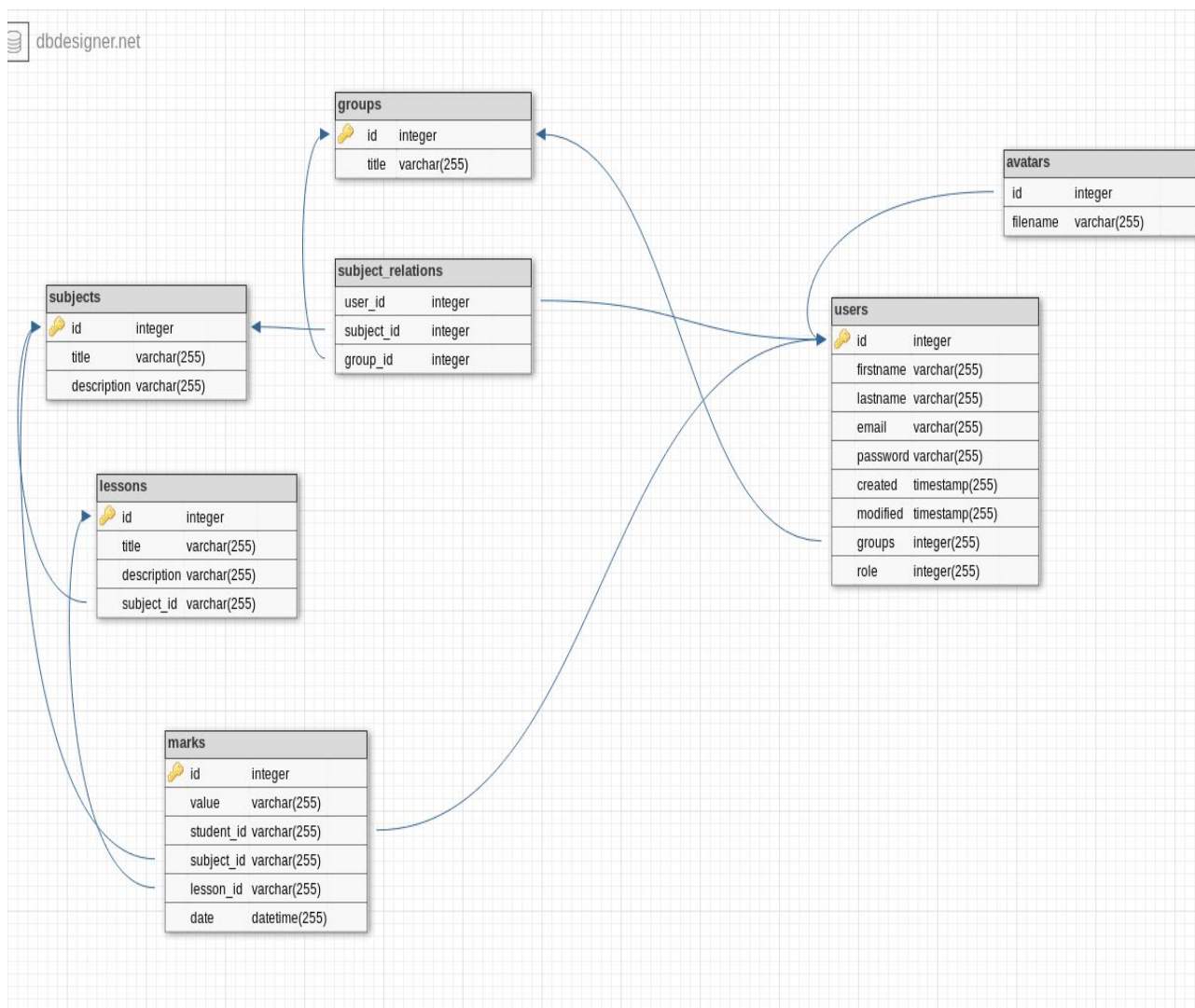


Рисунок 2.1 Загальна структура бази даних програмного засобу

2.6 Організація тестування та налагодження програмного засобу «ARMlect»

Тестування програмного забезпечення – це спосіб перевірити, чи відповідає фактичний програмний продукт очікуваним вимогам, і забезпечити відсутність дефектів програмного продукту. Він передбачає виконання програмних/системних компонентів за допомогою ручних або автоматизованих інструментів для оцінки однієї або декількох властивостей, що перевіряються. Метою тестування програмного забезпечення є виявлення помилок, недоліків або невиконаних вимог.

Тестування програмного забезпечення має важливе значення, оскільки, якщо в програмному забезпеченні є помилки або потенційно вразливі місця, їх

можна виявити заздалегідь і вирішити перед поставкою програмного продукту. Правильно протестований програмний продукт забезпечує надійність, безпеку та високу продуктивність, що в подальшому призводить до економії часу, економічності та задоволеності споживачів [41].

Це особливо важливо в процесі адаптації програмного засобу до особливостей використання у конкретному закладі вищої освіти для забезпечення максимальної ефективності саме в таких умовах.

Тестування в процесі розробки нашої системи – процес її оцінки, необхідний для того, щоб визначити, чи відповідає поточний програмний інкремент необхідним умовам. Процес тестування включає оцінку особливостей програмного продукту щодо вимог з точки зору відсутніх або повністю задоволених вимог, помилок, потенційних ризиків та продуктивності.

Під час етапів тестування в процесі розробки програмного забезпечення популярні такі стратегії.

Модульне тестування (Unit Testing). Цього підходу до тестування програмного забезпечення дотримується програміст для тестування модуля програми. Це допомагає розробникам зрозуміти, чи працює окрема одиниця коду належним чином.

Тестування інтеграції. Така стратегія зосереджується на побудові та дизайні програмного забезпечення. Тести такого типу перевіряють чи інтегровані блоки працюють ефективно та без помилок.

Тестування системи. На основі цього методу програмне забезпечення компілюється як одне ціле, а потім перевіряється ефективність роботи механізму в цілому. Ця стратегія тестування перевіряє функціональність, безпеку, портативність та інші властивості інтегрованої системи.

Такі три стратегії при тестуванні програмного забезпечення – це метод виконання фактичного програмного коду з метою перевірки поведінки програми та пошуку помилок. Програма виконується з даними тестового випадку для аналізу поведінки програми або відповіді на тестових даних. Ефективним тестуванням програми вважається таке, що має високі шанси знайти помилки.

Одним з найпопулярніших підходів до тестування є White Box Testing. White Box Testing – це техніка тестування програмного забезпечення, в рамках

якої внутрішня структура, дизайн та код програмного забезпечення тестуються для перевірки потоку вхідних даних та покращення ефективності, зручності використання та безпеки. У тестуванні такого типу код відкритий тестувальникам, тому його також називають тестуванням Clear box, тестуванням у відкритих коробках, тестуванням прозорих коробок, тестуванням на основі коду та тестуванням Glass box.

Це одна з двох частин підходу Box Testing до тестування програмного забезпечення. Його аналог, тестування Black Box, включає тестування з точки зору зовнішнього або кінцевого користувача, котрому не доступний програмний код засобу і тестування виконується через інтерфейс системи. З іншого боку, тестування методом «білої скриньки» під час розробки базується на внутрішній роботі програми та обертається навколо внутрішнього тестування.

Термін «White Box» був використаний через концепцію прозорого вікна. Ясна рамка символізує здатність проникати через зовнішню оболонку програмного забезпечення у внутрішні роботи. Подібним чином, «чорний ящик» у Black Box Testing символізує неможливість побачити внутрішню роботу програмного забезпечення, та використовується щоб перевірити лише досвід кінцевого користувача.

Тестування білого ящика передбачає тестування програмного коду на наступне:

- слабкі сторони внутрішньої безпеки;
- порушені або погано структуровані шляхи в процесах кодування;
- робота вхідних механізмів для даних;
- очікуваний результат;
- функціональність умовних циклів;
- тестування кожного твердження, об'єкта та функції на індивідуальній основі.

Тестування можна проводити на системному, інтеграційному та модульному рівнях розробки програмного забезпечення. Однією з основних цілей тестування білих скриньок є перевірка робочого потоку програми. Він включає тестування низки заздалегідь визначених входів щодо очікуваних або

бажаних результатів, та у випадках, коли конкретний вхід не призводить до очікуваного результату, тестування вказує на помилку.

Ми проводимо тестування на усіх трьох рівнях розробки. Одним з найважливіших тестів у нашій системі є модульне тестування (Unit testing). Це зазвичай перший тип тестування, який проводиться в додатку. Тестування одиниць проводиться у кожному блоці в процесі його розробки. Програма умовно поділяється на модулі – найменші одиниці, які підпадають тестуванню. У випадку тестування об'єктно-орієнтованих блоків програми ми приймаємо метод класу як відповідний модуль (unit). Після створення кожного такого методу, ми перевіряємо його, щоб переконатися, що він працює очікуваним чином на основі тестових вхідних даних. Лише у випадку успішного тесту ми переходимо до наступного етапу розробки.

Для тестування системи API, реалізованої через методи класів, ми використовуємо застосунок POSTMAN. POSTMAN дозволяє задати шлях до тестового модуля, передати набір параметрів та відповідних їм тестових значень й показати відповідь від тестового модуля. Таким способом ми реалізуємо модульне тестування. Одиницею, модулем тестування ми вважаємо метод із системи API. Це дозволяє протестувати ці одиниці на різних наборах тестових даних та переконатися, що взаємодія фронтенд та бекенд частин нашої програмної дає очікуваний результат.

2.7 Рекомендації по використанню та впровадженню програмного засобу «ARMlect»

Автоматизоване робоче місце викладача “ARMlect” призначене в першу чергу для використання у закладах вищої освіти. Система покликана спростити та автоматизувати робочий процес викладача. Більшість функцій розроблені саме для цієї мети, хоча система також може використовуватись іншими учасниками навчального процесу, зокрема і студентами. Для них також передбачений ряд функцій, в основному інформативних.

Під час використання продукту викладачі мають доступ до різних типових функцій пов'язаних з їхньою діяльністю, наприклад:

- читати загальнодоступну інформацію про портал;
- розміщувати та редагувати інформацію про себе на порталі;
- переглядати інформацію про студентів інших викладачів;
- створювати академічні групи студентів;
- створювати формалізовані сторінки для предметів навчальної програми;
- додавати заняття та їх опис;
- переглядати, вносити дані про відвідуваність та успішність у електронний журнал.

Під час використання продукту студенти мають доступ до певного набору функцій, зокрема:

- читати загальнодоступну інформацію про портал;
- розміщувати та редагувати інформацію про себе на порталі;
- переглядати інформацію про студентів інших викладачів;
- переглядати дані про відвідуваність та успішність у електронному журналі.

Доступ до системи мають і незареєстровані користувачі, гості системи. Проте набір їх привілеїв обмежений. Вони можуть лише читати загальнодоступну інформацію про портал. Для отримання розширених доступів та повноважень такі користувачі повинні пройти реєстрацію.

У електронному журналі системи ведеться облік інформації про студентів та показники, якими характеризується їх навчальний процес. Зокрема, там можна отримати інформацію про учасників академічних груп, їх відвідуваність та успішність.

Незважаючи на те, що система початково орієнтована на автоматизацію роботи викладачів закладів вищої освіти, її можливо адаптувати до особливостей роботи навчальних закладів різної форми власності та ступенів. Наприклад, її можна використовувати у школах, коледжах та на приватних курсах.

Для цього потрібно змінити особливості подання інформації та відповідним чином змінити назви категорій користувачів, також від системи можна програмним чином від'єднати одну з доступних функцій, якщо вона не

відповідає набору необхідних функцій такого закладу. Можливо зробити і зворотню дію та додати функції, які необхідні для нового закладу. Для цього треба програмним чином внести відповідні доповнення в інтерфейс продукту, а також додати в наявну систему API новий елемент, який відповідатиме за виконання такого завдання.

Після адаптації продукту до вимог конкретного закладу можна починати роботу з ним. На початковому етапі важливо призначити працівника закладу на ще одну важливу роль у системі — роль адміністратора. Адміністратор має розширені повноваження та певним чином контролює роботу системи та проводить моніторинг її користувачів. Він відповідає зокрема за те, щоб:

- підтверджувати реєстрацію викладачів;
- видаляти користувачів системи, які порушують правила користування нею.

Система написана за допомогою вебтехнологій, тому для досягнення максимальної ефективності використання її найбільш доцільно впроваджувати та розгорнути та базі повноцінного вебсервера. Це дозволяє забезпечити роботу у мережі Інтернет без прив'язки до особистого робочого місця викладача або локальної мережі навчального закладу. Таким чином можна організувати загальнодоступну систему автоматизованих робочих місць для масиву викладачів навчального закладу, а студенти можуть у будь-який час використовувати її як інформаційний ресурс. Також, за потреби таку систему можна впровадити і у більш вузькому масштабі, наприклад для потреб одиничного робочого місця викладача або ж локальної мережі відповідного навчального закладу. У будь-якому варіанті вибору масштабу роботи мережі необхідно розгорнути її на вебсервері.

Оптимальним рішенням є використання стеку технологій LEMP на базі групи операційних систем Linux, наприклад на основі ОС Ubuntu. До складу входять популярний вебсервер Nginx, PHP та підтримка баз даних MySQL.

Для розгортання проекту необхідно обрати кореневу цільову папку на серверній машині, розпакувати туди файли проекту та налаштувати конфігураційний файл сервера. Приклад такого мінімально достатнього файлу конфігурації віртуального хоста вебсервера виглядає наступним чином:

```
server {
```

```

listen 80;
server_name social www.social www.social.com;
root /var/www/social;
index index.php index.htm index.html;
location / {
    try_files $uri $uri/ =404;
}
location ~ /\.php$ {
    include snippets/fastcgi-php.conf;
    fastcgi_pass unix:/var/run/php/php7.0-fpm.sock;
}
location ~ /\.ht {
    deny all;
}
}, де

```

`server_name social www.social www.social.com` – адреса сайту,
`root /var/www/social` — розташування кореневої папки системи,
`fastcgi_pass unix:/var/run/php/php8.0-fpm.sock` — тут варто вказати версію PHP,
яка використовується на сервері. Для оптимальної роботи системи варто
використовувати версію PHP від 8.2 та вище.

За схожим принципом можна розгорнути проект на інших популярних
вебсерверних стеках, зокрема LAMP, WAMP, XAMPP. Використання серверів
WAMP та XAMPP дозволяє розгорнути систему на базі операційних систем
Windows.

В додатку С подано інструкція користувачу.

ВИСНОВКИ

У роботі розглянуто питання автоматизації робочого процесу викладача. Основною метою роботи є реалізація онлайн робочого місця викладача.

Для досягнення поставленої мети було описано особливості робочого процесу викладача на основі актуальних теоретичних досліджень, проаналізовано поточні підходи до вирішення проблеми реалізації онлайн робочого місця викладача, здійснено огляд та аналіз аналогічних програмних розробок, сформульовано вимоги до програмного продукту, обрано модель розробки та визначено структуру програмного засобу, обрано середовище та технології розробки, розроблено онлайн робоче місце викладача за допомогою обраних інструментальних засобів, виконано тестування розробленого програмного засобу та сформульовано вимоги до його впровадження та використання.

В результаті було створено онлайн робоче місце викладача «ARMlect», за допомогою якого можна вносити, отримувати та редагувати інформацію стосовно предметів та занять навчальної програми, працювати з даними про викладачів та студентів, вести облік та контроль навчального процесу, вносити та редагувати дані про успішність та відвідуваність студентів у електронний журнал та виконувати інші функції дотичні до роботи викладача вищого навчального закладу.

Створений програмний продукт дозволяє в онлайн режимі, шляхом постійного та безперебійного доступу до системи за допомогою браузера, реалізувати механізми реєстрації та авторизації, автентифікації, групування користувачів та розмежування прав, збереження формалізованої інформації у базах даних та, за допомогою цих механізмів, реалізувати основні функції робочого процесу викладача. Система адаптована для ефективного використання пристроями на основі різних ОС та з різною роздільною здатністю екрана, забезпечує безпеку та обмеження доступу до даних користувачів, характеризується низьким порогом входу та легкістю роботи з основними функціями.

Розроблений програмний продукт необхідно ввести в експлуатацію для використання у закладах вищої освіти з урахуванням особливостей їх функціонування. Такий підхід дозволить виявити потенційні помилки, неточності та ризики, які не були помічені під час етапів розробки та тестування.

В результаті впровадження такої системи автоматизації робочого процесу викладача можна підвищити його ефективність, надати викладачам можливість віддаленого доступу до керування навчальним процесом, оперативного отримання необхідної інформації, пов'язаної з предметами навчальної програми та студентами. Це підвищить прозорість і достовірність оцінки діяльності всіх учасників навчального процесу та полегшить роботу викладачів закладів вищої освіти.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Киселев А.М. Автоматизация оценки деятельности преподавателей и кафедр как инструмент управления качеством в ВУЗе. URL: <http://edu.znate.ru/docs/14/index-187435.html> (дата звернення 02.12.2020).
2. Сахнюк П.В. Автоматизация рабочего процессу викладача ВНЗ. *Вітчизняна наука на зламі епох. Проблеми та перспективи розвитку*: матеріали всеукр. наук.-практичної інтернет конф. (Переяслав, 15 грудня 2020 р.). Переяслав, 2020. №66. С. 170–172.
3. Кудайберген Е. М. Автоматизированное рабочее место преподавателя. URL: https://storage.tusur.ru/files/176/КСУП%20-1011_%20Автоматизированное рабочее место преподавателя.pdf (дата звернення 05.10.2020).
4. Автоматизированное рабочее место преподавателя: стаття. URL: <https://cyberleninka.ru/article/n/avtomatizirovannoe-rabochee-mesto-prepodavatelya-vuza/viewer> (дата звернення 19.10.2020).
5. Ларин М. Г. Автоматизированное рабочее место преподавателя. *Использование программных продуктов ИС в учебных заведениях*: матеріали конф. Саранск, 2011. С. 54–57.
6. Иванов А. Л. Автоматизированное рабочее место секретаря агентства “Репетитор”: дипломна робота. Ижевск: ИЖГУ. 2006. С. 7–10.
7. Балычева Н. Г. Автоматизированное рабочее место учителя начальных классов. URL: <https://core.ac.uk/download/pdf/132416842.pdf> (дата звернення 19.10.2020).
8. Автоматизированные рабочие места и их оснащение информационными средствами: реферат. URL: <https://works.doklad.ru/view/gMVuj1QoNus.html> (дата звернення 11.10.2020).
9. Автоматизоване робоче місце державного службовця. URL: <https://sites.google.com/site/avtomatizovanerobocemisce/home/avtomatizovane-roboce-misce-derzavnogo-sluzbovca> (дата звернення 4.12.2020).
10. Сорока А. В. Автоматизоване робоче місце завідувача кабінету інформатики. URL:

- https://dspace.vspu.edu.ua/bitstream/handle/123456789/5839/ZNP_%20%D0%9C%D0%A4%D0%A2%D0%9E%202019-60-63.pdf?sequence=1&isAllowed=y
(дата звернення 18.10.2020).
11. Кузьміна Н.М., Струтинська О.В. Автоматизоване робоче місце майбутнього вчителя економіки : курсова робота. URL: <https://sj.npu.edu.ua/index.php/kosn/article/view/248/221> (дата звернення 28.10.2020).
 12. Кантар І. Л. Автоматизовані робочі місця управлінського апарату. URL: <http://pulib.if.ua/referat/view/5160> (дата звернення 01.12.2020).
 13. Конончук Н. Використання табличного процесора для створення АРМів: реферат. Київ: КНЛУ. 2012. С. 3–4.
 14. Заблуждение о том, что АРМ - это ПК. URL: <http://nemtsov.ners.ru/articles/zabluzhdenie-o-tom-chto-arm---eto-pk.html>
(дата звернення 4.10.2020).
 15. Застосування інформаційних технологій для автоматизації діяльності викладача вишу / П. М. Гроза та ін. *Системи обробки інформації*. 2016. №139. С. 209–211.
 16. Инкрементальная (инкрементная, поступательная) модель разработки. URL: https://eoppearhiiv.edu.ee/e-kursused/eucip/arendus_vk/1222_____.html
(дата звернення 5.10.2020).
 17. Глушань В.М, Афанасьев А.Ю., Лященко Н.И. Концепция создания автоматизированного рабочего места (арм) преподавателя. URL: <http://rudocs.exdat.com/docs/index-342688.html> (дата звернення 03.11.2020).
 18. Концепція створення автоматизованого робочого місця (АРМ) викладача: курсова робота. URL: <http://studcon.org/konceptsiya-stvorennuya-avtomatyzovanogo-robochogo-miscya-arm-vykladacha> (дата звернення 13.10.2020).
 19. Обзор 4 бесплатных систем электронного обучения: Moodle, Atutor, Pias, Diskurs. URL: <https://lmslist.ru/free-sdo/> (дата звернення 4.9.2020).
 20. Ромашко С. М. Опорний конспект лекцій з дисципліни «Інформаційні системи в менеджменті» : конспект лекцій. Львів : ЛІМ, 2007. 49 с.

21. Трохимчук Н. А. Організація автоматизованих робочих місць (арм) референта офісу. URL:
https://knowledge.allbest.ru/programming/3c0a65635a3ac79a4d53b89421316d37_0.html.
22. Мирончук Н. М. Особливості праці викладача вищого навчального закладу в контексті самоорганізації його професійної діяльності. *Проблеми освіти*. Київ, 2015. Вип. 85. С. 147–151. – про роботу викладача
23. Харламов И.Ф. Педагогика: навч. посіб. Минск: Университетское, 2001. 272 с.
24. Понятие и принципы моделирования построения АРМ, исследование состава и структуры АРМ : курсова робота. URL:
<https://works.doklad.ru/view/4BChqQBD8b0/all.html>
(дата звернення 10.10.2020).
25. Резник С. Д. Преподаватель вуза: технологии и организация деятельности: навчальний посібник. Пенза: ПГУАС. 2014. 356 с.
26. Артюхина А. И., Чумаков В. И., Болучевская В. В. Преподаватель профессионального образования: навч.-метод. посіб. / за ред. Артюхина А.И. Волгоград: Изд-во ВолгГМУ. 2017. 109 с.
27. Сахнюк П.В. Реалізація онлайн робочого місця викладача. *Актуальні проблеми природничих і гуманітарних наук у дослідженнях молодих учених «Родзинка – 2020»*: матеріали ХХІІ всеукр. наук.-практ. конф. молодих вчених (Черкаси, 23-24 квіт. 2020 р.). Черкаси, 2020. С. 407–408.
28. Мирончук Н. М. Роль планування в забезпеченні ефективності професійної діяльності викладача вищого навчального закладу. *Проблеми освіти*. Київ, 2015. – Вип. 85. С. 82–88.
29. Сайт підтримки програмного комплексу «Ефективна школа–ХХІ». URL:
<http://www.smit.com.ua/school/> (дата звернення 5.10.2020).
30. Морозов М. Н. Системы искусственного интеллекта: курс лекций. Йошкар-Ола: Изд-во ПГТУ. 2017. С. 135–137.
31. Ганашок А. І. Створення автоматизованого робочого місця вчителя інформатики засобами офісних програм. *Інформаційні технології і засоби навчання*. Київ, 2012. №32. С. 70–82.

32. Горбунов А. А. Створення автоматизованого робочого місця для лаборанта кафедри. *Інформаційні технології в освіті та науці*: матеріали всеукр. наук.-практ. конф. (Мелітополь, 21 січня. 2018 р.). Мелітополь, 2020. URL: https://www.researchgate.net/publication/339363066_Stvorennia_avtomatizovanogo_robocogo_misca_dla_laboranta_kafedri (дата звернення 1.12.2020).
33. Структура педагогічної діяльності. URL: <http://psychology.univer.kharkov.ua/dist2020/materialy/Shvedova/PedM.pdf> (дата звернення 4.9.2020).
34. Савенко И. И. Технология разработки программного обеспечения: конспект лекцій. Томск : Изд-во Томского политехнического университета, 2014. 64 с. URL: https://portal.tpu.ru/SHARED/i/IGSAVENKO/academic/Tab/Tab3/trpo_lections_230100_2014.pdf (дата звернення 5.12.2020).
35. Дідовська М. В. Технології проектування програмного забезпечення URL: http://mmsa.kpi.ua/sites/default/files/disciplines/Розробка%20і%20тестування%20програм/didkovska_m_v_testing_lecture_1.pdf (дата звернення 3.11.2020).
36. Fielding R. T. Architectural Styles and the Design of Network-based Software Architectures: dissertation. Irvine: University of California. 2000. P. 76–106. URL: https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm (дата звернення 26.11.2020).
37. Basics of PhpMyAdmin. URL: <https://www.geeksforgeeks.org/basics-of-phpmyadmin/> (дата звернення 14.11.2020).
38. Official website of phpMyAdmin. URL: <https://www.phpmyadmin.net/> (дата звернення 22.11.2020).
39. PHP Code Editor and Debugger – Features | PHPStorm. URL: https://www.jetbrains.com/phpstorm/features/php_code_editor.html (дата звернення 12.11.2020).

ДОДАТКИ

Додаток А

Технічне завдання

Вступ

Повна назва програмного забезпечення: «Онлайн робоче місце викладача ARMIect».

Програмний засіб призначений для використання у вищих навчальних закладах, школах, коледжах та інших закладах освіти для використання викладачами та вчителями таких закладів. Розробка покликана автоматизувати та спростити робочий процес таких користувачів, частково звільнивши їх від виконання повсякденних обов'язків та пропонує зручні механізми для ведення електронного журналу. Програмний засіб корисний не тільки для викладачів, а й для студентів, для котрих реалізовано ряд інформативних функцій.

Розроблений застосунок володіє функціоналом, схожим на функціонал Moodle та Google Classroom, а також інших аналогічних до цих системи. Особливістю розробки є те, що більшість функцій доступні саме викладачам, а не здобувачам освіти.

Підстави для розробки

Робота ведеться на підставі виконання кваліфікаційної роботи «Розробка електронного кабінету викладача».

Призначення розробки

Призначення даної розробки полягає в автоматизації та спрощенні робочого процесу викладачів, реалізує механізми виконання типових для них функцій, а також може слугувати як інформативний ресурс для студентів.

Експлуатаційне призначення

Програма повинна експлуатуватися в будь-яких вищих навчальних закладах, які шукають способи реалізації АРМ викладача та забезпечення його функціонування в онлайн режимі. Також систему можна адаптувати для використання у інших навчальних закладах різної форми власності та рівня

акредитації. Для оптимального функціонування програмного засобу, у закладі освіти повинні бути встановлені мережевий та веб сервери або використовуватись зовнішній хостинг. Кінцевим користувачем даної програми повинен викладач закладу освіти, а також студенти, що там навчаються та гості системи.

Вимоги до програмного продукту

«Онлайн робоче місце викладача ARMIect» повинне розроблюватись з використанням вебтехнологій та бути доступ до використання у мережі Інтернет у вигляді вебсайту. Розробка повинна містити функції для контролю навчального процесу у формі реалізації АРМ викладача.

Вимоги до функціональних характеристик

Програмний засіб повинен забезпечувати виконання наступних функціональних вимог:

- реалізація механізму реєстрації та авторизації користувачів, автентифікації в процесі роботи з системою;
- збереження формалізованої інформації у базі даних;
- реалізація основних функцій робочого процесу викладача;
- реалізація груп користувачів та розмежування прав;
- контроль доступу до інформації та функцій;
- читати загальнодоступну інформацію про портал розміщувати та редагувати інформацію про себе на порталі;
- створювати формалізовані сторінки для предметів навчальної програми
- додавати заняття та їх опис;
- переглядати, вносити дані про відвідуваність та успішність у електронний журнал.

Вимоги до надійності

У програмі повинен бути передбачений контроль введення невірних даних при аварійних ситуаціях. Дії користувача повинні бути контрольованими. Вибір середовища розробки залишити на розробника.

Умови експлуатації

Для адміністрування даної програми або системи повинна бути мінімум 1 людина, що має навички в роботі з комп'ютером, досвід роботи з подібними програмами та мінімальні знання з вебпрограмування. Для використання цієї програми користувачами та для повних її функцій повинно бути не менше 2-х комп'ютерів. Для роботи з програмою користувачу необхідно зареєструватись та увійти в систему. Пропускна здатність мережі повинна мати відклик не більше 90 мс для конкретної роботи систем та запобіганню повільного завантаження сторінок програми.

Вимоги до інформаційної та програмної сумісності

Серверна частина програмної розробки повинна підтримувати повноцінний режим розгортання та роботи у популярних дистрибутивах на основі Linux, зокрема в Ubuntu, операційних системами групи Windows, починаючи з Windows 7, та операційних системах MacOS. Для розробки програмного забезпечення повинно використовуватись середовище вебпрограмування з підтримкою перевірки синтаксису мов PHP та JavaScript. Код програми повинен бути відкритим та доступним для перегляду. На серверній машині повинні бути встановлені вебсервер (Nginx, Apache або ін.), система баз даних, система управління базами даних, реалізована підтримка PHP, бібліотеки Bootstrap 5, jquery.

Вимоги до програмної документації

Для розуміння цієї програми користувачами, розробниками програми повинна бути розроблена та впроваджена програмна документація у наступному складі:

- інструкція з початку роботи;
- опис роботи програми;
- відомості про розробників.

Стадії та етапи розробки

Розробка програми ведеться в послідовності, поданій в табл. А.1.

Таблиця А.1

Етапи розробки програми		
Етап	Зміст роботи	Результат
1	Розробка технічного завдання формування основних алгоритмів вирішення проблеми.	Формування основних вимог та вказівок на розробку даного ПЗ
2	Формування основних алгоритмів вирішення проблеми у вигляді блок-схем.	Формування уяви про майбутній проект, його структури.
3	Добір доцільного середовища програмування.	Вибране середовище розробки, яке відповідає вимогам, які описані в технічному завданні.
4	Реалізація проекту	Формування більш-менш готового програмного засобу який містить функції, описані в технічному завданні.
5	Тестування та налагодження програмного засобу	Знаходження та виправлення помилок, які не були помічені під час розробки програми.
6	Впровадження та супровід	Формування документації та відправлення програми в експлуатацію замовникам.

Порядок контролю і приймання

Програмний продукт має бути повністю перевірений на наявність помилок та дефектів за допомогою таких тестів:

- інсталяційне тестування;
- тестування сумісності;
- тестування білої скриньки;
- смоук тестування.

Програмний продукт повинен бути доставлений замовникам і встановлений уповноваженим фахівцем для запобігання виникненню можливих помилок при встановленні. Фахівець також зобов'язаний провести навчання замовника функціям програми та перед установкою перевірити відповідність системних вимог на пристрої, у який ведеться встановлення програмного продукту.

ДОДАТОК Б

Функція отримання списку студентів

```

function getAllStudents(){
    $query = "SELECT `id`, `firstname`, `lastname` FROM `users` WHERE
`role`=1";
    $stmt = $this->conn->prepare($query);
    // Виконуємо запит
    // Якщо виконання успішне, то інформація про користувача буде
збережена в базі даних
    if($stmt->execute()) {
        $num = $stmt->rowCount();
        if($num>0){
            $group = array();
            $user_ids = array();
            while($row = $stmt->fetch(PDO::FETCH_ASSOC)){
                $user_ids[] = $row["id"];
                $group[] = ['id'=>$row["id"], 'firstname'=>$row["firstname"],
'lastname'=>$row["lastname"], 'group_id'=>"];
            }
        }else{return false;}
    }
    $query = "SELECT `user_id`, `group_id` FROM `users_and_groups`
WHERE `user_id` IN (".implode(',', $user_ids).")";
    $stmt = $this->conn->prepare($query);

    if($stmt->execute()) {
        $num = $stmt->rowCount();
        if($num>0){
            $students_groups =array();

```

```

$count_grouped_users = 0;
while($row = $stmt->fetch(PDO::FETCH_ASSOC)){
    foreach($group as $key=>$value){
        if($row['user_id']==$value['id']){
            $count_grouped_users++;
            $group_id = $row['group_id'];
            $query = "SELECT `title` FROM `groups` WHERE
`id`=:group_id";

            $stm = $this->conn->prepare($query);
            $stm->bindParam(':group_id', $group_id);
            $stm->execute();
            $row_with_title = $stm->fetch(PDO::FETCH_ASSOC);
            $num = $stmt->rowCount();
            if($num>0){
                $row = $stmt->fetch(PDO::FETCH_ASSOC);
                $group_title = $row["title"];

                }else{$group_title = "};
            $group[$key]['group_id']=$row_with_title['title'];
        }
    }
}
$group['count_grouped_users']=$count_grouped_users;
return $group;
}else{return false;}
}
return false;
}

```

Функція отримання фото профілю користувача

```

function getAvatar(){
    // запит, щоб перевірити, чи існує електронна пошта

```



```

$query = "SELECT filename
FROM avatar
WHERE id = :id";

$stmt = $this->conn->prepare($query);

// ін'єкція
$this->id=htmlspecialchars(strip_tags($this->id));

// прив'язуємо значення
$stmt->bindParam(':id', $this->id);

// Виконуємо запит
// Якщо виконання успішне, то інформація про користувача буде
збережена в базі даних
if($stmt->execute()) {
    $num = $stmt->rowCount();
    if($num>0){
        $row = $stmt->fetch(PDO::FETCH_ASSOC);
        $image = $row["filename"];
        return $image;
    }else{return false;}
}
// вернемо 'false', якщо адрес електронної пошти не існує в базі даних
}

```

Функція перевірки на унікальність реєстрації електронної адреси

```

function emailExists(){
    // запит, щоб перевірити, чи існує електронна пошта
    $query = "SELECT id, firstname, lastname, password
FROM " . $this->table_name . "

```

```
WHERE email = ?  
LIMIT 0,1";
```

```
// підготовка запиту
```

```
$stmt = $this->conn->prepare( $query );
```

```
// ін'єкція
```

```
$this->email=htmlspecialchars(strip_tags($this->email));
```

```
// прив'язуємо значення e-mail
```

```
$stmt->bindParam(1, $this->email);
```

```
// виконуємо запит
```

```
$stmt->execute();
```

```
// отримуємо кількість рядків
```

```
$num = $stmt->rowCount();
```

```
// якщо електронна пошта існує,
```

```
// присвоїмо значення властивостям об'єкта для легкого доступу і  
використання для php сесій
```

```
if($num>0) {
```

```
    // отримуємо значення
```

```
    $row = $stmt->fetch(PDO::FETCH_ASSOC);
```

```
    // присвоїм значення властивостям об'єкта
```

```
    $this->id = $row['id'];
```

```
    $this->firstname = $row['firstname'];
```

```
    $this->lastname = $row['lastname'];
```

```
    $this->password = $row['password'];
```

```
// вернем 'true', якщо в базі даних існує електронна пошта */
```

```
return true;
```

```
}
```

```
// вернем 'false', якщо в базі даних не існує електронна пошта
```

```
return false;
```

```
}
```

Інструкція користувачу

«ARMIect» це вебпродукт з архітектурою клієнт-сервер. Клієнтська сторона є інтерактивною, що забезпечує зручну навігацію для користувачів. На головній сторінці користувачеві потрібно створити обліковий запис, пройшовши процес реєстрації. Для цього необхідно вказати електронну пошту та секретний PIN-код або пароль. Також можна додати зображення профілю (за бажанням).

The screenshot shows a web application interface. At the top, there is a dark navigation bar (Navbar) with links for 'Homepage', 'Account', 'entrance', and 'registration'. Below the navbar, a light gray box contains a 'Welcome!' message and a notification: 'You are a guest. You will not be able to access the home page and account pages if you are not signed in.' A blue link 'You can sign up here ->' is provided. Below this, another dark navigation bar contains links for 'Account', 'entrance', and 'registration'. The main content area is titled 'registration' and contains four input fields: 'Name', 'Surname', 'Email', and 'Password'. A blue 'Register now' button is located at the bottom of the form.

Після підтвердження електронного листа користувачеві потрібно буде натиснути кнопку з позначкою "вхід", щоб отримати доступ до форм облікових даних та пароля, які система вимагає для автентифікації ідентифікації користувача перед наданням доступу, натиснувши кнопку "Надіслати" увійдіть'.

[Account](#) [entrance](#) [registration](#)


entrance

Email address


Password

Користувач може вибрати редагування або зміну певної інформації після доступу до сторінки профілю.

[Navbar](#) [Homepage](#) [Account](#) [Log Out](#)



John Doe
jdoee@gmail.com




My Groups

KNIT-53 KNIT-63

An experienced educator, passionate about delivering a word of wisdom to the Computer Science students.

Це може бути ім'я, пароль, електронна пошта, зображення профілю користувача. Наступні фрагменти показують рамку зображення профілю та сторінку профілю за замовчуванням:

Account Output



vasya3@coder.com

About me, my custom user text ...

Homepage Account Output

Account update

Name

Surname

Email

Password

Here you can add / change an avatar

 No file chosen

Інші інтерактивні функції та інтерфейси включають, кнопки для подання файлів, вікторини та кнопки виходу. Як інструмент для зворотного зв'язку викладачів та студентів, продукт простий і зручний у використанні і навіть інтегрується в існуючу систему. Її можуть розгорнути та отримати до неї доступ усі основні вебпереглядачі.

Surname	First name	Email address	Тестові запитання	1. Проектування комбінацій...	2. Проектування комбінацій...	3. Мультиплексор. Демульт...	4
Бондар Олег		olegobond2@gmail.com	-	-	-	-	-
Парфенюк Сергій		parfenyuk.sergij2018@vnu.edu.ua	-	-	-	-	-
Петрук Діана		petruk.diana2018@vnu.edu.ua	-	-	-	-	2.00
Романюк Тетяна		romanyuk.tetyana2018@vnu.edu.ua	-	-	2.00	-	-
Сидорчук Денис		sydorchuk.denys2018@vnu.edu.ua	-	-	-	-	0.00
Трофімук Артур		trofimuk.artur2018@vnu.edu.ua	-	-	-	-	3.00

Приклад інтерфейсу журналу

Анотація

Сахнюк В. Є. – Розробка електронного кабінету викладача. – Рукопис.

Магістерська робота за напрямом підготовки 122 Комп'ютерні науки. – Волинський національний університет імені Лесі Українки, Луцьк. – 2024 р.

Розглянуто основні питання, що стосуються процесу створення автоматизованого робочого місця викладача у мережі Інтернет. В результаті було розроблено бекенд частину програмного засобу «ARMlect» з використанням мови програмування PHP та інших допоміжних технологій, в якому реалізовано різні механізми, покликані спростити робочий процес викладача.

За допомогою системи можна виконувати інформаційні операції стосовно предметів та занять навчальної програми, працювати з даними про викладачів та студентів, вести облік та контроль навчального процесу, вносити та редагувати дані про успішність та відвідуваність студентів у електронний журнал та виконувати інші функції дотичні до роботи викладача вищого навчального закладу.

Розроблений програмний продукт дозволяє в онлайн режимі реалізувати механізми реєстрації та авторизації, автентифікації, групування користувачів та розмежування прав, збереження формалізованої інформації у базах даних та, за допомогою цих механізмів, реалізувати основні функції робочого процесу викладача. Система адаптована для ефективного використання пристроями на основі різних ОС та з різною роздільною здатністю екрана, забезпечує безпеку та обмеження доступу до даних користувачів, характеризується низьким порогом входу та легкістю роботи з основними функціями.

В результаті впровадження даної системи автоматизації робочого процесу викладача можна ефективність навчального процесу, прозорість і достовірність оцінки діяльності всіх учасників навчального процесу та полегшить роботу викладачів закладів вищої освіти.

Ключові слова: автоматизоване робоче місце, технології бекенд розробки, модель розробки, викладач.

Abstract

Sakhniuk V. E. – Development of a teacher's electronic office. – Manuscript.

Master thesis in the field of study 122 – Computer Science. – Lesya Ukrainka Volyn National University, Lutsk. – 2024.

The paper examines the key issues related to the creation of an automated teacher's workplace on the Internet. As a result, the backend part of the "ARMlect" software was developed using the PHP programming language and other auxiliary technologies, which implements various mechanisms designed to simplify the teacher's workflow.

The system allows performing information operations related to subjects and classes of the curriculum, working with data on teachers and students, keeping records and monitoring the educational process, entering and editing data on students' progress and attendance in an electronic journal, and performing other functions related to the work of a higher education institution teacher.

The developed software product allows implementing online registration and authorization mechanisms, authentication, user grouping and access control, storing formalized information in databases, and, using these mechanisms, implementing the main functions of the teacher's workflow. The system is adapted for efficient use by devices based on different OS and with different screen resolutions, ensures security and access restriction to user data, is characterized by a low entry threshold and ease of working with basic functions.

As a result of implementing this system for automating the teacher's workflow, it is possible to increase the efficiency of the educational process, the transparency and reliability of assessing the activities of all participants in the educational process, and to facilitate the work of teachers in higher education institutions.

Keywords: automated workplace, model of development, back-end development technologies, lecturer.