

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ХЕРСОНСЬКИЙ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

**Факультет комп'ютерних наук, фізики та математики  
Кафедра інформатики, програмної інженерії та економічної  
кібернетики**

**МОДЕЛЮВАННЯ ТА РОЗРОБЛЕННЯ ПРОГРАМНОГО  
ЗАБЕСПЕЧЕННЯ ДИСТАНЦІЙНОГО ТЕСТУВАННЯ**

**Кваліфікаційна робота (проект)**

на здобуття ступеня вищої освіти «бакалавр»

Виконав: студент 4 курсу

Спеціальності 121 Інженерія програмного  
забезпечення

Освітньо-професійної програми «Інженерія  
програмного забезпечення» першого  
(бакалаврського) рівня освіти

Туркова Альона Вячеславівна

Керівники: кандидат фізико-математичних наук,  
доцент Кравцов Геннадій Михайлович

Рецензент: кандидат фізико-математичних наук,  
доцент Кузьмич Валерій Іванович

Херсон – 2020

## ЗМІСТ

<b>ВСТУП.....</b>	<b>4</b>
<b>РОЗДІЛ 1 Моделювання систем дистанційного тестування.....</b>	<b>6</b>
1.1 Специфікації стандартів IMS і SCORM для системи дистанційного тестування. Типи тестів і питань.....	6
1.2 Аналіз аналогів системи дистанційного тестування Moodle .....	17
1.3 Вимоги до моделі дистанційного тестування.....	21
1.4 Моделювання інтерфейсу Web-орієнтованої системи дистанційного тестування.....	22
<b>РОЗДІЛ 2 Проектування та розроблення Web-орієнтованої системи дистанційного тестування .....</b>	<b>26</b>
2.1 Огляд технологій, які будуть використані при розробленні Web-орієнтованої системи .....	26
2.2 Проектування програмного забезпечення системи тестування.....	36
<b>ВИСНОВОК .....</b>	<b>43</b>
<b>СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....</b>	<b>44</b>
<b>ДОДАТКИ.....</b>	<b>47</b>
Додаток А.....	47

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

AJAX	Asyncchronous JavaScript and XML
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
IMS	Information Management System
JS	JavaScript
SCORM	Sharble Content Object Reference Model
VS Code	Visual Studio Code
WEB	World Wide Web
XML	eXtensible Markup Language

## ВСТУП

У сучасній освітній системі існує безліч методів контролю отриманих знань. Одним з основних методів контролю знань є тестування.

Тестування як інструмент дистанційного контролю знань особливо добре себе показує під час екстрених перевірок повноти засвоєння студентами понять і окремих тем. У такі моменти тестування показує критичні результати і підвищує якість навчального процесу. Тестові завдання повинні охоплювати весь спектр дидактичних одиниць в освітній програмі. Даний метод дозволяє типізувати вимоги студентів для засвоєння дисциплін.

Дистанційне тестування – це перевірка отриманих знань студента на відстані, яке здійснюється за допомогою сучасних комп'ютерних і телекомунікаційних технологій у реальному часі.

Зараз існує багато систем дистанційного тестування, але в кожній з них є свої недоліки, найвагоміший з них – це платна система. Виходячи з цього є актуальним розробити свою безкоштовну та легку у використанні систему дистанційного тестування для студентів нашого університету з використанням нових технологій, які суттєво скорочують процес розробки додатку та добре впливають на швидкість та працездатність програми.

**Актуальність:** безкоштовний додаток для проходження тестувань для всіх учасників навчального процесу з використанням нових технологій.

**Об'єктом дослідження** є системи дистанційного навчання.

**Предметом дослідження** є проектування та розроблення Web-орієнтованої системи тестування.

**Метою дослідження** є створення Web-орієнтованої системи дистанційного тестування для учасників навчального процесу.

**Завдання дослідження:**

- Моделювання системи дистанційного тестування.
- Вимоги до моделі дистанційного тестування.
- Проектування та розроблення Web-орієнтованої системи дистанційного тестування.

**Структура досліджень:** Дипломна робота складається зі вступу, списку умовних скорочень, двох розділів, висновків, списку використаних джерел та додатків.

## **РОЗДІЛ 1**

### **Моделювання систем дистанційного тестування**

#### **1.1 Специфікації стандартів IMS і SCORM для системи дистанційного тестування. Типи тестів і питань.**

Дистанційне навчання – це дуже високий рівень розвитку заочної форми навчання в інформаційному суспільстві. Цей рівень відображає сучасну філософію освіти, реалії постіндустріальної культури, які стали причиною зміни парадигми віддаленого навчання в умовах інформатизації суспільства і глобалізації освітніх процесів.

Вищі навчальні заклади є основним джерелом у розвитку дистанційної форми навчання і продовжують її розвивати в Україні, тому що є такі переваги:

- позитивний вплив на вирішення комплексу соціально-економічних проблем (подолання безробіття, злочинності, наркоманії, тощо) засобами дистанційно-віртуальної освіти;
- ліквідація відставання периферійних районів від їх центрів шляхом отримання вільного доступу до освіти та залучення до великих об'ємів інформації та культурних досягнень людської цивілізації;
- об'єднання інформаційного, інтелектуального, творчого та науково-педагогічного потенціалу всього світового співтовариства в інтересах людини;
- організація інклюзивного та неперервного навчання, навчання засуджених та дорослого населення, здійснення підвищення кваліфікації без відриву від виробництва;
- підняття рівня освіти до високотехнологічного рівня розвинених держав, інтеграція у європейський освітній простір.

## IMS

**IMS Global Learning Consortium** – міжнародна некомерційна організація, основним завдання якої є забезпечення поширення використання сучасних навчальних технологій в освітній сфері. Основою діяльністю IMS є розробка стандартів у сфері ДО.

Проект IMS Global Learning Consortium розробляє та допомагає розповсюдженню відкритих специфікацій для підтримки діяльності в рамках розподіленого навчання, таких як розміщення та використання освітніх матеріалів, спостереження за прогресом об'єкта навчання, складання звітів про його успішність та обмін записами про учнів між адміністративними системами.

У IMS є дві основні мети: визначення технічних стандартів для забезпечення інтероперабельності прикладних програм та послуг у неперервному навчанні й підтримка включення специфікацій IMS у продукти та послуги по всьому світу. IMS робить зусилля для загального визнання специфікацій, що дасть змогу працювати разом, тобто інтерперувати розподіленим навчальним середовищам та матеріалам багатьох авторів.

Проект IMS народився 1997 року в рамках ініціативи EDUCAUSE «National Learning Infrastructure Initiative». Консорціум почав роботу, акцентуючи увагу на вищій освіті, однак специфікації, що публікуються сьогодні, як і поточні проекти, охоплюють вимоги в широкому діапазоні навчальних контекстів, включаючи середню школу, корпоративне та державне навчання. Сфера застосувань специфікацій загалом визначається як «life-long learning», включає як «он-лайн», так і «офф-лайн» оточення, що відбувається і синхронно, і асинхронно.

Це означає, що навчальні контексти, які ефективно використовують специфікації IMS, включають Internet-орієнтовані середовища (такі як системи управління курсами на основі Web), а також навчальні сценарії, котрі включають «офф-лайн» електронні ресурси (такі як навчальні ресурси на CD-ROM). Учні можуть перебувати в традиційному навчальному середовищі (шкільна класна кімната, університетська аудиторія), в корпоративному або державному оточенні, вдома.

IMS виконав величезний обсяг роботи: було зібрано вимоги за допомогою спеціалізованих груп та інших джерел у всьому світі, щоб встановити критичні аспекти інтеперабельності на освітніх ринках. Усю проаналізовану та апробовану інформацію включено до специфікацій IMS, які є вільнодоступними для усіх користувачів. Опубліковані специфікації були подані на розгляд авторитетним організаціям у сфері стандартизації для гарантії їх відповідності міжнародним вимогам та визнання. Проект об'єднує сучасні досягнення розробок у дистанційній освіті й специфікує їх у форматі XML. Багато фірм і державних організацій, які займаються дистанційною освітою, прийняли цей стандарт, і на сьогодні він позиціонується як формат обміну даними між освітніми організаціями.

Консорціумом було створено пакет стандартів у шести областях: профілі, метадані, зміст, тести, керування та компетентність. Інформаційна модель управління (IMS Enterprise Information Model) описує структури даних, які специфікують взаємодію систем навчання з використанням Internet та систем, що обслуговують реальний освітній заклад.

Основними класами прикладних програм, які підтримуються цією моделлю, є системи:

- управління процесом навчання;



- адміністрування студентів;
- бібліотеки;
- управління людськими ресурсами.

Ця модель головним чином специфікує взаємодію з системами, які знаходяться всередині одного закладу. Цей стандарт не специфікує обмін даними між різними закладами.

Інформаційна модель управління підтримує чотири процеси, котрі зазвичай потрібні для взаємодії систем навчання з системами управління навчальними закладами:

- Збереження персональних даних - це дані, які зберігаються в системах адміністрування та управління реального освітнього закладу та необхідні електронним системам навчання.
- Управління групами - управління та збереження даних про групи студентів.
- Управління реєстрацією - це операції над такими видами даних, як призначення викладачів курсам або призначення інструктора для проведення заняття.
- Обробка кінцевих результатів - цей процес полягає в обробці та збереженні результатів тестування груп.

Найвідомішими стандартами IMS на сьогоднішній день є:

- **IMS Question&test Interoperability Specification** – містить у собі вимоги до XML-даних, які використовуються для організації обміну

навчальними матеріалами, призначеними для тестування учнів або студентів і оцінки результатів тестування.

- **IMS Learning Resource Meta-data Specification** – містить вимоги до опису метаданих навчальних матеріалів.
- **IMS Content Packing Specification** – містить вимоги до компонування змісту навчального матеріалу.
- **IMS Learner Information Packing** – містить вимоги до формування даних про студента.

Вища освіта є провідним напрямком досліджень і розробок в області навчання і технологій навчання, але підтримка в системі освіти К-12 зростає. Корпоративні і державні навчальні організації також постійно беруть участь у створенні нових підходів до навчання та підтримки продуктивності. Рада директорів IMS GLC заявляє, що прагне підтримувати безперервне навчання у всіх сегментах.

Робота, яку роблять IMS GLC і її члени, критично важлива, тому що вона руйнує розрізненість всередині системи освіти, а також між викладачами та постачальниками продуктів навчальних технологій і видавцями. Визначаючи стандарти взаємодії, IMS GLC також дозволяє викладачам створювати інноваційні рішення, що підтримують навчання. Це, в свою чергу, надає приклади і додатки, які можна використовувати в багатьох контекстах навчання.[24]

## **SCORM**

Еталонна модель об'єкта спільно використовуваного контенту (SCORM) відноситься до мови програмування курсів електронного навчання, який був створений в 1999 році як стандартна еталонна структура для індустрії електронного навчання. Це формат, в якому будь-

який контент, незалежно від системи, за допомогою якої він був створений (хмарне програмне забезпечення або індивідуальне програмування), сумісний з усіма навчальними платформами. І тому (в цьому його головна сила) контентом можна легко ділитися.

Перша версія стандарту SCORM (SCORM 1.0) з'явилася в січні 2000 року. У документі вперше чітко було визначено поняття SCO, як навчального об'єкта, який взаємодіє з системою дистанційного навчання. Також в стандарті версії SCORM 1.0 було визначено спосіб обміну даними між SCO і системою дистанційного навчання - через API, що надається СДО.

У 2001 році виходить стандарт SCORM версії 1.1. У стандарті SCORM версії 1.1 були більш чітко сформульовані правила взаємодії SCO і системи дистанційного навчання. У тому ж році виходить наступна версія стандарту SCORM - 1.2. У версії стандарту SCORM 1.2 були сформульовані правила "упаковки" навчальних матеріалів. Дана версія стандарту SCORM стала базовою на довгий час і по суті залишається такою і по сьогоднішній день.

Поточна версія стандарту SCORM: SCORM 2004 року (4-я редакція). В даний час найбільшого поширення все ще має стандарт SCORM версії 1.2. Незважаючи на те, що стандарт SCORM останньої версії отримує все більшу популярність, його використовує не значне число користувачів дистанційного навчання.

Поняття SCORM є акронімом від Sharable Content Object Reference Model, що означає: еталонна модель контенту для совісного використання.

SCORM визначає технічну основу для середовища навчання, побудованої з використанням Web-технологій. SCORM об'єднує безліч

взаємопов'язаних технічних вимог, стандартів і нормативів. SCORM описує модель агрегації змісту (Content Aggregation Model) і середу виконання (Run-Time Environment) для навчальних об'єктів, забезпечуючи адаптивне навчання, засноване на навчальних цілях, пріоритетах, продуктивності та інших факторах. SCORM також описує модель послідовності і навігації (Sequencing and Navigation) для динамічного відображення вмісту в залежності від потреб учнів.

SCORM прагне пов'язати воедино внесок різних груп і інтересів в освітньому співтоваристві. SCORM призначений для координації нових технологій і можливостей в рамках комерційних і державних ініціатив.

SCORM базується на таких основних принципах:

- система дистанційного навчання повинна використовувати навчальний контент, розроблений з використанням засобів розробки, створених різними виробниками;
- системи дистанційного навчання, створені різними виробниками, повинні мати можливість використовувати один і той же навчальний контент;
- системи дистанційного навчання повинні мати можливість звернення до спільного сховища навчального контенту і використовувати навчальний контент, який там зберігається.

В рамках системи дистанційного навчання SCORM фокусується на інтерфейсі між навчальним контентом і системою дистанційного навчання. SCORM не розглядає питання пов'язані з функціональними можливостями систем дистанційного навчання. З точки зору SCORM система дистанційного навчання є інформаційною системою, яка надає кошти для управління і надання навчального контенту учнем. Іншими словами, в

SCORM, система дистанційного навчання визначає, що надати тому, якого навчають, і коли, відстежує процес роботи учня з навчальним контентом.

SCORM - дійсно потужний інструмент для всіх, хто бере участь в онлайн-навчанні. За допомогою SCORM LMS контент можна створювати один раз і використовувати в багатьох різних системах і ситуаціях без змін. Функція plug-and-play може бути потужною всередині організації, але тим більше для всіх організацій. Контент можна продавати і доставляти користувачеві швидше, ефективніше і за нижчою ціною.[25]

### **Типи тестів і питань**

Виділяють наступні форми тестових завдань:

- завдання з вибором єдиної правильної відповіді;
- завдання з декількома можливими правильними відповідями;
- завдання, що мають дві відповіді: один з яких правильний, в торую неправильний;
- завдання на завершення пропозицій; питання на встановлення взаємозв'язків;
- завдання з короткими відповідями. [4]

Узагальнюючи дослідження з різновидів тестів, В.С. Аванесов підкреслює: «форма тестових завдань - це спосіб організації, упорядкування та існування змісту тесту».[6]

Більш докладно описані різні форми завдань у Аванесова В.С., Майорова А.М., М.Б. Челишкова, С.К. Калдибаева [6; 7; 8; 9; 10], посилаючись на зарубіжну літературу пропонує таку класифікацію тестових завдань:

- завдання закритої форми (з безліччю вибором), в яких учні вибирає правильну відповідь з даного;
- завдання на додаток (відкриті завдання) вимагає самостійного отримання відповідей;
- завдання на встановлення відповідності (з множинним вибором), виконання яких пов'язане з виявленням відповідності між елементами кількох множин;
- завдання на встановлення правильної послідовності, в яких від учня вимагає вказати порядок дій або процесів, перерахованих педагогами.

**Завдання закритої форми.** Під тестовим завданням закритої форми розуміють таке тестове завдання, де є готові відповіді, з яких тестований повинен вибрати.

У закритій формі тестових завдань можна виділити кілька видів:

- тестові завдання з вибором однієї правильної відповіді;
- тестові завдання з вибором кількох правильних відповідей.

Під тестовим завданням з вибором однієї правильної відповіді розуміють тестові завдання закритої форми, в якому серед запропонованих відповідей лише один правильний.

У завданнях з вибором однієї правильної відповіді можна виділити кілька підвидів. В.С. Аванесов [5] виділяє наступне:

- завдання з двома варіантами відповідей;
- завдання з трьома варіантами відповідей;
- завдання з чотирма видами відповідей;
- завдання з п'ятьма видами відповідей.

Завдання такої форми більше за інших поширені в тестовій практиці. Деякі автори саме їх називають тестами. Причина такого широкого поширення цієї форми тестових завдань кілька: порівняльна простота виконання, традиційність, зручність для швидкого та автоматизованого контролю знань.

При складанні тестових завдань найбільш істотним є підбір варіантів відповідей, які називаються неправильними, які розташовані поруч з правильними. Вважається, що ці варіанти повинні бути "правдоподібними". Неправильні, але правдоподібні відповіді називається дистракторами [9] (від англ. To distract- відволікати). Якщо в завданні є  $k$  відповідей, то тоді відповідь, частка вибору якого близька до значення  $1/k$ -ідеальний дистрактор; частка вибору менше, ніж  $1/k$ - домінуючий дистрактор в завданні.

Сучасна практика показує що, найчастіше можна зустріти завдання з однією правильною відповіддю, в яких пропонується 4 або 5 можливих відповідей. Наприклад, при використанні чотирьох варіантів відповідей ймовірність вгадування правильної відповіді становить 25%, а при п'яти відповідях -20%.

**Завдання з множинним вибором (з вибором кілька правильних відповідей).** Під тестовим завданням з вибором кількох правильних відповідей розуміють тестове завдання закритої форми в якому допускається вибір кількох правильних відповідей з числа пропозицій.

Ефективність використання цього виду завдання підвищується якщо:

- використовується серія завдань;
- формулювання завдання чітка і зрозуміла;

- завдання не призначені для виявлення високого рівня, засвоєння комплексу елементів;
- час виконання обмежено і його досить мало.

Найчастіше такі тести використовується для перевірки засвоєння базових понять, основних властивостей, елементів змісту.

**Тестові завдання відкритої форми.** Завданнями відкритої форми називають завдання без вказівки можливих варіантів відповіді. Такі завдання називають ще відкритими завданнями або завданнями на додаток. Такі завдання вимагає від учнів самостійно сформулювати відповідь, а не вибрати готовий. Перевагою завдань закритої форми можна вважати те, що вони не допускають можливості вгадування [6].

**Тестові завдання на встановлення відповідності.** Під тестовим завданням на встановлення відповідності розуміють тестове завдання, в якому необхідно встановити відповідність елементів однієї множини елементами іншого. При цьому бажано, щоб кількість елементів в цих множинах було неоднаковим.

**Тестові завдання на встановлення відповідності правильної послідовності.** Завдання на відповідність вимагає встановити відповідність кожного елемента одного безлічі елементами іншої множини. Найбільш цінним є те, що такі завдання дозволяє визначити серед інших асоціативну складову знань, тобто знання взаємозв'язку визначень, і чинників, авторів і творів, і характеристик параметрів різних приладів і пристроїв, зв'язків між законами, формулами і т.д.

Ці завдання розробляються для перевірки виробленості умінь здійснювати послідовність процесів, дій і операцій, виявлення



сформованості конкретних понять. Разом із завданням пропонується у виборі правильних порядків проходження.

## 1.2 Аналіз аналогів системи дистанційного тестування Moodle

Сервіс дистанційного навчання Moodle – це, мабуть, одна з найвідоміших систем, яка призначена для дистанційного навчання у країнах СНГ (Рисунок 1.1).

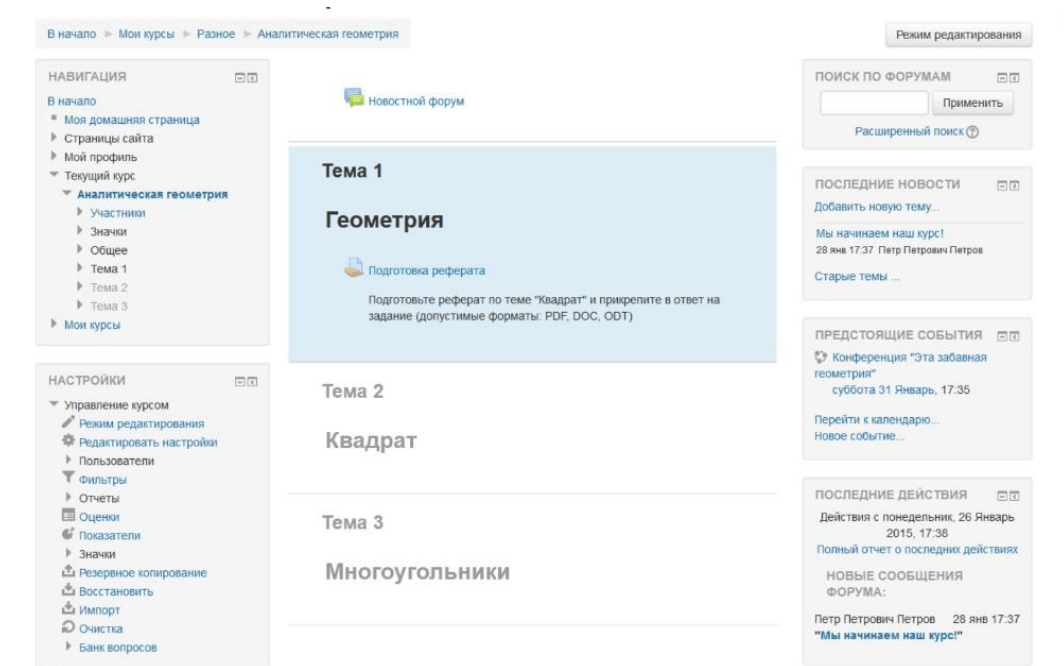


Рисунок 1.1 - Система дистанційного навчання Moodle

Преваги Moodle:

- безкоштовна система, яка вже готова до впровадження.
- з її допомогою можна створити повноцінні якісні курси для навчання на відстані.
- система має досить потужний апарат тестування.

- дає доступ до організації диференційованого навчання.
- підтримує різноманітні освітні стратегії.
- система має настройки варіантів управління доступами до курсів.
- дозволяє відстежувати прогрес учнів.
- можливість публікувати контент різноманітного формату.

### Недоліки Moodle:

- система є безкоштовною, але її необхідно встановити, тобто потрібен хостинг.
- дана система дуже вимоглива до сервера.
- для системи необхідно багато ресурсів, відповідно фінансові витрати.
- систему необхідно досконально вивчити перед тим, як приступити до роботи.

### Edmodo

Edmodo - ця система позиціонує себе як соціальна мережа для навчання (Рисунок 1.2).

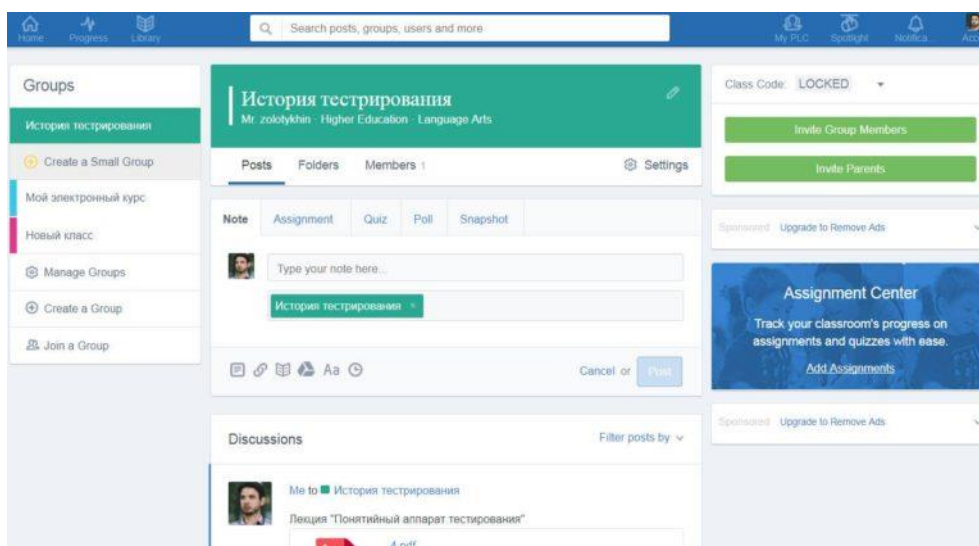


Рисунок 1.2 - Система дистанційного навчання Edmodo

Переваги цієї системи:

- система є безкоштовною.
- немає реклами, яка відволікає від навчання.
- досить легка реєстрація.
- у цій системі користувачі поділяються на три категорії: викладачі, учні та батьки.

Недоліки системи:

- в системі відсутня українська та російська мови, що може посприяти погіршенню сприйняття інформації учнями.
- групи Edmodo не можна поєднати між собою.
- набір навчальних елементів досить малий.

## iSpring Online

iSpring Online – система для організації дистанційного навчання (Рисунок 1.3).

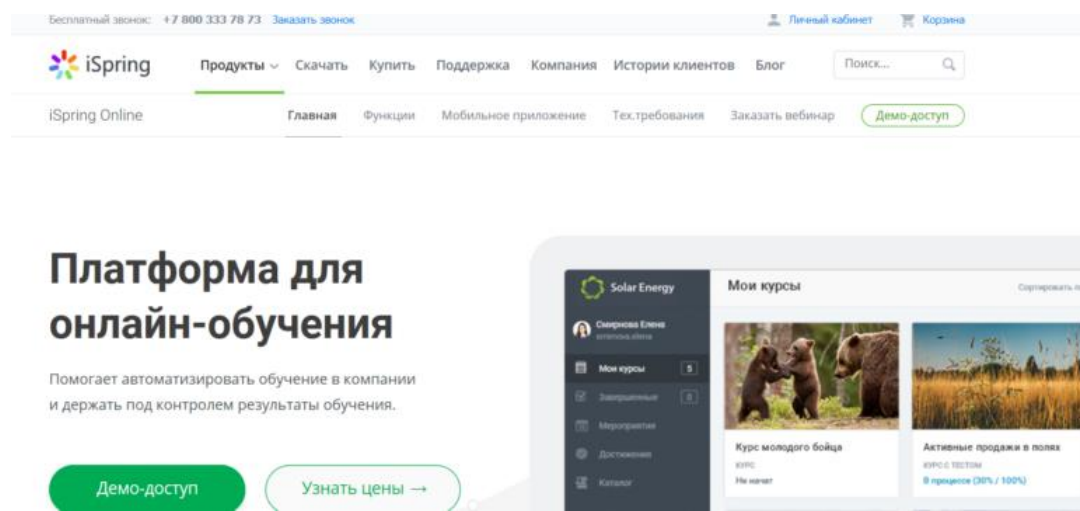


Рисунок 1.3 - Система дистанційного навчання iSpring Online

Переваги цієї системи:

- безлімітне сховище. У систему можна завантажити необмежену кількість навчальних матеріалів.
- навчання через мобільний пристрій.
- повний контроль за успішністю учнів.
- можливість оформити навчальний портал на свій розсуд.
- можна демонструвати робочий стіл, презентацію або відео, писати в загальний і особистий чат.

Недоліки цієї системи:

- iSpring Online - платна система. Вартість навчання одного користувача коштує 97 рублів на місяць.
- мінімальна кількість користувачів – 50.

### **Getcourse.ru**

Getcourse.ru – це платформа для навчання і продажу онлайн-курсів, тому вона більш підходить для репетиторів або тренерів.

Переваги цієї платформи:

- створення сайтів за готовими шаблонами.
- можна налаштувати власноруч візуалізацію навчального порталу.
- можливість налаштування навчання в формі уроків, тренінгів, курсів.
- автоматичні Email-розсилки по базі даних.

Недоліки системи:

- ця система платна.
- технічна підтримка здійснюється переважно через соцмережі.

### 1.3 Вимоги до моделі дистанційного тестування

Вимоги до моделі дистанційного тестування можна поділити на два пункти. Перший пункт – це вимоги до сервісної частини, а другий – це вимоги до клієнтської частини, тобто до інтерфейсу системи дистанційного тестування.

Розглянемо спочатку вимоги до сервісної частини системи:

- Наявність БД, завдяки якій можна буде створювати сутності користувача, розкладу та структур.
- Можливість авторизації для студента та викладача.
- Це повинна бути закрита система, тобто доступ до даних можуть мати лише користувачі, які пройшли авторизацію.
- Рациональне за обсягами збереження інформації.
- При додаванні нових модулів БД не повинна створюватися знову або перероблятися.
- Закріплення змінюваних або замінюваних даних в базі даних, а також ухилення від втрати попередніх даних.
- Можливість додавання нових сутностей у рівнях структур та користувача.
- Захищеність від розповсюдження персональної інформації студентів або викладачів.

Далі треба розглянути вимоги до клієнтської частини (інтерфейсу системи):

- По-перше, це повинен бути зручний та зрозумілий дизайн системи з точки зору UI/UX.
- Функціонал повинен бути мінімальним для легкості у використанні.

- Зрозуміла навігація для адміністратора (викладача) при створенні тестування.
- В інтерфейсі системи має бути прогрес бар, де буде кількість усіх питань у тесті та скільки вже було пройдено.
- Досить важливим пунктом є зручна навігація при проходженні опитування, тобто перехід по запитання.

#### 1.4 Моделювання інтерфейсу Web-орієнтованої системи дистанційного тестування

Виходячи з того, що наша система буде використовуватися на базі університету, я вибрала наступну назву (Рисунок 1.4):



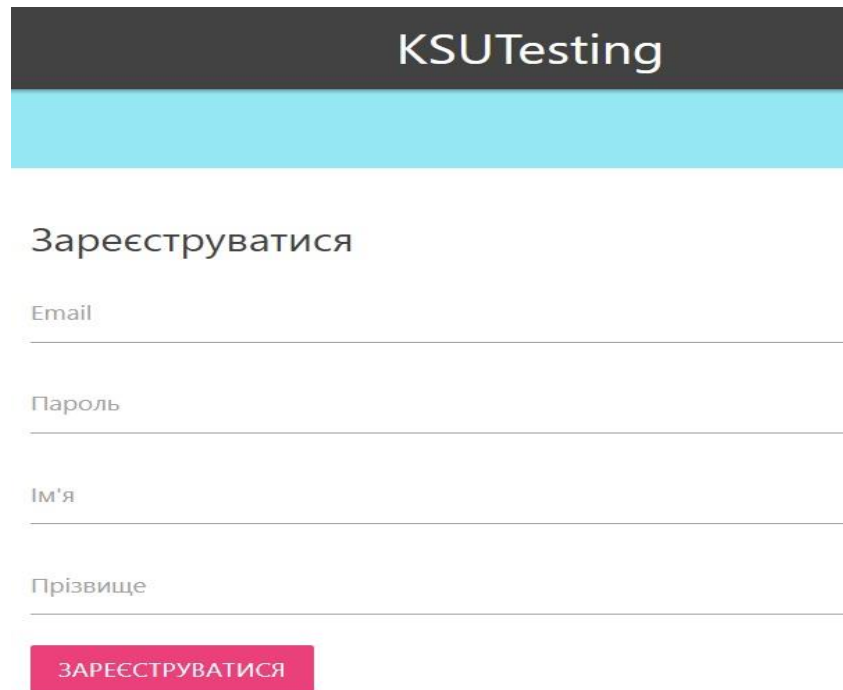
Рисунок 1.4 - Логотип сайту дистанційного тестування

Блок з меню містить у собі наступні кнопки: «Зареєструватися», «Увійти», «Нове опитування» (Рисунок 1.5).



Рисунок 1.5 - Блок з меню

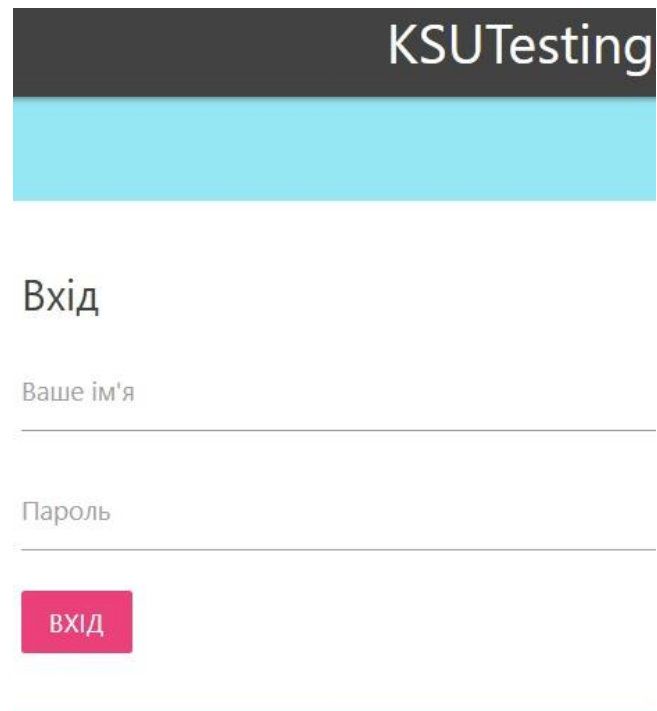
У формі реєстрації є наступні поля для введення: «Email», «Пароль», «Ім'я» та «Прізвище» (Рис. 1.6).



The registration form for KSUTesting features a dark grey header with the text 'KSUTesting' in white. Below the header is a light blue horizontal bar. The main content area is white and contains the title 'Зареєструватися' in a dark grey font. There are four input fields, each with a light grey label to its left: 'Email', 'Пароль', 'Ім'я', and 'Прізвище'. Each label is positioned above a thin horizontal line representing the input field. At the bottom of the form is a pink rectangular button with the text 'ЗАРЕЄСТРУВАТИСЯ' in white, uppercase letters.

Рисунок 1.6 - Форма реєстрації

У формі входу до системи є наступні поля: «Ваше ім'я», «Пароль» (Рисунок 1.7).



The login form for KSUTesting features a dark grey header with the text 'KSUTesting' in white. Below the header is a light blue horizontal bar. The main content area is white and contains the title 'Вхід' in a dark grey font. There are two input fields, each with a light grey label to its left: 'Ваше ім'я' and 'Пароль'. Each label is positioned above a thin horizontal line representing the input field. At the bottom of the form is a pink rectangular button with the text 'ВХІД' in white, uppercase letters. A thin light blue horizontal line is located below the button.

Рисунок 1.7 - Форма «Вхід»

Головна сторінка сайту виглядає так (Рисунок 1.8):

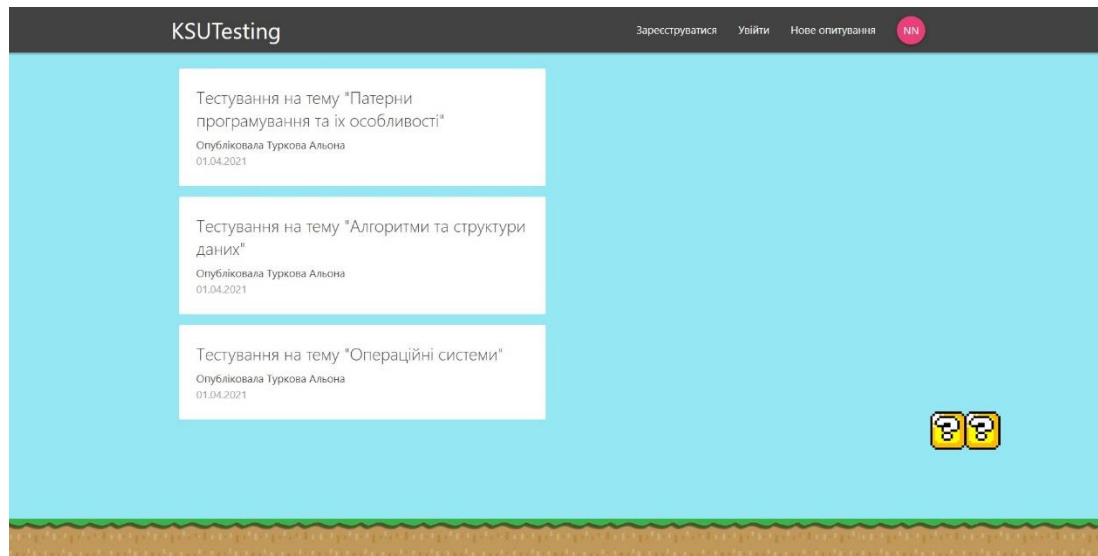


Рисунок 1.8 - Головна сторінка сайту

При проходженні опитування вікно з питаннями з однією правильною відповіддю буде виглядати так (Рисунок 1.9):

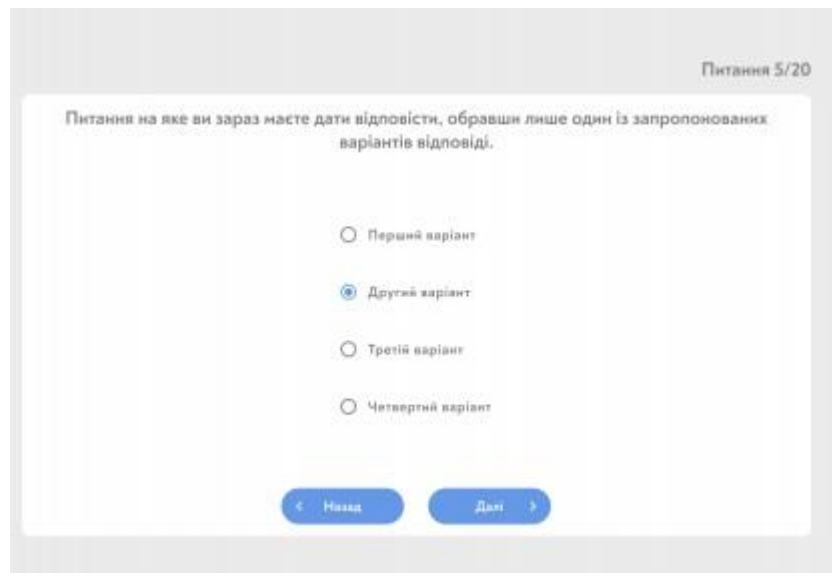
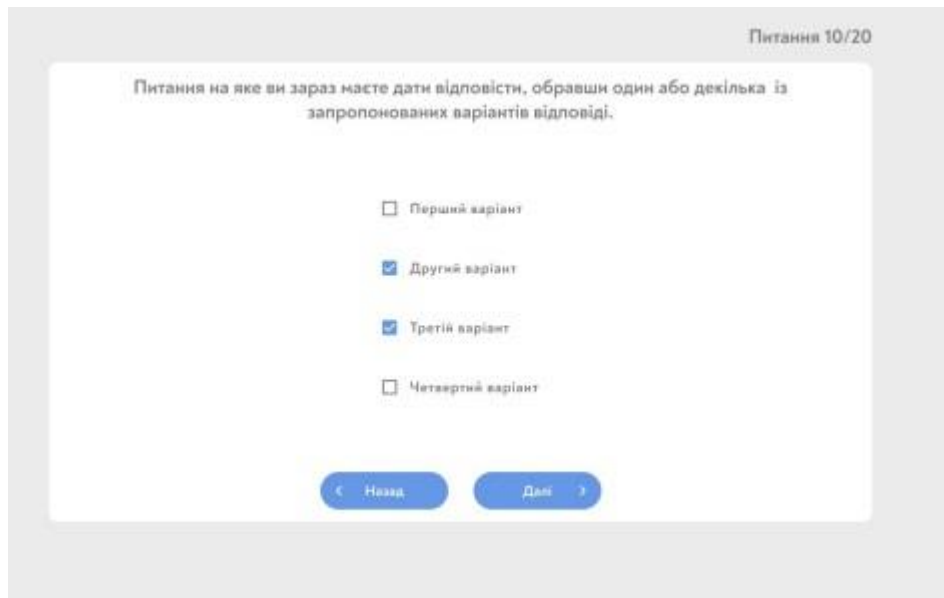


Рисунок 1.9 - Вікно за питаннями з однією правильною відповіддю



Питання з декількома варіантами відповідей ось так (Рисунок 1.10):



Питання 10/20

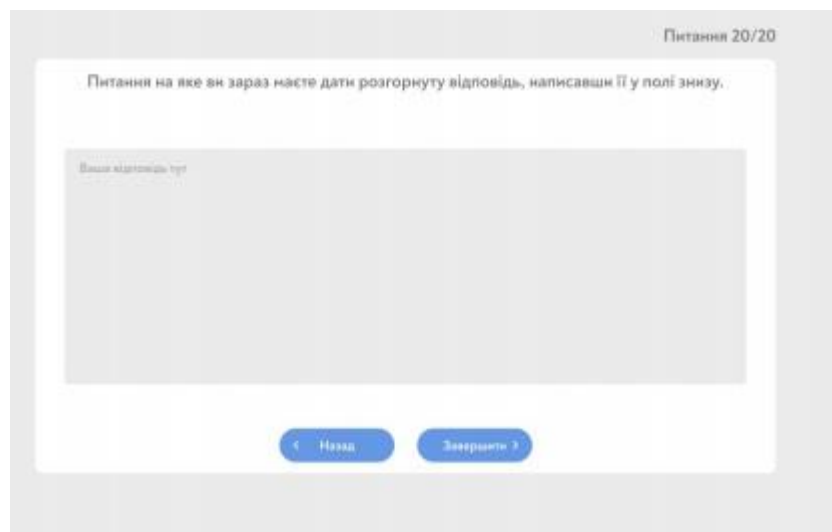
Питання на яке ви зараз маєте дати відповідь, обравши один або декілька із запропонованих варіантів відповіді.

- Перший варіант
- Другий варіант
- Третій варіант
- Четвертий варіант

[← Назад](#) [Далі →](#)

Рисунок 1.10 - Питання з декількома варіантами відповідей

Питання з розгорнутою відповіддю буде виглядати ось так (Рисунок 1.11):



Питання 20/20

Питання на яке ви зараз маєте дати розгорнуту відповідь, написавши її у полі знизу.

Ваша відповідь тут

[← Назад](#) [Закрити →](#)

Рисунок 1.11 - Питання з розгорнутою відповіддю

## РОЗДІЛ 2

### Проектування та розроблення Web-орієнтованої системи дистанційного тестування

#### 2.1 Огляд технологій, які будуть використані при розробленні Web-орієнтованої системи

При розробці проекту будуть використовуватися безліч різноманітних технологій та інструментів, тому є сенс розглянути кожен з них та пояснити за які переваги їх було обрано.

##### **Firestore**

Розробляючи програмний продукт, постає завдання зберігання, накопичення, а також оперативного доступу до інформації, що використовується додатком. Все це здійснюється завдяки такому засобу як база даних. База даних – це інформаційна модель, що дозволяє упорядковано зберігати дані про групу об'єктів, що володіють однаковим набором властивостей (Рисунок 2.1). Перш ніж приступити до створення самої бази даних, розглянемо які ж основні типи баз даних існують на даний момент і які плюси і мінуси у кожній з них.



Рисунок 2.1 - База даних Firebase

Ієрархічна модель (деревоподібна). Ця система являє собою відсортовану сукупність об'єктів типу «Дерево», що включають себе екземпляри типу «Запис» [2]. Для кращого розуміння звернемося до рисунку 2.2.

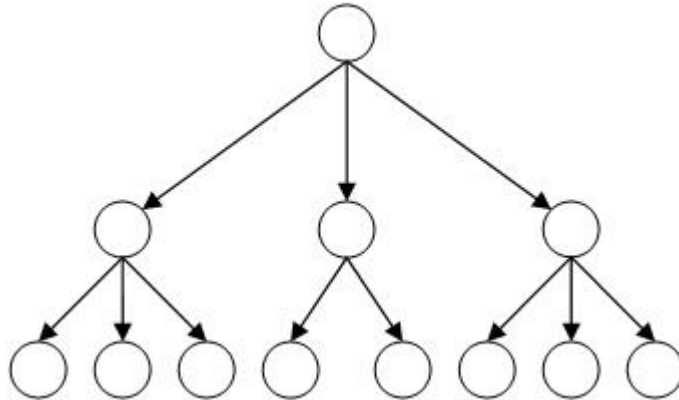


Рисунок 2.2 - Ієрархічна модель

Плюсами використання такої системи є:

1. використання пам'яті обчислювального пристрою максимально ефективно;
2. висока швидкість виконання операцій над даними.

Недоліки:

1. неінтуїтивна система зв'язків;
2. великий розмір такої моделі;
3. робота з такими базами даних вимагає тривалої підготовки, так як
4. їх розуміння роботи досить складно.

Реляційна модель бази даних. Вона являє собою сукупність

даних і складається з набору двовимірних таблиць.

Позитивними властивостями такої моделі є:

1. базою даної моделі є розвинений математичний апарат, який дозволяє точно охарактеризувати основні операції над даними;
2. проста форма подання інформації.

Негативними ж властивостями є:

1. дуже низька швидкість доступу до даних;
2. трудомісткість проектування.

Після розгляду основних існуючих моделей баз даних, встановимо властивості, які ми хочемо отримати під потреби нашого проекту:

- доступність;
- кроссплатформенність;
- інтеграція різних популярних рішень в області розробки БД.

Відповідно до переліченими вище чинниками мною була обрана ієрархічна модель бази даних, реалізована за допомогою інструменту від компанії «Google» під назвою Firebase.

Firebase - це платформа за допомогою якої розробляються, тестуються, розкручуються додатки і гри для Android, iOS і Web [3]. Вона надає потужний сервіс та включає безліч функцій, наприклад таких як MBaaS (Mobile backend as a service), Firebase Auth. Firebase Hosting.

Винахід в тому вигляді, в якому ми бачимо його зараз, з'явилося на ринку інформаційних технологій в 2016 році і вже набуло великої популярності серед розробників.

Так як наша мета це створення бази даних, ми зупинимося на такій функції Firebase як MBaaS. Вона представлена двома такими інструментами як Firebase Realtime Database і Firebase Storage за допомогою яких ми отримуємо можливість працювати з серверними базами даних і хмарою. Варто відзначити важливу характеристику даної функції, що всі оновлення в базах даних відбуваються автоматично.

## **React**

У React.js є все для створення веб та мобільних додатків. Це швидко, безпечно і масштабоване. React.js є домінуючою технологією JavaScript, яка стає все більш популярною. Деякі розробники програмного забезпечення воліють не називати це «фреймворком», оскільки він дає розробникам набагато більше свободи, ніж Angular або Vue.

Так само, як Angular підтримується Google, React.js підтримується Facebook і співтовариством розробників. Обидва з відкритим вихідним кодом і безкоштовне використання під ліцензією MIT. React.js всього шість років, що робить його відносно новою технологією. За допомогою JavaScript ви можете створювати динамічні додатки, в яких браузер виконує значну частину функцій, тому вони можуть працювати без звернення до сервера (Рисунок 2.3).

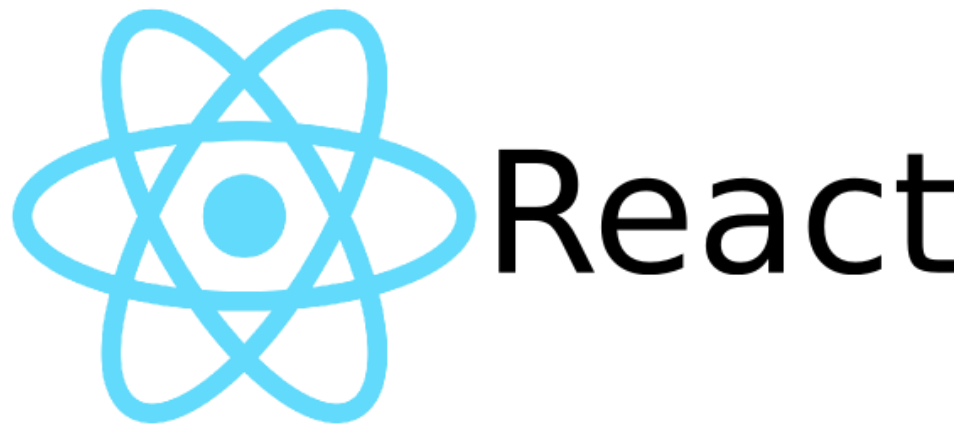


Рисунок 2.3 - React

Це також дозволяє незалежно оновлювати дані і інтерфейс тільки в частині додатка (без перезавантаження). React - це інструмент для створення як компонентів для користувача інтерфейсу, так і для цілих призначених для користувача інтерфейсів - всього, що стосується об'єднання візуальних елементів, прив'язки даних до цих елементів і визначення логіки.

Існує два можливих підходи до використання сучасних JavaScript-фреймворків - рендеринг на стороні клієнта, коли браузер завантажує код і відрисовує призначений для користувача інтерфейс, або рендеринг на стороні сервера, де рендеринг інтерфейсу виконується на бекенді. Різниця між рішеннями JavaScript (такими як React.js) і старими технологіями полягає в тому, що JS бере на себе набагато більше логіки і маніпуляцій з документами, як якщо б він взагалі не відображався на сервері.

Основною особливістю React.js, яка відрізняє його від інших популярних JavaScript-фреймворков, є гнучкість. Ви можете взяти бібліотеку і використовувати її для відображення простої сторінки або подання, але ви також можете комбінувати React.js з іншими

інструментами і використовувати її в якості основи, яка закладе фундамент для складного додатка. Це більше, ніж просто бібліотека, оскільки вона має динамічну екосистему, яка складається з інструментів, бібліотек і підходів.

## **Redux**

Сам по собі React є дуже потужним інструментом для розробки, але так чи інакше він все одно має набір певних недоліків, які не дають розробнику працювати швидко, не ускладнювати свій код, не псувати його структуру. Одним з найважливіших - незручне зберігання даних і стану в javascript-додатку. Даний недолік допомагає виправити бібліотека Redux.

Для початку варто розібратися в тому, як сам React зберігає і передає дані. Дані в React проходять через компоненти, тобто від батьківського компонента, до дочірнім. При цьому React дозволяє брати дані від одного компонента, в інший, але при цьому виникає проблема звана «спагеттікод», дане визначення означає, що код стає вкрай заплутаним, і не завжди зрозуміло звідки дані приходять і куди йдуть. Саме цю проблему і виправляє Redux, основною ідеєю Redux є зберігання даних в одному єдиному сховищі, з якого інші компоненти підтягують дані.

Дані в сховищі відправляють самі компоненти, а інші компоненти «підписуються» на сховище, і в разі зміни або появи нових даних, вони їх отримують з цього сховища. Для прикладу на малюнку 1 представлено зберігання даних в React (зліва) і React з використанням Redux (праворуч) (Рисунок 2.4).

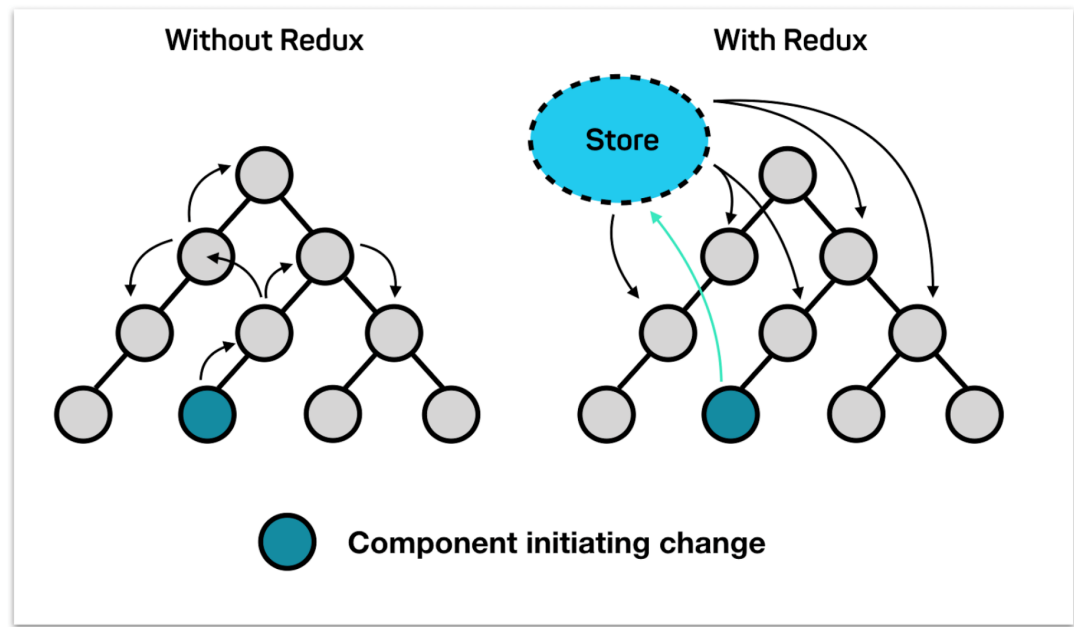


Рисунок 2.4 - Порівняння роботи програми з Redux та без

Виходячи з цього, розробнику стає зрозумілим, куди компоненти відправляють свою інформацію про зміну стану і звідки вони отримують цю інформацію. Концепція використання сховищ, як координатора стану програми називається Flux.

Основні ідеї даної концепції представлені нижче:

1. Redux використовує тільки одне сховище для зберігання відразу всіх станів додатки;
2. Стан доступно для читання, тобто виклик зміни стану це створення нового стану програми;
3. Так як Redux не дозволяє змінювати стан безпосередньо, то спеціальний action описує які зміни необхідно зробити, і потім редьюсери вносять зміни в новий стан додатки.

Головною думкою створення даних сховищ є не створення нових сховищ, а копіювання старих і збереження в них нової інформації,



отриманої від компонента. У великих програмах, одним сховищем обійтися важко, тому в Redux є можливість створювати множинні сховища. У даних видах сховища є кореневої об'єкт, в якому знаходяться інші сховища. У Redux існує початковий стан, яке визначається в сховище, але в разі, якщо користувач оновить сторінку, то всі дані віддані редьюсерамі скинуться, що вкрай небажано на проєкті.

Для виправлення цієї проблеми, можна передати спеціальне стан, зване `createStore ()`, завдяки цьому, можна як би «подорожувати в часі», що дозволяє налагоджувати додаток. В даній статті було розглянуто використання Redux разом з React, була розглянута логіка роботи Redux, виділені його основні особливості і способи використання Redux на реальному проєкті.

Виходячи з наведеної вище інформації, слід, що використання Redux, дозволяє сильно спростити розробку SPA-додатки. Дана бібліотека, що розширює стандартні можливості React, користується великою популярністю серед спільноти розробників, що позитивно впливає на кількість документації на різних мовах.

## **JavaScript**

JavaScript - це особлива мова програмування, особливість якого полягає в тому, що він виконується безпосередньо у вікні браузера. Програми, створені за допомогою JS, так і називаються - скрипти.

Вони не самостійні, а інтегруються в уже готовий HTML код сторінки. Коли така програма-скрипт спрацьовує, на сторінці відбуваються видимі зміни.

Основний принцип дії цієї мови програмування в тому, щоб кожен раз створювати подія у відповідь на дії користувача на сторінці.

Відбувається це так:

1. Користувач щось натиснув на сайті або просто навів курсор миші на потрібний елемент.
2. Браузер зафіксував цю дію і просигналив про його скоєнні.
3. Після цього запускається спеціальний скрипт, призначений для цієї події.
4. Скрипт вносить зміни на сторінці - відкриває нове вікно, підсвічує потрібний елемент, запускає анімацію і так далі.

Програми-скрипти, написані на JS, являють собою просто текст.

Інтерпретатори вбудовані в більшість сучасних браузерів, тому у всіх них і підтримується JavaScript.

Можливості JavaScript у веб-розробці практично необмежені. Все, що ви можете собі уявити на сайті, можна реалізувати за допомогою JS. А в зв'язці з технологією AJAX він взагалі може творити чудеса.

AJAX, якщо спрощено, це технологія, яка дозволяє оновлювати частина інформації на сторінці, без перезавантаження сторінки цілком. Завдяки цьому значно економиться трафік і, отже, збільшується швидкість роботи сайтів.

Якщо проводити аналогії, то HTML створює на сайті розмітку, CSS відповідає за її оформлення, а JavaScript - за «пожвавлення» всіх елементів.

Ось лише деякі можливості JS:

1. додавати всіяку анімацію - від падаючих крапель дощу до автоматично перегортати каталогу;

2. створювати події у відповідь на будь-які дії користувача - клік миші, наведення вказівника на потрібну область, натискання клавіші, скролінг і т.д .;
3. перевіряти на відповідність вимогам, що вводяться користувачем у формі, наприклад, формат номера телефону або електронної пошти;
4. витягувати і аналізувати інформацію про пристрій користувача - назва браузера, тип пристрою і т.п.;
5. створювати і змінювати HTML-теги;
6. вносити зміни в стилі CSS;
7. обробляти дані cookie.

Тепер розглянемо переваги мови програмування Java Script:

1. створювати програми можна в будь-якому текстовому редакторі, включаючи Microsoft Word і навіть Блокнот;
2. не дуже складний синтаксис, який під силу вивчити навіть новачкам (за умови, якщо докласти чималих зусиль);
3. широкі можливості при створенні найрізноманітніших сайтів;
4. JavaScript "читається" будь-яким сучасним браузером - як десктопних, так і мобільних;
5. він постійно вдосконалюється;
6. скрипти легко інтегруються в HTML-код;
7. на ринку праці багато фахівців, які знають JS, також багато вакансій для таких розробників.

## 2.2 Проектування програмного забезпечення системи тестування Visual Studio Code

Для початку роботи треба обрати інтегровану середу розробки, де безпосередньо буде розроблятися проект. Мій вибір пав на Visual Studio Code.

Visual Studio Code - редактор коду з відкритим вихідним кодом, який поширюється безкоштовно.

Основні переваги цього редактора:

- Кросплатформеність.
- Велика кількість розширень.
- Підтримка тем.
- Підтримка HTML / CSS.

Тепер розглянемо кожен пункт окремо.

Кросплатформеність. На відміну від свого «великовагового» родича Visual Studio, що має версію для Windows і версію для macOS, VS Code доступний для Windows, macOS і Linux. Працює в цілому однаково в усіх трьох операційних системах, установка для всіх систем проста і не вимагає особливих навичок.

Велика кількість розширень. VS Code має безліч розширень, які суттєво допомагають при вирішенні задач або проблем.

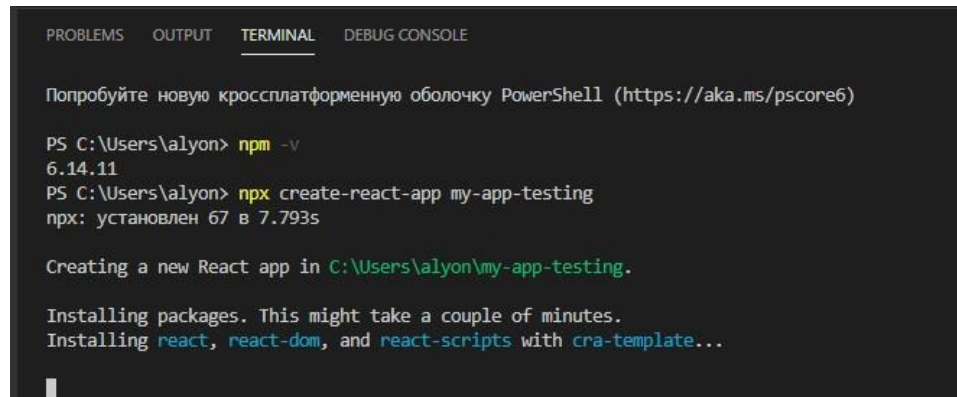
Підтримка тем. За замовчуванням в VS Code встановлена приємна для очей тема «default dark», яку ви могли бачити вище, але є й інші теми, які виглядають лаконічно і не відволікають від роботи. Адже візуал – це, теж, дуже важливий аспект при програмуванні.

Підтримка HTML / CSS. Це важливий пункт, тому що HTML – мова розробки гіпертексту, а CSS – каскадна таблиця стилів.

## Create React App

Create React App - зручне середовище для створення додатка на React. Інструмент налаштовує середу для використання нових можливостей JavaScript, оптимізує додаток для продакшена і забезпечує комфорт під час розробки.

Також при розробці потрібно встановити Node.js на ПК. Для створення проекту виконуємо наступні дії (Рисунок 2.5):



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Попробуйте нову кроссплатформенную оболочку PowerShell (https://aka.ms/powershell)

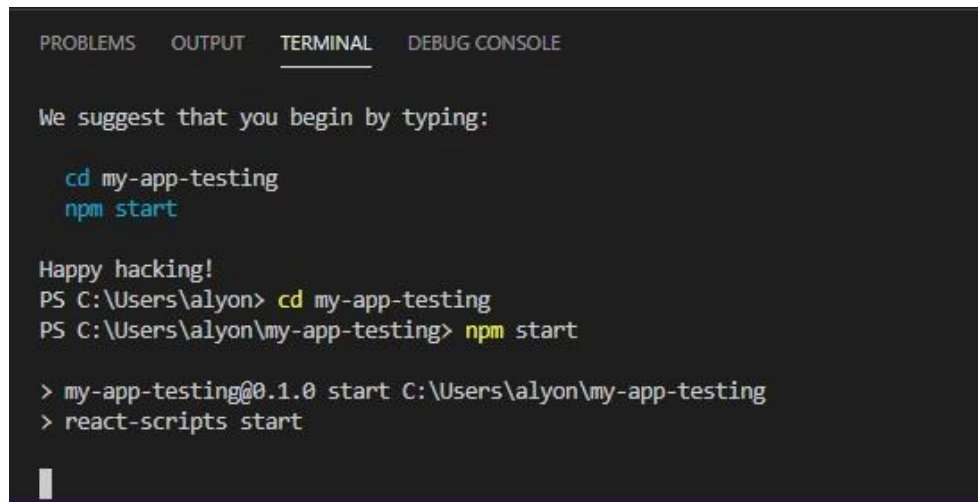
PS C:\Users\alyon> npm -v
6.14.11
PS C:\Users\alyon> npx create-react-app my-app-testing
npx: установлен 67 в 7.793s

Creating a new React app in C:\Users\alyon\my-app-testing.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
```

Рисунок 2.5 - Встановлення Node.js

Даною командою ми встановимо набір інструментів, які допоможуть створити наше React-додаток. По завершенні установки ми зможемо запустити додаток командами (Рисунок 2.6):



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

We suggest that you begin by typing:

  cd my-app-testing
  npm start

Happy hacking!
PS C:\Users\alyon> cd my-app-testing
PS C:\Users\alyon\my-app-testing> npm start

> my-app-testing@0.1.0 start C:\Users\alyon\my-app-testing
> react-scripts start
```

Рисунок 2.6 - Запуск додатка

Новий додаток автоматично відкриється в браузері (Рисунок 2.7):

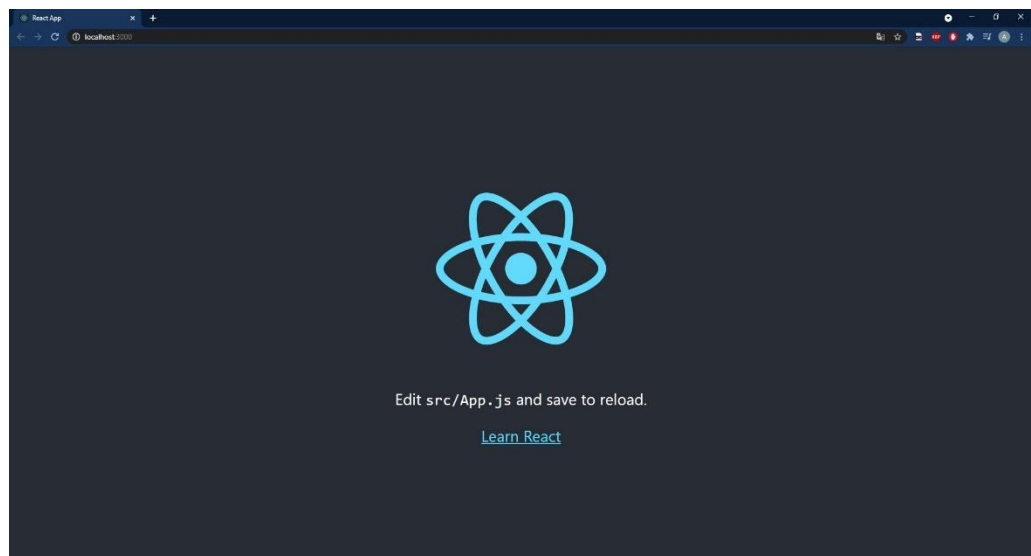


Рисунок 2.7 - Відкритий додаток у браузері

У редакторі коду відкриємо my-app-testing/src/App.js (Рисунок 2.8):

```

src > JS App.js > ...
1  import logo from './logo.svg';
2  import './App.css';
3
4  function App() {
5    return (
6      <div className="App">
7        <header className="App-header">
8          <img src={logo} className="App-logo" alt="logo" />
9          <p>
10             Edit <code>src/App.js</code> and save to reload.
11          </p>
12          <a
13             className="App-link"
14             href="https://reactjs.org"
15             target="_blank"
16             rel="noopener noreferrer"
17           >
18             Learn React
19          </a>
20        </header>
21      </div>
22    );
23  }
24
25  export default App;
26

```

Рисунок 2.8 - Головний файл у проекті App.js

App.js – це головний файл у структурі. Додаток складається з одного компонента, де функція `render()` є його головною складовою.

Далі у папці `src` створюємо папку `components`, де в свою чергу додаємо такі папки: `auth`, `dashboard`, `layout`, `projects`. Далі у кожній папці створюємо наступні js-файли:

- У папці `auth`: `SignIn.js`, `SignUp.js`. Ці файли потрібні для авторизації користувачей.
- У папці `dashboard`: `Dashboard.js`, `Notifications.js`. Для відображення наявних тестів на сайті.
- У папці `layout`: `Navbar.js`, `SignedInLinks.js`, `SignedOutLinkst.js`. Для блоку з меню та кнопками авторизації та вихіду з системи.

- У папці projects: CreateProject.js, ProjectDetails.js, ProjectList.js, ProjectSummary.js. Для створювання та додавання тестів.

## Створення бази даних Firebase

Firebase в основному забезпечує багатофункціональну серверну частину нашої системи, тому не потрібно турбуватися про налаштування сервера, управління їм, створення або написання коду для аутентифікації чи вилучення даних. Також, в Firebase буде розміщуватися наш веб-додаток після реалізації проекту. Це робить настройку динамічного проекту дуже легкою та безкоштовною.

Firebase поставляється з базою даних в реальному часі під назвою Firestore.

Створюємо базу даних під назвою «my-app-testing» (Рисунок 2.9).

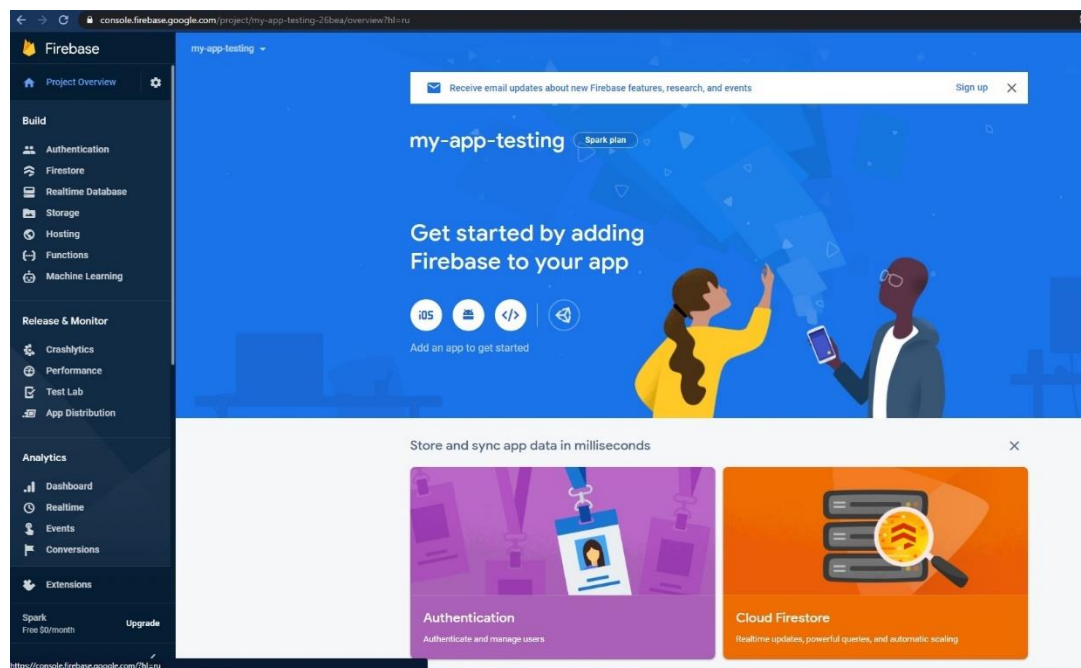
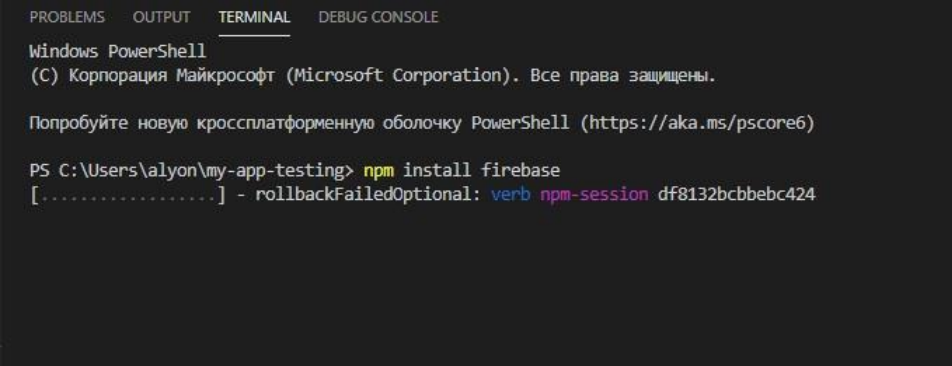


Рисунок 2.9 - Створення бази даних



В панелі управління ми бачимо, що у нас є всі потрібні нам сервіси. Аутентифікація сховища бази даних, функції, хостинг.

Тепер нам потрібно прив'язати базу даних до нашого проекту. Для установки ми будемо використовувати `npm` (Рисунок 2.10).



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/powershell)

PS C:\Users\alyon\my-app-testing> npm install firebase
[.....] - rollbackFailedOptional: verb npm-session df8132bcbbebc424
```

Рисунок 2.10 - Долучення бази даних до проекту

Далі ми створюємо файл `config` у папці `src` для настройки Firebase. Відповідно туди будуть входити всі конфігураційні файли. У цій папці створюємо файл `fbConfig` для конфігурації Firebase. Вставляємо туди наступний код (Рисунок 2.11):

```
1 import firebase from 'firebase/app'
2 import 'firebase/firestore'
3 import 'firebase/auth'
4
5 const config = {
6   apiKey: "AIzaSyDmpjcsr44ADrWspTKieeWteOnAWvfG48",
7   authDomain: "web-testing-e0eba.firebaseio.com",
8   projectId: "web-testing-e0eba",
9   storageBucket: "web-testing-e0eba.appspot.com",
10  messagingSenderId: "252708630540",
11  appId: "1:252708630540:web:2099b09be4bdf5c4b6be29",
12  measurementId: "G-T163TR8CVV"
13 };
14
15 firebase.initializeApp(config);
16 firebase.firestore().settings({ timestampsInSnapshots: true })
17
18 export default firebase;
```

Рисунок 2.11 - Файл fbConfig

Тут створюється об'єкт конфігурації, де є ключ API. Це ідентифікаційний ключ, щоб наш проект міг визначити, який проект Firebase на back end підключити. Далі ми ініціалізуємо додаток Firebase, що дозволяє підключити наш проект до бази даних. Наступне, що потрібно зробити це імпортувати з бібліотеки firebase потрібну для проекта функціональність. Крім того, ініціалізуємо firestore, щоб додати налаштування, в які ми передаємо властивість timestampsInSnapshots: true. В кінці ми експортуємо базу даних.

## ВИСНОВОК

У ході дипломної роботи була поставлена мета створення Web-орієнтованої системи тестування.

Для розроблення проекту були використані нові технології, такі як бібліотеки Java Script – React, Redux та база даних Firebase.

Були вирішені наступні задачі:

- Моделювання системи дистанційного тестування.
- Вимоги до моделі дистанційного тестування.
- Проектування та розроблення Web-орієнтованої системи дистанційного тестування.

Розроблена Web-орієнтована система дистанційного тестування вирішує проблеми з працездатністю та швидкістю відповіді, а також, спрощує роботу з сервісною частиною проекту. Система має перспективи подальшого розвитку та поширенню.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Гарбич-Мошора О., Косовська О., Шілінг А. Стандарт Scorm: короткий огляд. – 2011.
2. Вишнівський В. В. и др. Організація дистанційного навчання. Створення електронних навчальних курсів та електронних тестів. – 2014.
3. Власенко Л. В. Переваги та недоліки дистанційного навчання. – 2013.
4. Мишко С.А. Проблема тестирования в системе образования США. Дисс...к.п.н.-Ужгород, 1982.
5. Аванесов В.С. Научные проблемы тестового контроля знаний.- М.,1994.-135с.
6. Аванесов В.С. Форма тестовых заданий. 2-изд., переработанное и расширенное. М.: «Центр тестирования»,2005.-156с.
7. Майоров А.Н. Теория и практика создания тестов для системы образования .-М.: Народное образование,2000.-352с.
8. Калдыбаев С.К., Ажыбаев Д.М., Бекежанов М.М. Компьютерная диагностика результатов обучения в общеобразовательной школе:Практико-ориентированная монография.-Б:2007.-136с.
9. Чельшкова М.Б. Теория и практика конструирования педагогических тестов.-М.,2001 – 432 с.
- 10.Библиотека "learn-reactjs" / [Электронный ресурс]. - Режим доступа: <https://learn-reactjs.ru/core/topics>.
- 11.Библиотека "old.code.mu" / [Электронный ресурс]. - Режим доступа: <https://old.code.mu/books/advanced/javascript/react>.

- 12.Абдулхаков А.В., Ярошенко А.А., Кумратова А.М.  
Особенности использования Firebase:Статья в сборнике трудов конференции, 2019 – 288-289 с.
- 13.Система управления базами данных[Электронный ресурс]. –  
Режим доступа: [www.infosgs.narofd.ru/31.htm](http://www.infosgs.narofd.ru/31.htm)
- 14.Атеев К.О., Симакина Н.И. Применение информационной среды для проведения занятий в курсе Практической экстремальное программирование // Тенденции развития науки и образования – 2020. - №62. – С. 14-17
- 15.Тихомиров. В. П., Титарев Л. Г., Шевченко К. К.  
Технологические системы в открытом образовании // Збірник матеріалів міжнародної наради «Телематика і безперервна освіта».-К., 2001-С. 168-171
- 16.Печкурова О. М., Глибовець М. М. Синхронна та асинхронна взаємодії в дистанційному навчанні // Збірник матеріалів міжнародної наради «Телематика і безперервна освіта».-К., 2001-С 140-142.
- 17.Ладик О.І. Реалізація навчального матеріалу для дистанційного навчання на базі програмної платформи управління електронним навчанням Moodle / О.І. Ладик, С.О. Осипчук // Наукові записки УНДТЗ. – 2008. – №5(7). – С. 78-87.
- 18.Олійник В.В. Дистанційна освіта за кордоном та в Україні: Стислий аналітичний огляд. – К.: ЦППО, 2001. – 48 с.
- 19.Агапонов С.В., Джалишвили З.О., Кречман Д.Л. и др.  
Средства дистанционного обучения. Методика, технология, инструментарий / Под ред. З.О. Джалишвили – СПб.:БХВ-Петербург, 2003. – 336 с

20. Martin Fowler — GUI Architectures. Часть 2 [Электронный ресурс]. — 2009 — Режим доступа до ресурсу:  
<https://habr.com/post/53536/>.
21. Katty Sierra — Head first Java, 2nd edition [Электронный ресурс]. — 2005. — Режим доступа до ресурсу:  
<https://www.amazon.com/dp/0596009208>.
22. JavaScript [Электронный ресурс] // Вікіпедія. — Режим доступа до ресурсу: <https://ru.wikipedia.org/wiki/JavaScript>.
23. Трояновська Т. І. Розробка протоколу за стандартом SCORM для обміну даними між складовими СКП : дис. — «Наука і студія», 2013.
24. IMS [Электронный ресурс]. — Режим доступа до ресурсу:  
<https://iseazy.com/blog/en/what-is-scorm-advantages-of-using-scorm-with-authoring-tools-such-as-iseazy/>
25. SCORM [Электронный ресурс]. — Режим доступа до ресурсу:  
<https://www.talentlms.com/elearning/what-is-scorm-and-tincan>

## ДОДАТКИ

Додаток 1

КОДЕКС АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ  
ЗДОБУВАЧА ВИЩОЇ ОСВІТИ ХЕРСОНЬСЬКОГО  
ДЕРЖАВНОГО УНІВЕРСИТЕТУ

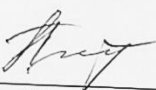
Я, Туркова Альона Вячеславівна, учасниця освітнього процесу Херсонського державного університету, УСВІДОМЛЮЮ, що академічна доброчесність – це фундаментальна етична цінність усієї академічної спільноти світу.

**ЗАЯВЛЯЮ**, що у своїй освітній і науковій діяльності **ЗОБОВ'ЯЗУЮСЯ**:

- дотримуватися:
  - вимог законодавства України та внутрішніх нормативних документів університету, зокрема Статуту Університету;
  - принципів та правил академічної доброчесності;
  - нульової толерантності до академічного плагіату;
  - моральних норм та правил етичної поведінки;
  - толерантного ставлення до інших;
  - дотримуватися високого рівня культури спілкування;
- надавати згоду на:
  - безпосередню перевірку курсових, кваліфікаційних робіт тощо на ознаки наявності академічного плагіату за допомогою спеціалізованих програмних продуктів;
  - оброблення, збереження й розміщення кваліфікаційних робіт у відкритому доступі в інституційному репозитарії;
  - використання робіт для перевірки на ознаки наявності академічного плагіату в інших роботах виключно з метою виявлення можливих ознак академічного плагіату;
- самостійно виконувати навчальні завдання, завдання поточного й підсумкового контролю результатів навчання;
  - надавати достовірну інформацію щодо результатів власної навчальної (наукової, творчої) діяльності, використаних методик досліджень та джерел інформації;
  - не використовувати результати досліджень інших авторів без використання покликань на їхню роботу;
  - своєю діяльністю сприяти збереженню та примноженню традицій університету, формуванню його позитивного іміджу;
  - не чинити правопорушень і не сприяти їхньому скоєнню іншими особами;
  - підтримувати атмосферу довіри, взаємної відповідальності та співпраці в освітньому середовищі;
  - поважати честь, гідність та особисту недоторканність особи, незважаючи на її стать, вік, матеріальний стан, соціальне становище, расову належність, релігійні й політичні переконання;
  - не дискримінувати людей на підставі академічного статусу, а також за національною, расовою, статевою чи іншою належністю;
  - відповідально ставитися до своїх обов'язків, вчасно та сумлінно виконувати необхідні навчальні та науково-дослідницькі завдання;
  - запобігати виникненню у своїй діяльності конфлікту інтересів, зокрема не використовувати службових і родинних зв'язків з метою отримання нечесної переваги в навчальній, науковій і трудовій діяльності;
  - не брати участі в будь-якій діяльності, пов'язаній із обманом, нечесністю, списуванням, фабрикацією;
  - не підроблювати документи;
  - не поширювати неправдиву та компрометуючу інформацію про інших здобувачів вищої освіти, викладачів і співробітників;
  - не отримувати і не пропонувати винагород за несправедливе отримання будь-яких переваг або здійснення впливу на зміну отриманої академічної оцінки;
  - не залякувати й не проявляти агресії та насильства проти інших, сексуальні домагання;
  - не завдавати шкоди матеріальним цінностям, матеріально-технічній базі університету та особистій власності інших студентів та/або працівників;
  - не використовувати без дозволу ректорату (деканату) символіки університету в заходах, не пов'язаних з діяльністю університету;
  - не здійснювати і не заохочувати будь-яких спроб, спрямованих на те, щоб за допомогою нечесних і негідних методів досягати власних корисних цілей;
  - не завдавати загрози власному здоров'ю або безпеці іншим студентам та/або працівникам.

УСВІДОМЛЮЮ, що відповідно до чинного законодавства у разі недотримання Кодексу академічної доброчесності буду нести академічну та/або інші види відповідальності й до мене можуть бути застосовані заходи дисциплінарного характеру за порушення принципів академічної доброчесності.

12.04.2021  
(дата)

  
(підпис)

Альона Туркова  
(ім'я, прізвище)