

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

ГЮІК.1489СТ.017_ПЗ
(позначення документа)

Адаптивна система управління освітнім процесом в навчальних закладах.
Електронний журнал
(тема)

Виконав: студент 2 курсу, групи СКСм-19-1

Роговий Є.О
(прізвище, ініціали)

Спеціальність: 123 – Комп'ютерна інженерія
(код і повна назва спеціальності)

Тип програми: Освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма: Спеціалізовані
комп'ютерні системи

(повна назва спеціалізації)
Керівник проф. Немченко В.П.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

(прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерної _____ інженерії _____ та управління _____

Кафедра _____ Автоматизації _____ проектування _____ обчислювальної техніки _____

Рівень _____ вищої _____ освіти _____ другий
(магістерський) _____

Тип _____ програми _____ Освітньо-професійна _____

Спеціальність _____ 123 – Комп'ютерна інженерія _____

Освітня програма _____

Спеціалізовані комп'ютерні системи _____

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« _____ » _____ 20 ____ р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові _____ Роговому Євгенію Олександровичу _____

(прізвище, ім'я, по батькові)

1. Тема роботи _____ Адаптивна система управління освітнім процесом в навчальних закладах. Електронний журнал _____

затверджена наказом по університету від 30 жовтня 2020 р. № 1489СТ

2. Термін подання студентом роботи до екзаменаційної комісії 15 грудня 2020 р.

3. Вихідні дані до роботи _____

Технологічний стек: Vue, Vuex, JavaScript,

PostgreSQL _____

Графічний редактор: Figma _____

IDE: Visual Studio Code _____

4. Перелік питань, що потрібно опрацювати в роботі _____

1 Аналіз літератури

2 Аналіз сучасних методів побудови клієнт-серверної системи.

3 Вибір технологічного стеку.

4 Програмна реалізація

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 17 слайдів


6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)


Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спец. частина	проф. Немченко		15.12.2020

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	03.09.2020 – 10.09.2020	
2	Аналіз предметної області	11.09.2020 – 30.09.2020	
3	Аналіз джерел з проблемної галузі	03.10.2020 – 15.10.2020	
4	Аналіз сучасних технологічних рішень	16.10.2020 – 20.10.2020	
5	Вибір технологічного стеку	21.10.2020 – 29.10.2020	
6	Реалізація клієнтської частини	02.11.2020 – 09.11.2020	
7	Реалізація серверної частини	15.11.2020 – 27.11.2020	
8	Оформлення пояснювальної записки	28.11.2020 – 14.12.2020	

Дата видачі завдання 03 вересня 2020 р.

Студент  _____

Керівник роботи _____ (підпис)  _____ Немченко В.П.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить 78 сторінок, 30 рисунків, 1 таблиця, 15 джерел за переліком посилань.

СИСТЕМА УПРАВЛІННЯ НАВЧАЛЬНИМ ПРОЦЕСОМ, NEST.JS, VUE.JS, BOOTSTRAP, TYPEORM, КЛІЄНТ-СЕРВЕРНИЙ ДОДАТОК, VUEX, ЕЛЕКТРОННИЙ ЖУРНАЛ, POSTGRESQL, RESTFUL

В атестаційній роботі була запропонована система управління навчальним процесом, а саме електронний журнал. Був проведений аналіз предметної області, сучасних способів створювання подібних систем.

Був здійснений вибір архітектурної концепції, технологій для реалізації системи управління навчальним процесом, як на клієнтській стороні, так і на серверній.

Була реалізована система управління навчальним процесом, а саме електронний журнал, що дозволила зручніше, більш організовано, ефективніше навчання для вчителів, батьків та учнів.

ABSTRACT

Master's thesis contains 78 pages, 30 figures, 1 table, 15 sources according to the list of links.

LEARNING MANAGEMENT SYSTEM, NEST.JS, VUE.JS, BOOTSTRAP, TYPEORM, CLIENT-SERVER APPLICATION, VUEX, ELECTRONIC JOURNAL, POSTGRESQL, RESTFUL

In the certification work, a system for managing the educational process was proposed, namely an electronic journal. The analysis of subject area, modern ways of creation of similar systems was carried out.

The choice of architectural concept, technologies for the implementation of the management system of the educational process, both on the client side and on the server side was done.

A system of educational process management was implemented, namely an electronic journal, which allowed for more convenient, more organized, more effective learning for teachers, parents and students.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

.....8

ВСТУП.....

.....9

1. АНАЛІЗ СУЧАСНОГО СТАНУ ОСВІТНІХ ОНЛАЙН СЕРВІСІВ В УКРАЇНІ ТА СВІТІ.....

11

1.1 Світові тренди змін в освітній сфері.....

13

1.1.1 Доповнена Віртуальна Реальність / Змішана
Реальність.....13

1.1.2 Набір приладів у
класі.....14

1.1.3 Перероблений навчальний
простір.....15

1.1.4 Штучний
інтелект.....15

1.1.5	Персоналізоване
навчання.....	16
1.1.6	
Гейміфікація.....	17
1.2	Вплив COVID-19 на освітню
сферу.....	18
1.2.1	Вплив на школи.....
18	
1.2.1	Вплив на родини.....
20	
1.2.3	Оцінювання учнів.....
21	
1.3	Проблеми української сфери освіти під час пандемії COVID-
19.....	22
1.3.1	Брак у вчителів та викладачів попереднього досвіду
дистанційного	
навчання.....	
24	
1.3.2	Брак універсального доступу до інтернету та обладнання, необ-
хідного	для
навчання.....	24
1.3.3	Неузгодженість телевізійних уроків та шкільної
програми.....	24
1.3.4	Неможливість відтермінування захисту кваліфікаційних
робіт..	26
1.4	Приклади електронних
журналів.....	26

2 АНАЛІЗ ФУНКЦІОНАЛЬНИХ ВИМОГ ТА ТЕХНОЛОГІЙ РЕАЛІЗАЦІЇ СИСТЕМИ УПРАВЛІННЯ НАВЧАЛЬНИМ ПРОЦЕСОМ	30
---	----

2.1 Дослідження предмета, цілей і особливостей систем управління навчальним процесом	30
--	----

2.2 Порівняльний аналіз інструментів, фреймворків та технологій розробки	30
--	----

2.2.1 Життєвий цикл та теоретичні засади технології SPA	
---	--

31

2.3 Дослідження механізмів використання фреймворків для розробки системи управління навчальним процесом	31
---	----

2.3.1 Фреймворк Backbone.....	33
-------------------------------	----

2.3.2 Фреймворк Knockout.....	34
-------------------------------	----

2.3.3 Фреймворк Ember.....	36
----------------------------	----

2.3.4 Фреймворк Vue.....	36
--------------------------	----

2.3.5 Фреймворк React.....	37
----------------------------	----

2.4 Огляд та аналіз існуючих систем, розроблених за SPA концепцією	
--	--

37

44	2.5 Постановка завдання
46	2.6 Обґрунтування вибору фреймворку для розробки системи
..47	3 ДОСЛІДЖЕННЯ ТА АНАЛІЗ АРХІТЕКТУРНИХ РІШЕНЬ ТА ЗАСОБІВ РОЗРОБКИ СИСТЕМИ
47	3.1 Вибір загальної архітектури системи
48	3.2 Визначення загального паттерна розробки додатку
и.....52	3.3 Вимоги до клієнт-серверної архітектури систем
54	3.4 Вибір програмного каркасу для реалізації системи
	3.5 _____ Технологія _____ для _____ управління пакетами55
	3.6 Застосування технологій Gulp і Webpack для збірки проекту57
59	3.7 Обґрунтування вибору CSS-фреймворку Bootstrap 3
	3.8 Вибір серверної технології для розробки системи60
..64	4 ПРОЕКТУВАННЯ ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ НАВЧАЛЬНИМ ПРОЦЕСОМ
	4.1 Основна концепція побудови системи
.....64	

<u>4.2 Проектування бази даних системи</u>	
66	
<u>4.3 Керівництво користувача-</u>	
<u>вчителя</u>	<u>68</u>
4.4 Керівництво користувачів батьків.....	
70	
<u>ВИСНОВКИ</u>	
<u>74</u>	
<u>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ</u>	
<u>76</u>	

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Скорочення

БД – база даних

ІС – інформаційна система

ПЗ – програмне забезпечення

СУБД – система управління базами даних

API – Application Programming Interface – програмний Інтерфейс додатку

JSON – JavaScript Object Notation – форма представлення даних, в який самі дані представляються у вигляді об'єкта мови JavaScript, виду «Ключ»:«значення»

JWT – JSON Web Tokens – відкритий стандарт (RFC 7519) для передачі так званих «запитів» (англ. claims) між двома сторонами, взаємодіючими з допомогою веб-додатків

HTML – HyperText Markup Language – мова гіпертекстової розмітки

SPA – Single Page Application – односторінкові додатки це веб-додатки, які, фактично, розміщуються в одному html-документі, і здійснюють взаємодію з користувачем шляхом динамічного завантаження HTML, JavaScript і CSS, необхідних для роботи, з метою забезпечення користувачем досвідом використання програм схожих з настільними додатками

Терміни

Токен – деякий об'єкт, який представляє право на здійснення операцій

Хостинг – послуга з надання простору для розміщення інформаційних систем в Інтернеті

REST – стиль архітектури програмного забезпечення, при якому вся інформація про запит зберігається в самому запиті, відсутня зберігання сервером станів

ВСТУП

В теперішніх умовах життя вже досить складно уявити собі суспільство без використання такого блага цивілізації, як Інтернет. На сучасному етапі розвитку інформаційних технологій для отримання та передачі будь-яких видів інформації використання комп'ютеру та мобільних гаджетів стають єдиними засобами, що надають широкі можливості керування інформацією.

У зв'язку с пандемією COVID-19 світ зустрівся величезними проблемами через велику кількість блокувань. Мабуть всі сфери людини так чи інакше постраждали від неї. Одна з цих сфер – це освітня. В один момент всі освітні заклади зачинились на карантин та почали адаптувати навчання в он-лайні.

Основним призначенням адаптивної системи управління навчальним процесом, а саме електронного журналу є створення єдиного сучасного сервісу, в якому кожен учасник процесу навчання зможе контролювати та відслідковувати в режимі реального часу стан навчання. Велика кількість шкіл в Україні не мали змоги користуватися підготовленими сервісами для швидкої і безперервної адаптації процесу навчання. Із-за цього виникла низка проблем: вчителі не могли зручно проводити уроки з своїх дисциплін, батьки не мали змоги контролювати процес навчання за допомогою щоденника, деякі учні не мали змоги заходити вчасно, або взагалі на сесії уроків через різні проблеми з апаратною частиною, або інтернетом, а також учні молодших класів не мали змоги нормально адаптуватися до нового етапу життя.

Метою створення адаптивної системи управління навчальним процесом в рамках магістерської роботи є проектування та програмна реалізація мережевої системи електронного журналу для вчителів та батьків 1-4 класів, в якому інтерфейс спроектовано згідно зі змінами до оцінювання учнів молодших класів. Сервіс дозволяє зручно і швидко вчителям провести урок, оцінити

кожного учня, давати коментарі батькам, переглядати розклад занять, показувати статистику навчання та інше.

Практичне значення одержаних результатів. Розроблена адаптивна система управління навчальним процесом, а саме електронний журнал для використання вчителями та батьками учнів 1-4 класів.

Робота складається з 4 розділів. У першому розділі розглянуті основні тренди та проблеми у світовій сфері навчання та Україні. У другому розділі проаналізовані функціональні вимоги та технології для реалізації адаптивної системи управління навчальним процесом. У третій частині досліджені та проаналізовані архітектурні рішення та засоби розробки системи. У четвертій частині описане проектування та практична реалізація системи управління навчальним процесом. У висновках стисло наведено підсумок розробки та дослідження. У додатках міститься технічне завдання, лістинг коду основних частин програми та ілюстративний матеріал до захисту.

1 АНАЛІЗ СУЧАСНОГО СТАНУ ТРЕНДІВ І ПРОБЛЕМ ОСВІТНЬОЇ СФЕРИ У СВІТІ ТА УКРАЇНІ

1.1 Світові тренди в освітній сфері

Викладачі з усіх класів приходять, щоб усвідомити переваги технологій у класі. Як правило, освіта є однією з останніх галузей, яка зазнала значних змін, тримаючись за застарілі методи та практики. Але завдяки цифровій трансформації та піднесенню освітніх технологій вчителі почали вносити кардинальні зміни в свої інструкції, оцінки, навіть фізичний склад своїх класів, і набагато швидшими темпами, ніж очікувалося. Ці сучасні тенденції стають головними в освіті через те, як вони впливають на навчання студентів.

1.1.1 Доповнена Віртуальна Реальність / Змішана Реальність.

Давно минули часи, коли від студентів очікували, щоб вони спокійно сиділи за партами. Освітні технології успішно роблять навчання спільним та інтерактивним. Розширена, віртуальна та змішана реальність - це приклади трансформаційних технологій, які покращують навчання вчителів, одночасно створюючи захоплюючі уроки, які цікаві та цікаві для учня.

Віртуальна реальність має можливість перенести зовнішній світ у клас і навпаки. Такі програми, як Unimersiv, можуть перевезти студентів до Давньої Греції, тоді як Cospaces дозволяє студентам ділитися своїми віртуальними творіннями зі світом. Ад'юнкт-професор Університету Вілкса та незалежний освітній технолог Кеті Шрок приходять до висновку, що віртуальна реальність може підвищити візуальну грамотність, технологічну грамотність та увагу до аудиторії. Ідея поєднання AR / VR / MR дуже очікувана. Візьмемо, наприклад, приватну компанію Magic Leap. Незважаючи на те, що він ще нічого не продав,

Magic Leap вже оцінюється в чотири з половиною мільярди доларів! Це говорить про прогнозовані нескінченні можливості технологічної трансформації класів.



Рисунок 1.1 – Світові тренди майбутньої освіти

1.1.2 Набір приладів у класі

Школи віддаляються від BYOD або приносять власний пристрій, і учням більше не потрібно відвідувати технологічну лабораторію для доступу до комп'ютера чи ноутбука. Останні роки продемонстрували збільшення наборів комп'ютерів у класі, що стало можливим частково завдяки федеральному фінансуванню. Школи за титулом I отримали кошти згідно із Законом "Кожен

учень досягає успіху", а кілька грантів та пожертв обладнали навчальні кабінети по всій країні iPad та ноутбуками для кожного учня.

На Chromebook Google припадає більше половини пристроїв у навчальних класах США. У 2014 році в навчальних закладах було використано понад три мільйони Chromebook. Оскільки ця кількість продовжує зростати, зростає і потреба у більшій увазі до програм, що навчають навичкам цифрового громадянства. Сьогодні всепроникне Інтернет-середовище відкриває захоплюючі можливості, які вимагають належної освіти студентів щодо кібербезпеки та індивідуальної відповідальності.

1.1.3 Перероблений навчальний простір

Зайдіть у більшість навчальних кабінетів по всій країні, і навряд чи ви знайдете ряди письмового столу, спрямовані в передню частину кімнати. З тих пір викладачі зрозуміли, що їхні класи повинні імітувати робочу силу, що надихнуло їх на створення просторів для співпраці для полегшення навчання учнів. Впровадження технологій підтримало їхню діяльність.

Навчальні кімнати XXI століття - це дошки SMART, а не дощі, а майданчики SMART - замість окремих місць для сидіння. Студенти відправляються у віртуальні екскурсії, а не просто читають текст; вони створюють ЗМІ, а не просто дивляться на них. Перероблений навчальний простір насичений інтегрованими технологіями, що означає, що студенти не просто використовують ці речі, але вони розуміють, як ними користуватися для досягнення конкретної мети. Більше того, деякі з цих навчальних просторів відсутні навіть у класі. Коледжі та університети створюють більше неформальних навчальних місць у кампусі, оскільки вони розуміють важливість створення та співпраці цілодобово, не лише під час занять.

1.1.4 Штучний інтелект

Використання ІІІ у вищих навчальних закладах вже виявилось корисним. Австралійський університет Дікін використовував IBM Watson для створення віртуальної консультативної служби для студентів, яка була доступна цілодобово та без вихідних. Віртуальні радники Уотсона поставили понад 30 000 запитань у першому триместрі, звільнивши фактичних радників для вирішення більш складних питань. Інше використання ІІІ включає чат-боти.

Оскільки чат-боти оснащені прогресом природної мови, як це знайдено в Siri, вони мають людську здатність відповідати на запитання щодо домашнього завдання, допомагати студентам в оформленні документів, таких як фінансова допомога або оплата рахунків, та полегшувати навантаження людей, які зазвичай обслуговують ці ролі. Інші застосування ІІІ в освіті включають персоналізацію навчання (про що детальніше йдеться нижче), оцінку якості навчальної програми та змісту та сприяння індивідуальному навчанню за допомогою інтелектуальних систем навчання. Технологія не має на меті замінити вчителів, а лише доповнити їх.

1.1.5 Персоналізоване навчання

Зараз ми можемо персоналізувати навчання більше, ніж будь-коли. Починаючи від вибору школи - державної, приватної, чартерної, віртуальної - до доступних варіантів того, як навчається студент, освіта може бути адаптована відповідно до кожної людини. Комбіноване навчання дає більшу відповідальність перед учнем, оскільки воно передбачає менше прямих вказівок викладача та більше методів навчання на основі відкриттів. Змішане навчання є прикладом того, як студенти можуть контролювати певні елементи свого навчання, приймаючи рішення щодо таких речей, як те, куди і з яким темпом вони рухаються через матеріал.

Адаптивне навчання подібне до змішаного тим, що воно також дозволяє студентам приймати рішення щодо таких речей, як часові рамки та шлях їх

навчання. Адаптивна технологія навчання збирає інформацію про поведінку студентів, коли вони відповідають на запитання, а потім згодом використовує цю інформацію для надання миттєвого зворотного зв'язку для відповідного коригування навчального досвіду. Навчальні засоби з адаптивною послідовністю постійно аналізують дані студентів у режимі реального часу та приймають рішення за секунду на основі цих даних. Він автоматично змінює те, що буде наступним у послідовності, будь то змінений зміст або інший порядок навичок, у відповідь на те, як студент виконує студент.

Ще одна навчальна платформа, "Осмос", була створена лікарями для лікарів і зробила революцію в навчанні студентів-медиків: "Використовуючи освітні концепції, що базуються на фактичних даних, такі як запитання, картки та відео, зображення, що співвідносяться з якорями пам'яті, адаптивне повторення, спільне навчання та гейміфікація, Осмос максимізує навчання та утримання ". Така персоналізація перетворює освіту на метод навчання «вибирай собі власну пригоду», використовуючи зацікавленість та залучення студентів.

1.1.6 Гейміфікація

Гра та навчання стикаються, коли класи використовують ігри як інструктаж. Ігрові технології роблять вивчення важких предметів більш захоплюючим та інтерактивним. У міру розвитку технології вона швидко використовується для вдосконалення навчальних ігор у кожній дисципліні. Старший віце-президент Університету Дрекселя Сьюзен Олдрідж приписує ці ігри дзеркальним відображенням реальних життєвих проблем, вимагаючи від студентів використання цінного набору навичок для їх вирішення: «Ці віртуальні ігрові світи дають унікальну можливість застосувати нові знання та зробити критично важливими рішення, одночасно виявляючи перешкоди, враховуючи різні перспективи та репетируючи різні відповіді ».

Оскільки ці ігри покликані забезпечити негайний зворотний зв'язок, студенти мають внутрішню мотивацію продовжувати грати в них, відточуючи навички протягом усього. Нові технології та нові моделі навчання захоплюють і пропонують студентам немислимі раніше можливості, але вони потребують постійної ІТ-підтримки. Оскільки освітні установи продовжують стримуватись і приймати ці тенденції цифрової трансформації, ми повинні враховувати поточну парадигму навчання технологіям та рухатись до командного підходу. У міру збільшення очікувань студентів, відповідь на ці потреби також повинен зростати.

1.2 Вплив COVID-19 на освітню сферу

Пандемія COVID-19 - це перш за все криза здоров'я. Багато країн (справедливо) вирішили закрити школи, коледжі та університети. Криза кристалізує дилему, з якою стикаються політики, які стоять між закриттям шкіл (зменшенням контактів та порятунком життя) та утриманням їх відкритими (дозволяючи працівникам працювати та підтримувати економіку). Сильні короткочасні зриви відчувають багато сімей у всьому світі: навчання вдома - це не лише значний шок для продуктивності батьків, але й для соціального життя та навчання дітей. Навчання рухається в Інтернеті, у неперевіреному та безпрецедентному масштабі. Оцінювання студентів також рухається в Інтернеті, з великою кількістю спроб і помилок та невизначеністю для всіх. Багато оцінок просто скасовано. Важливо, що ці перебої будуть не лише короткотерміновою проблемою, але також можуть мати довгострокові наслідки для постраждалих когорт і, ймовірно, збільшуватимуть нерівність.

1.2.1 Вплив на школу

Ходити до школи - найкращий інструмент державної політики для підвищення кваліфікації. Хоча навчальний час може приносити задоволення і може підвищувати соціальні навички та соціальну обізнаність, з економічної точки зору першочерговою точкою перебування в школі є те, що це підвищує здібності дитини. Це робить навіть порівняно короткий час у школі; навіть порівняно короткий період пропуску школи матиме наслідки для зростання кваліфікації. Але чи можемо ми підрахувати, наскільки переривання COVID-19 вплине на навчання? Не дуже точно, оскільки ми перебуваємо в новому світі; але ми можемо використовувати інші дослідження, щоб отримати порядок величини.

Корисними є два докази. Карлссон та ін. (2015) розглядають ситуацію, коли юнаки Швеції мають різну кількість днів, щоб підготуватися до важливих тестів. Ці відмінності є умовно випадковими, що дозволяє авторам оцінити причинний вплив шкільного навчання на навички. Автори показують, що навіть лише десять днів додаткового навчання значно підвищує бали на тестах використання знань ("кристалізований інтелект") на 1% від стандартного відхилення. Як надзвичайно грубий показник впливу поточного закриття шкіл, якщо ми просто екстраполюємо ці цифри, дванадцять тижнів менше навчання (тобто 60 навчальних днів) означає втрату 6% від стандартного відхилення, що є нетривіальним. Вони не знаходять істотного впливу на навички вирішення проблем (приклад „рідинного інтелекту“).

Інший шлях до цього питання походить від Lavy (2015), який оцінює вплив на вивчення відмінностей у навчальному часі в різних країнах. Можливо, як не дивно, між країнами існують дуже суттєві відмінності в годинах навчання. Наприклад, Лаві показує, що загальна кількість тижневих годин навчання математики, мови та природознавства в Данії на 55% вища, ніж в Австрії. Ці відмінності мають значення, що спричиняє суттєві відмінності в результатах тестових балів: ще одна година на тиждень протягом навчального року з

основних предметів збільшує результати тестів приблизно на 6% від стандартного відхилення. У нашому випадку втрата, можливо, 3-4 години на тиждень викладання математики протягом 12 тижнів може бути подібною за величиною до втрати години на тиждень протягом 30 тижнів. Отже, досить химерно і, безсумнівно, випадково, ми знову закінчуємося втратами приблизно 6% від стандартного відхилення. Залишаючи близьку подібність осторонь, ці дослідження, можливо, припускають можливий ефект не більше 10% від стандартного відхилення, але, безумовно, вище нуля.

1.2.2 Вплив на родини

Можливо, на розчарування деяких, дітей зазвичай не відправляли додому грати. Ідея полягає в тому, що вони продовжують навчання вдома, в надії не пропустити занадто багато.

Сім'ї займають центральне місце в освіті, і за ними широко погоджуються надавати основні вклади у навчання дитини, як описано Бьорклундом та Сальванесом (2011). Поточне глобальне розширення домашнього шкільного навчання на перший погляд може сприйматися цілком позитивно, оскільки воно може бути ефективним. Але, як правило, ця роль розглядається як доповнення до шкільного матеріалу. Батьки доповнюють навчання математики дитини, практикуючи підрахунок чи виділення простих математичних задач у повсякденному житті; або вони висвітлюють уроки історії поїздками до важливих пам'яток чи музеїв. Інше питання - бути головним рушієм навчання, навіть у поєднанні з онлайн-матеріалами; і хоча багато батьків у всьому світі успішно навчають своїх дітей вдома, це здається навряд чи узагальненим для всього населення.

Тож, хоча глобальне домашнє навчання, безсумнівно, призведе до натхненних моментів, сердитих моментів, веселих моментів та моментів розчарування, але здається дуже малоімовірним, що воно в середньому

замінить втрачене в школі навчання. Але головний момент полягає в наступному: між сім'ями, ймовірно, будуть суттєві розбіжності в тому, наскільки вони можуть допомогти своїм дітям вчитися. Ключові відмінності включають (Oreopoulos et al. 2006) кількість часу, який можна приділити навчанню, некогнітивні навички батьків, ресурси (наприклад, не всі матимуть набір для доступу до найкращих онлайн-матеріалів), а також обсяг знань - важко допомогти дитині навчитися чогось, чого ви самі не можете зрозуміти. Отже, цей епізод призведе до збільшення нерівності зростання людського капіталу для постраждалих когорт.

1.2.3 Оцінювання учнів

Закриття шкіл, коледжів та університетів не лише перериває навчання студентів у всьому світі; закриття також збігається з ключовим періодом оцінювання, і багато іспити були відкладені або скасовані.

Внутрішні оцінки, мабуть, вважаються менш важливими, і багато з них були просто скасовані. Але їх суть полягає в тому, щоб надати інформацію про успіхи дитини для сімей та вчителів. Втрата цієї інформації затримує визнання як високого потенціалу, так і труднощів у навчанні і може мати шкідливі довгострокові наслідки для дитини. Андерсен і Нільсен (2019) розглядають наслідки великої ІТ-аварії в системі тестування в Данії. В результаті цього деякі діти не змогли пройти тест. Автори виявляють, що участь у тесті збільшила оцінку в тесті читання через два роки на 9% від стандартного відхилення, маючи подібні ефекти в математиці. Ці ефекти найбільші для дітей із неблагополучних верств населення.

Що важливо, блокування установ не лише впливає на внутрішні оцінки. Наприклад, у Великобританії всі іспити за основними державними кваліфікаціями - GCSE та рівнями A - були скасовані для всієї когорти. Залежно від тривалості блокування, ми, мабуть, спостерігатимемо подібні дії у всьому

світі. Однією з потенційних альтернатив скасованих оцінок є використання “прогнозованих оцінок”, але Мерфі і Вайнесс (2020) показують, що вони часто є неточними, і що серед студентів з високим успіхом прогнозовані оцінки для тих, хто має неблагополучні верстви, нижчі, ніж оцінки з більш сприятливий досвід. Іншим рішенням є заміна сліпих іспитів оцінками викладачів. Докази з різних параметрів показують систематичні відхилення між сліпим та сліпим обстеженнями, де напрям упередженості, як правило, залежить від того, чи належить дитина до групи, яка зазвичай добре працює (Burgess and Greaves 2013, Rangvid 2015). Наприклад, якщо дівчатка зазвичай краще виступають у предметі, неосвічена оцінка успішності хлопчика, швидше за все, буде упередженою. Оскільки такі оцінки використовуються як ключова кваліфікація для вступу до вищої освіти, перехід до невідомих суб’єктивних оцінок може мати потенційні довгострокові наслідки для рівності можливостей.

Також можливо, що кар’єра деяких студентів може отримати користь від перебоїв. Наприклад, у Норвегії вирішено, що всі учні 10-х класів отримають ступінь середньої школи. І Морін та МакНеллі (2008) показують, що відмова від нормальних процедур іспитів у Франції у 1968 році (після студентських заворушень) призвів до позитивних довгострокових наслідків на ринку праці для постраждалої когорти.

У вищих навчальних закладах багато університетів та коледжів замінюють традиційні іспити інструментами онлайн-оцінювання. Це нова область як для викладачів, так і для учнів, і оцінки, ймовірно, матимуть більшу похибку вимірювання, ніж зазвичай. Дослідження показують, що роботодавці використовують сортові показники, такі як класифікація ступенів та середні оцінки для сортування заявників (Piorunik et al. 2020). Отже, посилення шуму сигналів заявників потенційно знизить ефективність пошуку нових випускників на ринку праці, які можуть відчувати повільніший ріст заробітків та вищі

показники розподілу робочих місць. Це дорого як для особистості, так і для суспільства в цілому (Fredriksson et al., 2018).

1.3 Проблеми української сфери освіти під час пандемії COVID-19

Процес навчання під час карантину був напруженим для всіх учасників освітнього процесу: вчителів, школярів, батьків, викладачів та студентів. Через відсутність або брак попереднього досвіду дистанційного навчання та реакції з боку уряду виникла ціла низка освітніх проблем. Нижче ми наведемо їх перелік, який в основному ґрунтується на вторинних даних, а саме онлайн-опитуваннях щодо навчання в закладах загальної середньої освіти, проведених Державною службою якості освіти (ДСЯО) та освітнім омбудсменом. Треба зауважити, що всі ці опитування проводилися ще навесні, а отже, показують проблеми, з якими стикнулись учасники навчального процесу ще в період жорсткого карантину, коли всі заклади були на дистанційному навчанні. Однак більшість з цих проблем досі є актуальними та невирішеними.

1.3.1 Брак у вчителів та викладачів попереднього досвіду дистанційного навчання.

За даними опитування, яке проводила ДСЯО, у 55,53% шкіл дистанційне навчання було проблемою, оскільки вони раніше не мали досвіду такої форми навчання, тому не були готові до навчання під час карантину. 47,5% викладачів (20 590 осіб) зазначили, що раніше не використовували технології дистанційного навчання у своїй педагогічній діяльності. Натомість, як ми зазначали раніше, вчителі не отримували рекомендацій щодо методики проведення дистанційного навчання або навчання за допомогою освітніх онлайн-інструментів. Водночас для вчителів корисним були б розроблені поради щодо того, як пояснювати новий матеріал, давати зворотній зв'язок,

тренувати навички, оцінювати прогрес та визначати тривалість уроків в умовах дистанційного навчання.

На жаль, в Україні наразі не проводилися ґрунтовні дослідження про те, як проходило дистанційне навчання в закладах вищої освіти. Однак можна припустити, що викладачів так само стикалися з проблемою браку досвіду та навичок дистанційного навчання, організовуючи роботу самостійно. Водночас бракує рекомендації щодо організації занять в умовах дистанційного навчання, а також інструментів, за допомогою яких викладачів змогли б обмінюватися вже набутим досвідом одні з одними.

1.3.2 Брак універсального доступу до інтернету та обладнання, необхідного для навчання.

Ця проблема зачепила як вчителів, так і школярство. За даними ДСЯО, 62,35% директорів зазначили, що більшість вчителів та/або учнів не мали необхідного обладнання вдома, а 46,9% вказали на низькошвидкісне підключення до інтернету. Більша частина вчителів користувалися комп'ютером разом з іншими членами сім'ї, а 6% вчителів не мали доступу до швидкісного інтернету. Про низьку якість під'єднання до інтернету свідчать 22,3% опитаних батьків. Крім того, 8,8% батьків сказали, що їхні сім'ї не мають комп'ютера.

За даними опитування, проведеного освітнім обмудсменом, більшість дітей (81,6%) використовували мобільні телефони, імовірно смартфони, для дистанційного навчання. Натомість менше користуються ноутбуком (45,6%) та стаціонарним комп'ютером (34,3%). Водночас не всі завдання можливо виконувати на телефоні, тому діти могли потребувати ноутбуків чи комп'ютерів, які належали батькам. Якщо ж батьки також працювали дистанційно, це могло ускладнювати як навчання, так і роботу. Ще складнішою могла бути ситуація в сім'ях, де більше однієї дитини шкільного віку. Те саме

стосується і вчителів, які могли бути вимушені ділити технічні засоби зі своїми дітьми або іншими членами родини.

Можна припустити, що з цими самими проблемами стикалися й студенти та університетські викладачі, які мали ділити гаджети для дистанційного навчання з іншими членами родини.

1.3.3 Неузгодженість телевізійних уроків та шкільної програми.

«Всеукраїнська школа онлайн» — телевізійний проєкт, який транслювався по телебаченню та на YouTube-каналі МОН для учнів 5-11 класів з 11 предметів. Проте за результатами опитування керівників закладів загальної середньої освіти лише у 29,3% школах розклад дистанційних занять узгоджувався з розкладом телевізійних уроків. Відповідно, учнівство могло отримувати нерівномірно розподілену інформацію з окремих дисциплін. Наразі МОН планує запустити нові телеуроки, які будуть транслюватися на безкоштовній платформі і які розробляють ГО «Освіторія» у співпраці з Міністерством цифрової трансформації.

Значна частина навчального матеріалу була залишена для самостійного вивчення. У багатьох навчальних закладах дистанційне навчання виглядало як надсилання матеріалу для самостійного опрацювання з підручника, письмові завдання для перевірки знань та оцінювання результатів без обговорення, зворотного зв'язку та пояснень.

На думку учнів, опитаних ДСЯО, вчителі частіше їм надсилали перелік параграфів та вправ з підручника на самостійне опрацювання (41,2%), ніж проводили дистанційні заняття (17,25%). Намагання видати самостійне навчання за дистанційне через брак попереднього досвіду також підтверджують результати опитування щодо інструментів у взаємодії між вчителями та учнівством, які застосовувались найчастіше. Зокрема, на першому місці опинився Viber — ним користувалися 92,4% вчителів. Відповідно, вони

могли відправляти через месенджер теми та завдання для опрацювання, що фактично є самостійним навчанням, а не дистанційним.

Схожа ситуація була й у закладах вищої освіти: частина викладачів замінювали дистанційні заняття на відпрацювання у вигляді письмових робіт, що призводило до надмірного завантаження студентів. Такі методи не сприяють якісному засвоєнню матеріалу, а, навпаки, створюють додатковий стрес. При чому ситуація в межах одного університету могла бути різною, оскільки деякі викладачів давали письмові відпрацювання, а інші проводили дистанційні заняття. Водночас координації між самими викладачів та контролю з їхнього боку за навантаженням на студентів могло не бути. Ця проблема знову актуалізується зараз, адже більша частина закладів вищої освіти перейшли на дистанційне навчання.

1.3.4 Неможливість відтермінування захисту кваліфікаційних робіт.

Держава фінансує вищу освіту, сплачуючи університету за кожну людину, яка навчається на бюджеті. Відповідно, якщо людина навчалася, але не отримала вчасно диплом, буде вважатися, що кошти витрачені неефективно. Через брак фінансової автономії заклади вищої освіти могли отримати через це санкції «за нецільове використання бюджетних коштів». Щоб цього не допустити, у своєму листі МОН рекомендував «закінчити навчальний рік вчасно». Це не дало університетам достатньо гнучкості, аби відтермінувати захисти кваліфікаційних робіт. Студенти, які не встигли завершити освітню програму у визначений термін, мали або оплатити переслуховування кредитів, які не встигли опанувати, або бути відрахованими.

1.4 Приклади електронних журналів

Провівши опитування серед учителів ХСШ №62, я зібрав низку проблем з якими зустрілися вчителі під час карантинних заходів, а це:

- велика кількість сервісів та інтерфейсів для проведення уроку;
- батьки постійно задають питання у месенджери (Telegram, Viber);
- складність користування інтерфейсами для проведення уроку (Google Meet, Zoom, Google Classrooms);

Дані проблеми вирішуються за рахунок спеціальних систем управління навчальним процесом, а саме електронним журналом, в якому вчителі можуть зручно виставляти оцінки, контролювати зручно всі навчальні процеси, а також батьки мають змогу бачити успіхи своєї дитини, список домашніх завдань та інше. Основна мета даних сервісів – збільшити зручність та швидкість навчальних процесів, а також зменшити кількість навантажень на всіх учасників процесу.

На жаль, таких сервісів в Україні не багато та більшість з них платні і не кожна школа може дозволити їх купити у користування.

Нижче список сервісів з електронними журналами в Україні:

- 1) Urkschools (<http://ukrschools.com.ua>) – сервіс, який надає можливість ко

-

ригуватись таким функціоналом:

- Журнал оцінок. Батьки, учні, вчителі мають змогу контролювати успіхи дитини;
- Навчальні матеріали. Дозволяє зберігати будь-які навчальні матеріали - книжки, корисні посилання, відеофайли та інше. Все це завжди доступно учням на нашому сайті в зручному вигляді;
- Домашні завдання.

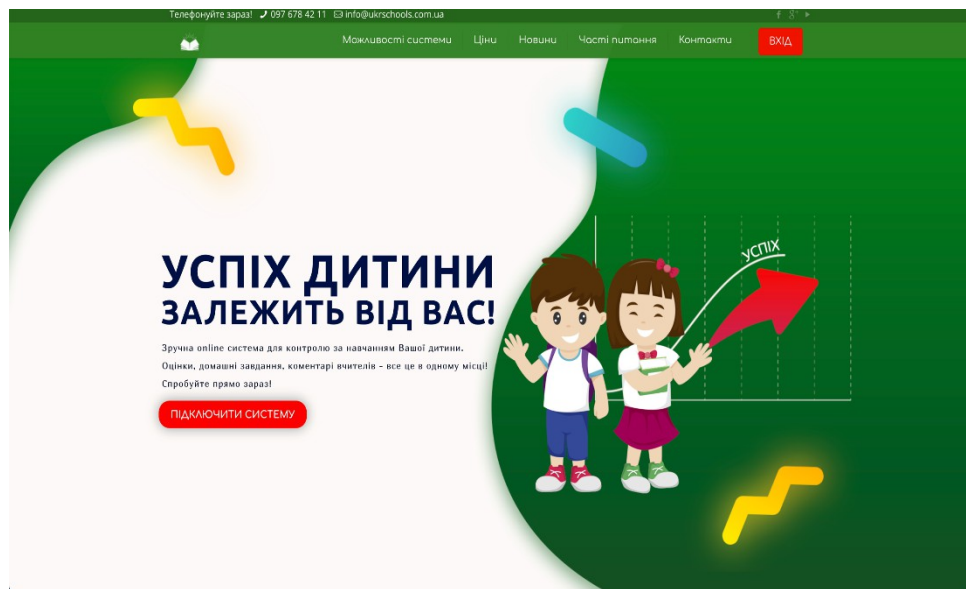


Рисунок 1.2 – Сервіс Ukrschools

2) Проект Підручник.ua (<http://pidruchnik.ua>) від команди, яка займається сервісом 3D-друку Kwambio і цифровим видавництвом Gutenbergz, спочатку створювався як платформа онлайн-підручників для українських шкіл. Але з часом команда вирішила на його основі побудувати «електронну школу» - щоб вчителі і учні мали доступ до підручників, журналів, розкладом, щоденникам в онлайні. Спочатку Підручник.ua запускався в Одесі в 2017 році, пропонуючи підписку на 600 підручників, які рекомендує Міністерство освіти і науки України. Всього за час запуску там зареєструвалося понад 40 000 користувачів, на сайті - близько 250 000 користувачів з Одеси, 20 000 активних користувачів в мобільному додатку. Сервіс працює за моделлю передплати, але він безкоштовний для вчителів і учнів, його закуповують місцеві департаменти освіти. Вартість передплати на підручники - 140 грн на рік на учня, сервіс «електронна школа» буде коштувати близько 554 грн на одного учня.

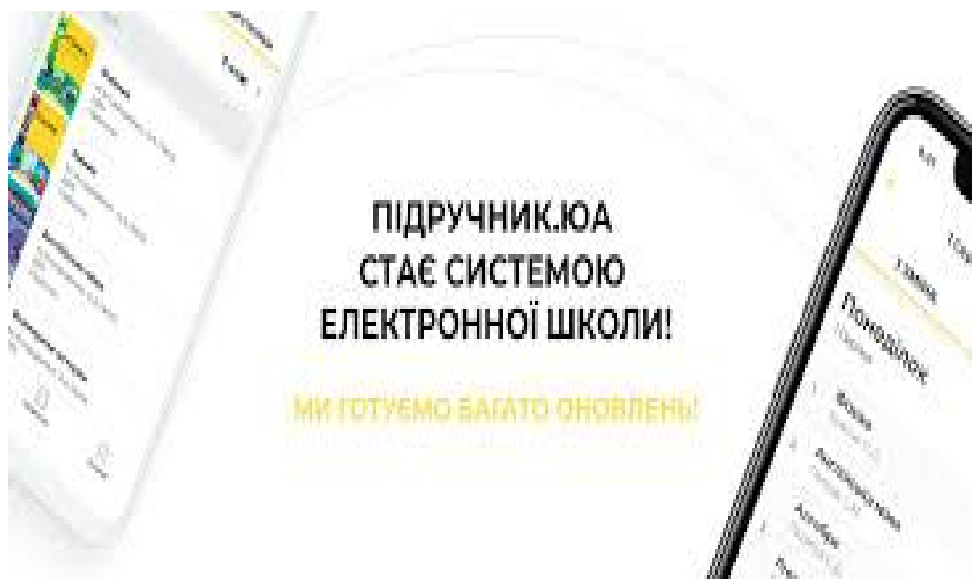


Рисунок 1.3 – Проект Підручник.ua

3) SMLS (smls.com.ua) – сервіс, призначений для загальноосвітніх навчальних закладів. Керівник будь-якої школи за допомогою комп'ютера з доступом в Інтернет може приєднатися до SMLS, зареєструвавши своє освітня установа. SMLS, надаючи інформаційну безпеку, перетворює кожен школу-користувача в надійного партнера. SMLS доступний всі 365 днів в році.

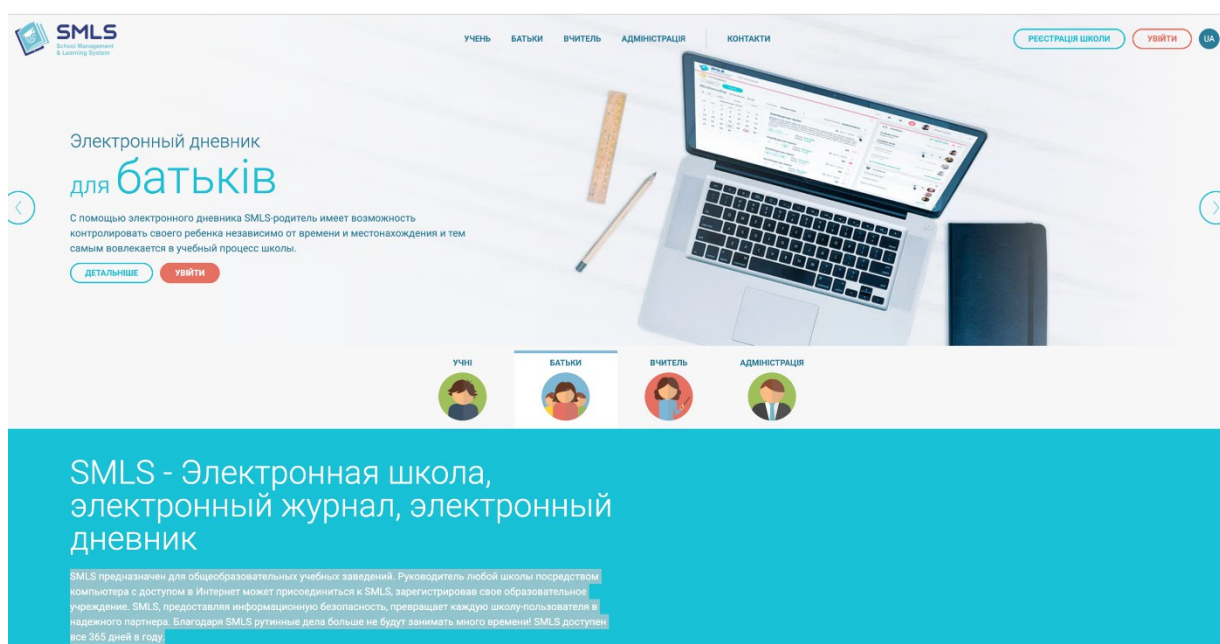


Рисунок 1.4 – Сервіс SMLS

2 АНАЛІЗ ФУНКЦІОНАЛЬНИХ ВИМОГ ТА ТЕХНОЛОГІЙ РЕАЛІЗАЦІЇ СИСТЕМИ УПРАВЛІННЯ НАВЧАЛЬНИМ ПРОЦЕСОМ

2.1 Дослідження предмета, цілей і особливостей систем управління навчальним процесом. Електронний журнал

У зв'язку з тим, що після введення карантинних обмежень в Україні, всі навчальні заклади були переведені на дистанційну форму навчання. Проте не

всі навчальні заклади та учасники процесу навчання (учні, батьки) мали змогу повноцінно і якісно адаптуватися до цих змін.

Причин велика кількість, але основні – це відсутність у учнів пристроїв для виходу в інтернет, перебої з інтернетом, невміння вчителів користуватися сучасними засобами комунікації (наприклад, Zoom, Viber, Google Classrooms та інші), адаптація дітей, які тільки роблять свої перші освітні кроки (діти 1-4 класів), незмога батьків контролювати процес навчання, вчасно и розгорнуто отримувати інформацію про свою дитину.

Основна ціль електронного журналу – це вирішити основну низку питань, приведену вище. Він повинен бути з сучасним та зручним інтерфейсом, адаптований під особливості навчання учнів 1-4 класів, а саме: багато предметів не оцінюється за 12-ти бальною шкалою, оцінювання активності учнів в сучасній формі.

2.2 Порівняльний аналіз інструментів, фреймворків та технологій розробки

На сьогоднішній день існує декілька основних та найпопулярніших фреймворків. Вибір подібного фреймворку потребує детального порівняльного аналізу для виявлення найбільш оптимального. Найпопулярніші фреймворки такі: [Angular](#), [React](#), Backbone, Knockout, [Vue](#).

2.2.1 Життєвий цикл та теоретичні засади технології Single Page Application

Для прикладу розглянемо звичайний, стандартний за своєю концепцією сайт – Wikipedia. Відкриваючи якусь сторінку веб-сайту користувач повинен чекати доки його веб-браузер встановить зв'язок з сервером і далі буде

обробляти його запит. Далі веб-браузер повинен завантажити різноманітні дані ресурсу, щоб вони відобразились на сайті, водночас відтворюючи ті, які вже загрузились.

Весь цей час, доки відбувається цей процес (а іноді він може тривати відносно дуже довго), користувач повинен очікувати. Після того, як користувач ознайомиться зі сторінкою і захоче переглянути іншу сторінку, щось скоректувати на існуючій, або щось ще зробити на даному сайті, на нього чекає повне оновлення сторінки за новим URL. Деякі дані та ресурси цього сайту будуть кешовані веб-браузером і завантаження наступної сторінки пройде трохи швидше, але все одно це не вирішує проблеми і недоліки такого підходу залишаються.

За таким способом, на сьогоднішній день, працює значна більшість існуючих веб-сайтів. Однак Web існує вже багато років і за весь цей час відбулось багато змін, створились нові технології та підходи, AJAX, JavaScript і т.д. Сучасні веб-сайти перестали бути просто сторінками з інформацією, вони почали набувати інтерактивності і взаємодією з користувачем.

На сьогоднішній день існує багато різноманітних технологій, які дозволяють розробникам зробити з веб-сайту систему, дуже схожу на нативні додатки для операційних систем. Щоб це реалізувати, потрібно всі зміни обробляти за допомогою скриптів і запити на сервер робити за допомогою AJAX.

Відкриваючи подібний веб-сайт користувач має лише один раз завантажити сторінку цього веб-сайту і далі все, що користувач буде робити на сайті (перехід по сторінкам, внесення якихось даних, тощо) буде відбуватись без перезавантаження сторінки і відбуватись все буде дуже швидко, майже миттєво. Це і є концепція технології SPA (односторінкового додатку). Сьогодні дуже багато нових проектів створюються саме за допомогою цієї технології.

Angular – JavaScript-фреймворк, створений на основі TypeScript. Розроблений і підтримуваний компанією Google, він описується як JavaScript MVC-фреймворк. Angular (він же – Angular 2+, він же – Angular 2 або ng2) є переписані наступником AngularJs (він же – Angular.js або AngularJS 1.x). Незважаючи на те, що AngularJS (рання версія) був випущений в жовтні 2010 року, його творці досі усувають недоліки даного фреймворка.

На даний момент самою останньою версією є 5. Фреймворк Angular використовується такими компаніями, як Google, Wix, weather.com, healthcare.gov і Forbes. На рисунку 2.1 та рисунку 2.2 порівнюється життєвий цикл традиційного веб-сайту та односторінкового додатку.



Рисунок 2.1 – Життєвий цикл традиційного веб-сайту



2.3 Дослідження механізмів використання фреймворків для розробки системи управління навчальним процесом

в основі фреймворків, про які йде мова, лежать певні елементи. Елемент отримує команду введення і після деяких внутрішніх поводжень, обчислень він повертає запитований UI-шаблон (наприклад, область вводу, виводу або елемент списку) в якості результату. При цьому не повинно виникати труднощів при повторному використанні певних елементів. Наприклад, є `grid` - елемент, що складається з `header` і декількох `row`-елементів, з різними параметрами (колонками таблиць, `header`, `row` і т.д.), який може бути повторно використаний з іншими параметрами для іншої веб-сторінки.

2.3.1 Фреймворк Backbone.

Backbone – JavaScript фреймворк який допоможе в створенні односторінкових додатків вирішуючи деякі основні проблеми що виникають при їх розробці.

Основні можливості:

- моделі зі зв'язування за ключом й користувацькими подіями;
- відкритий сирцевий код (розповсюджується за MIT ліцензією);
- колекції з великим набором методів для роботи з перерахованими сутностями;
- представлення з декларативною обробкою подій;
- інтеграція за існуючим RESTFUL JSON API;
- клієнтська маршрутизація.

Backbone – це простий фреймворк, який вписується в один JavaScript файл. Backbone був розроблений Джеремі Ашкенасом за допомогою CoffeeScript. Фреймворк базується на шаблоні проектування MVP (англ. Model-ViewPresenter). Моделі дозволяють зв'язати ключ-значення і події для обробки зміни даних.

Backbone особливо популярний серед розробників, які шукають просту структуру для невеликих веб-додатків, без застосування великих фреймворків як Angular або Ember. Backbone найкраще використовувати при побудові простих односторінкових додатків. При виборі програмного каркаса слід враховувати те, що в якості загального паттерна розробки програми був обраний SPA у вигляді товстого клієнта, при цьому інтерфейс клієнт-серверної взаємодії – REST.

Крім цього, слід також спиратися на наступні важливі не функціональних вимог до додатка:

- гнучкість і масштабованість;
- високий рівень інтерактивності призначеного для користувача інтерфейсу;
- швидкість роботи програми повинна бути досить комфортною для роботи користувача з додатком;
- користувач повинен мати можливість працювати з додатком за допомогою ПК і планшета.

Спираючись на перераховані вимоги, а також з огляду на плюси і мінуси різних фреймворків, як програмного каркаса для front-end частини програми буде використаний Angular фреймворк.

2.3.2 Фреймворк Ember.

Ember – JavaScript фреймворк який спрощує створення масштабованих односторінкових додатків. Представляє з себе каркас що реалізує MVC шаблон [2].

Основні можливості:

- клієнтська маршрутизація;
- відкритий сирцевий код (розповсюджується за MIT ліцензією);
- моделі які містять данні що відповідають поточному стану додатку;
- контролеривикористовуються щоб надати моделі логіку відображення;
- бібліотека Ember Data для зв'язку з сервером й отримання JSON даних;
- шаблони (прототипування) на мові HTMLBars.

У цьому досить популярному фреймворку упор робиться на підвищення продуктивності програміста. Ember теж відноситься до MVC фреймворків. У ньому використовується шаблонізація і вбудований рендерер відображення, який автоматично оновлює дані так само, як і Angular, Backbone і React. Ember підтримує технологію веб-компонентів, що дозволяє розширювати HTML за допомогою власних тегів, в цьому він схожий на Angular.

Однак, розробники фреймворка зробили великий акцент на його самодостатність, при цьому при цьому програвши в гнучкості. Ember на дає великого контролю розробнику, вирішуючи багато речей самостійно.

Як наслідок, Ember не підходить для розробки веб додатків, в яких однією з вимог є гнучкість [2].

2.3.3 Фреймворк Knockout.

Knockout – JavaScript фреймворк для створення візуального інтерфейсу який використовує шаблон MVVM.

Основні особливості:

- безкоштовний;
- відкритий сирцевий код (розповсюджується за MIT ліцензією);
- написаний на чистому JavaScript;
- легкий (54 кб мінімізований);
- не має залежностей;
- підтримується всіма сучасними браузерами (IE 6+, Firefox 3.5+, Chrome, Opera, Safari (desktop/mobile));
- повністю задокументований.

Основні можливості:

- декларативне зв'язування;
- автоматичне оновлення візуального інтерфейсу при оновленні даних моделі;
- відстежування залежностей;
- шаблони (прототипування) [2].

2.3.4 Фреймворк Vue.

Vue - це прогресивний фреймворк для створення користувацьких інтерфейсів. На відміну від фреймворків-монолітів, Vue створений придатним для поступового впровадження. Його ядро в першу чергу вирішує завдання рівня уявлення (view), що спрощує інтеграцію з іншими бібліотеками та існуючими проектами. З іншого боку, Vue повністю підходить і для створення складних односторінкових додатків (SPA, Single-Page Applications), якщо

використовувати його спільно з сучасними інструментами та додатковими бібліотеками.

це прогресивний фреймворк для створення користувацьких інтерфейсів. На відміну від фреймворків-монолітів, Vue створений придатним для поступового впровадження. Його ядро в першу чергу вирішує завдання рівня уявлення (view), що спрощує інтеграцію з іншими бібліотеками та існуючими проектами. З іншого боку, Vue повністю підходить і для створення складних односторінкових додатків (SPA, Single-Page Applications), якщо використовувати його спільно з сучасними інструментами та додатковими бібліотеками.

2.3.5 Фреймворк React.

ReactJS має відкритий вихідний код і розвивається в основному за допомогою Facebook за участю інших великих технологічних компаній. React описує себе як JavaScript бібліотека для створення користувацьких інтерфейсів.

В архітектурному плані, React найближче до концепції патерну проектування Model-View-Controller. Він відповідає за подання даних, отримання та обробку введення користувача. При цьому спосіб зберігання даних цілком лягає на розсуд розробника.

Однак, React — це перш за все уявлення (вид) додатку. У складі фреймворка відсутня будь-яка готова система події, або шар даних. При розробці додатків на React велику кількість часу йде на підбір і зв'язку відповідного технологічного стека. Тому React підійде в основному для написання відносно простих веб-додатків [2].

2.4 Огляд та аналіз існуючих систем, розроблених за SPA концепцією

При здійсненні аналізу існуючих систем у мережі Інтернет, можливо виявити переваги та недоліки застосування технології SPA, яка була використана при проектуванні та розробці Інтернет-додатків. Google за час свого існування створив багато онлайн сервісів (рис. 2.3), більшість з яких цілком або майже цілком зроблені як односторінкові.

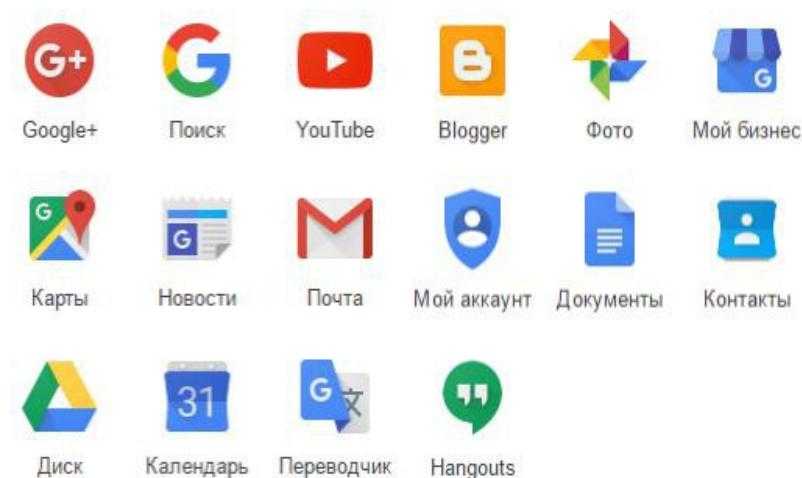


Рисунок 2.3 – Web-сервіси Google 15

Google Mail або Gmail – веб-інтерфейс до пошти від Google (рис. 2.4). Він був одним із найперших додатків виконаних за SPA концепцією. Додаток представляє із себе список електронних листів з каталогами та пошуком, також є можливість редагувати та переглядати листи у невеличкому вікні. Оскільки на момент створення Gmail не існувало майже ніяких відкритих фреймворків для SPA, додаток був написаний на чистому HTML та JavaScript без використання додаткових бібліотек, хіба що власних від Google. За час існування він оновлювався і зараз виглядає доволі сучасно.

Google вже запустила наступника під назвою Inbox (рис. 2.5), який теж є односторінковим додатком. Однак Gmail все ще користується популярністю. Довгий час успішної роботи доводить що використання концепції SPA було вдалим рішенням. Про таке кажуть «перевірено часом».

Цей веб сервіс є гарним прикладом SPA, проте перейняти з його архітектури майже нічого не можливо оскільки ніяких бібліотек використано не було. Тому ми будемо розглядати інші, новіші веб-сайти.

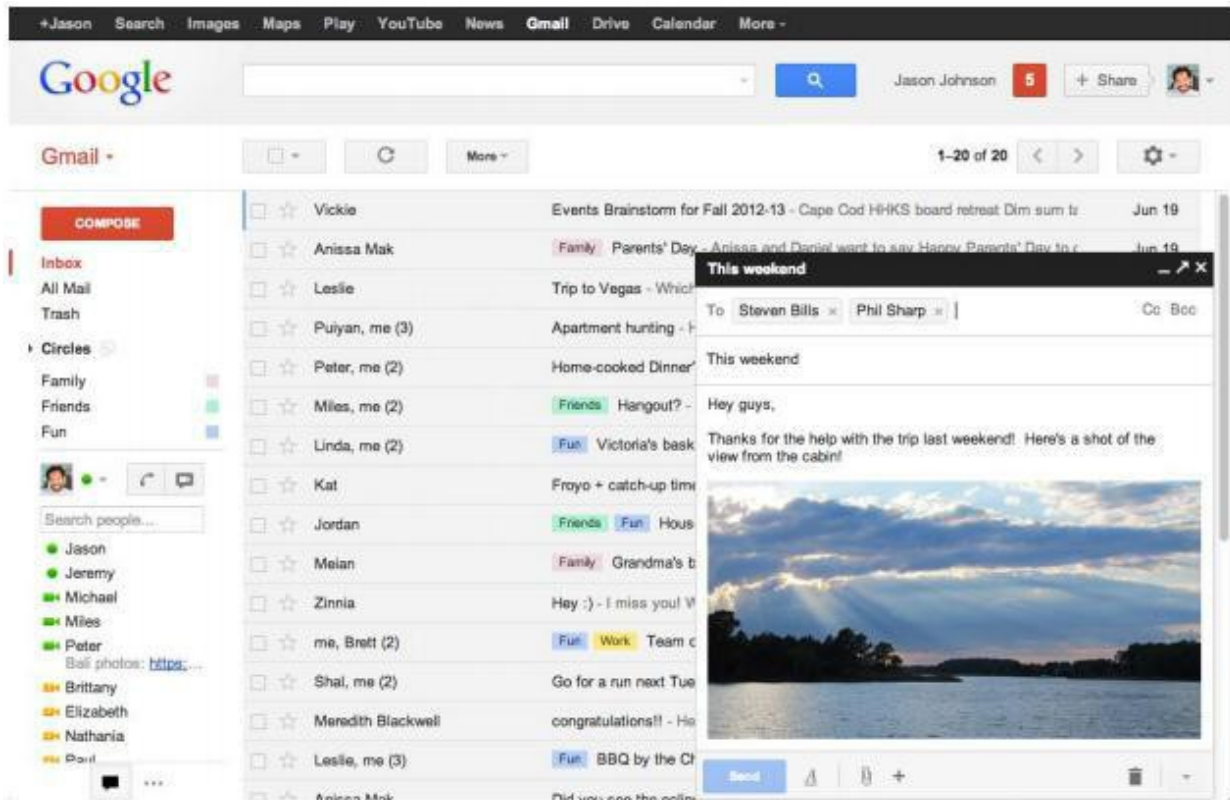


Рисунок 2.4 – Веб-інтерфейс сервісу Google Mail

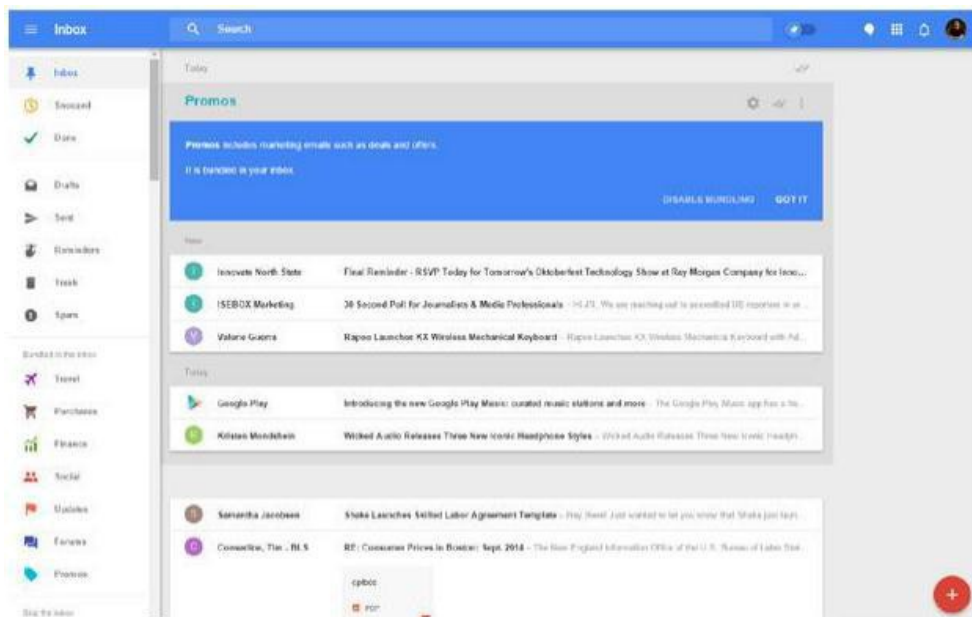


Рисунок 2.5 – Веб-інтерфейс сервісу Google Inbox

Наступний популярний онлайн сервіс яким користуються по всьому світі це Google Maps (рисунок 2.6). Карти від Google дозволяють дивитися карти будь якого місця на планеті у схематичному або супутниковому вигляді з великої кількості інформації. Також сервіс дозволяє будувати маршрути між точками використовуючи різні види транспорту та віртуально мандрувати вулицями в крупних містах.

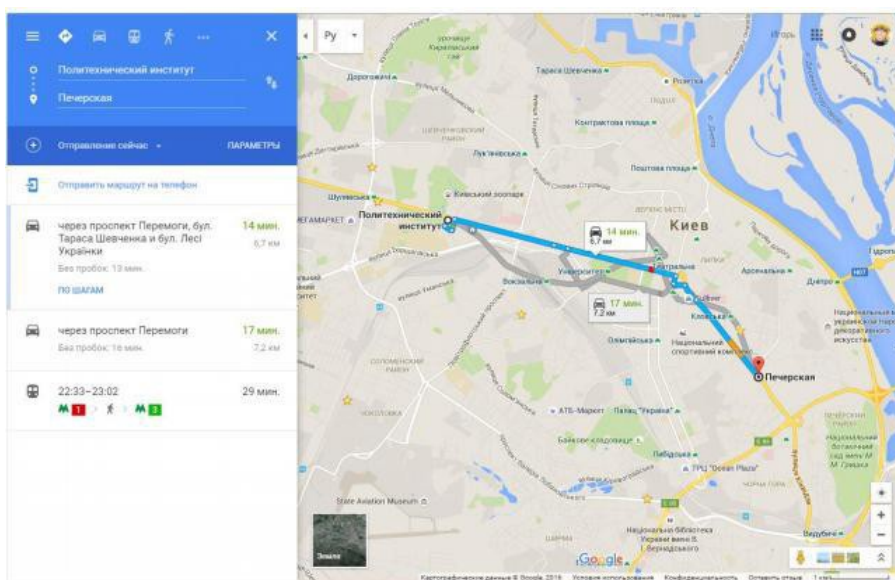


Рисунок 2.6 – Вигляд сервісу Google Maps

Сервіс Microsoft Office Online.

Microsoft, так само як і Google, має багато односторінкових додатків серед своїх онлайн сервісів (рис. 2.7). Найцікавіші з них це онлайн версії популярних програм з офісного пакету Microsoft Office, такі як Word Online (рис. 2.8), Excel Online (рис. 2.9), PowerPoint Online та сервіс хмарного сховища OneDrive. Всі ці додатки пов'язані між собою й дозволяють безпосередньо відкривати та редагувати офісні документи збережені в хмарному сховищі.



Рисунок 2.7 – Онлайн сервіси Microsoft

Пакет необхідних програм Microsoft Office завжди був пакетом стандартних додатків, який ставився одразу після встановлення Windows. Однак зараз кожен може скористатися знайомими офісними додатками просто відвідавши сайт з будь якого пристрою що має сучасний браузер. Звісно онлайн версія офісу має деякі обмеження й неповний функціонал, але для більшості задач вона прекрасно підійде. Також не можна не відмітити те, що вона є безкоштовною, на відміну від нативного пакету.

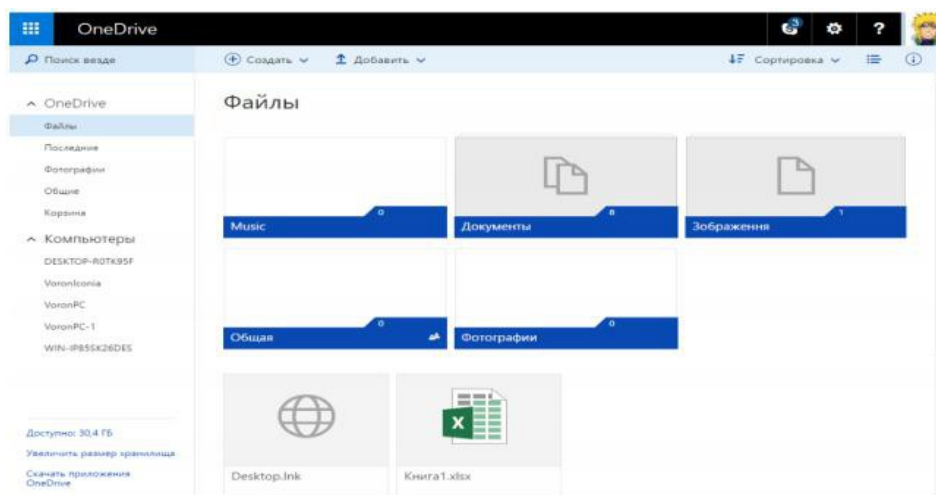


Рисунок 2.8 – Сервіс OneDrive від Microsoft

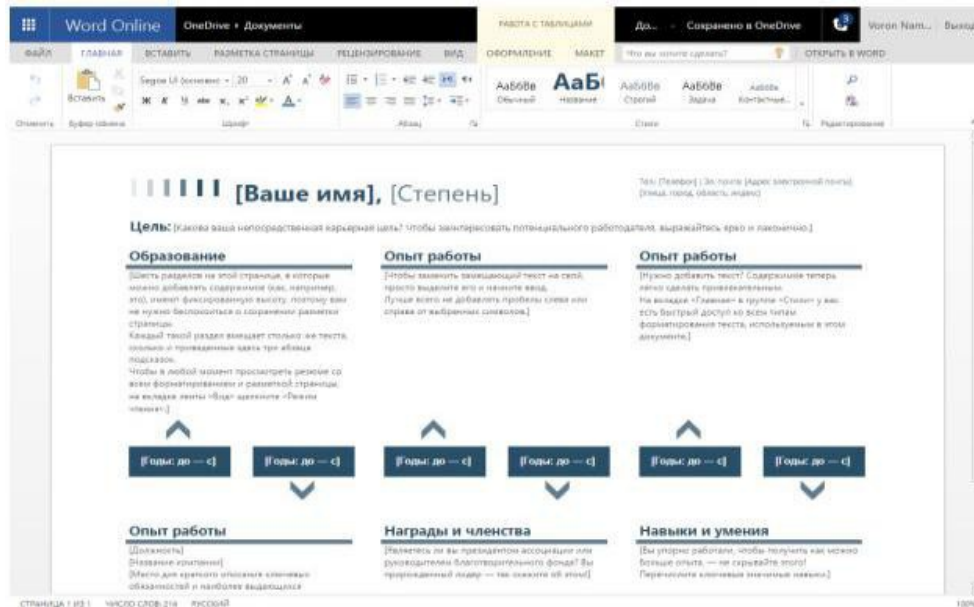


Рисунок 2.9 – Сервис Word Online від Microsoft

Web-сервис Azure Portal.

Azure portal – це веб портал для керування хмаровою платформою Microsoft Azure. Цей веб сайт, що за виглядом нагадує самостійну операційну систему, насправді звичайний веб-сайт виконаний за концепцією SPA.

Заходячи на сайт користувач бачить анімацію завантаження поки не завантажаться основні необхідні для роботи додатки скрипти та ресурси. Після завантаження користувач опиняється на панелі з плитками що відображують цікаву йому інформацію (рис. 2.10). Зліва є панель що дозволяє керувати ресурсами. Замість відкриття нових вікон та вкладок у браузері при навігації по сайту, усі вікна відкриваються в середині сайту, один за одним й складають горизонтальну ленту вікон, що добре відображує їх дочірність. Цей дизайн горизонтально розташованих вікон нагадує леза бритви, тому розробники назвали вікна блейдами (англ. Blade – лезо).

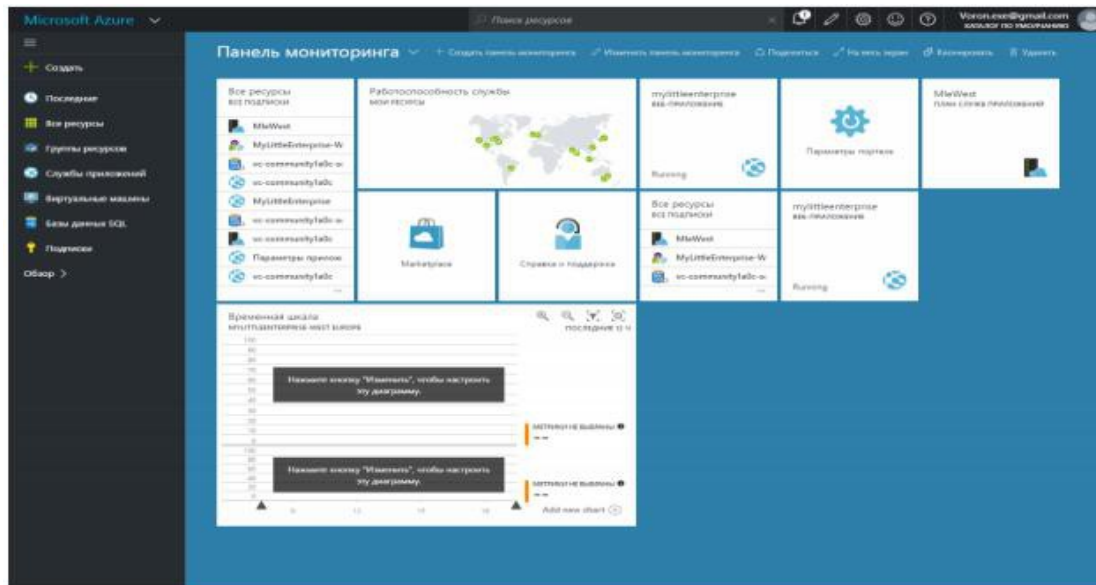


Рисунок 2.10 – Сервис Azure Portal Dashboard19

Сайт надає величезну кількість можливостей керування платформою. Все відбувається швидко, оскільки все необхідне для роботи інтерфейсу завантажується одразу при вході на сайт. Додаткові ресурси підвантажуються асинхронно за допомогою скриптів, тому користувач майже не помічає цього.

Портал був розроблений у 2014 року на заміну старому порталу, архітектура якого не могла забезпечити зручне керування всіма ресурсами Microsoft Azure, кількість яких швидко зростала. Новий портал, завдяки модульності, має чудову масштабованість й зможе за незначних змін розширити функціонал у майбутньому.

Технології що були використані:

- TypeScript;
- KnockoutJS;
- LESS.

2.5 Постановка завдання

У ході магістерської роботи необхідно здійснити проектування та розробити систему управління навчальним процесом, а саме електронний журнал з застосуванням технології Single Page Application.

Систему управління навчальним процесом, а саме електронний журнал необхідно реалізувати як мережеву інформаційну систему у вигляді веб-сервісу для користувачів.

Основним призначенням створення системи управління навчальним процесом, а саме електронного журналу є створення додатку в мережі Інтернет для того, щоб користувачі могли переглядати свої успіхи в навчанні, проводити уроки та оцінювати учнів, бачити розклад занять та важливих занять. Також користувачі можуть ознайомитися з інформацією по кожному уроку, переглянути детальні успіхи, завантажити статистику за період.

Метою створення системи управління навчальним процесом, а саме електронного журналу є забезпечення безкоштовного та без обмеженого контролю та користуванню під час навчальних процесів.

При розробці систему бухгалтерського обліку для об'єднань співвласників будинків визначені наступні користувачі:

- користувач-адміністратор;
- користувач- вчитель.
- користувач-батьки.

Користувачу-адміністратору повинні пропонуватися наступні функціональні можливості:

- додавати, редагувати та видаляти користувачів системи;
- додавати та видаляти предмети;

- відображати історію уроків;
- формувати класи та зв'язувати їх з вчителями та розкладами;
- формувати розклади для кожного класу.

Користувачу-вчителю повинні пропонуватися наступні функціональні можливості:

- передивлятися весь список предметів на сьогоднішній день;
- передивлятися список класів;
- передивлятися список учнів;
- проводить урок та оцінювати учнів;
- переглядати електронний журнал;
- переглядати та давати коментарі щодо поведінки та успіхів учнів;

Користувачам-батькам повинні пропонуватися наступні функціональні можливості:

- передивлятися весь список предметів на сьогоднішній день;
- передивлятися список предметів;
- передивлятися детальну інформацію про кожен з предметів;
- переглядати розклад на весь період навчання;
- переглядати детальну статистику навчання;
- отримувати та реагувати на коментарі від вчителів.

Для розробки даного додатку була вибрана саме технологія SPA, тому що на сьогоднішній день вона являється дуже прогресивною і дає багато додаткових можливостей при розробці Web-додатків. Багато серверних функцій переходять під управління клієнта, тобто з back-end частини у front-end. Під час розробки можна використовувати багато специфічних конструкцій, таких наприклад як v-if або v-for, і це дає змогу зменшити навантаження на сервер, що в свою чергу дає значне збільшення швидкості роботи самого додатку.

2.6 Обґрунтування вибору фреймворку для розробки системи

Таблиця 1.1 – Порівняння фреймворків

	Backbone	Ember	Knockout	Vue
Перший випуск	2010	2011	2015	2015
Ліцензія	MIT	MIT	MIT	MIT
Основний шаблон	MVP	MVC	MVVM	MVVM
Зв'язування	Ні	Так	Так	Так
Клієнтська маршрутизація	Так	Так	Ні	Так
Відкритий вихідний код	Так	Так	Так	Так
Шаблони (прототипування)	Ні	Так	Так	Так

Зараз існує достатньо фреймворків для комфортної розробки односторінкового додатку. Кожен з них має свої недоліки й переваги. Вибір треба роботи в залежності від вимог та цілей до розроблюваного продукту.

Наступна підсумкова таблиця показує що Vue 2 володіє найбільшою кількістю функцій та можливостей. Приймаючи до уваги, що розвиток фреймворку дуже стрімкий, Vue є гарним вибором. Результати відображені в таблиці 1.1

Таким чином можна зробити висновок що Vue є дуже перспективним й тому стає найкращим вибором для подальшого детального розгляду й надбання практичних навичок в будівництві односторінкового веб-додатку.

3 ДОСЛІДЖЕННЯ ТА АНАЛІЗ АРХІТЕКТУРНИХ РІШЕНЬ ТА ЗАСОБІВ РОЗРОБКИ СИСТЕМИ

3.1 Вибір загальної архітектури системи

Сучасні програмні додатки та інформаційні системи досягли високого рівня розвитку і термін або поняття «архітектура» у застосуванні до них дозволяє правильно побудувати і сконструювати інформаційну систему в цілому, забезпечуючи її ефективне і надійне функціонування.

Архітектура інформаційної системи – концепція, визначальна модель, структуру, виконувані функції і взаємозв'язок компонентів інформаційної системи.

У міру розвитку програмних систем все більшого значення набуває їх інтеграція один з одним з метою побудови єдиного інформаційного простору підприємства. Для того щоб побудувати правильну і надійну архітектуру і грамотно спроектувати інтеграцію програмних систем необхідно чітко слідувати сучасним стандартам в цих областях. Без цього велика ймовірність, створити архітектуру, яка нездатна розвиватися і задовольняти зростаючим потребам користувачів ІТ.

Класифікація програмних систем за їх архітектурою представляється таким чином:

- централізована архітектура;
- архітектура «файл-сервер»;
- дворівнева архітектура «клієнт-сервер»;
- багаторівнева архітектура «клієнт-сервер»;
- архітектура розподілених систем;
- архітектура веб-додатків;

– сервіс-орієнтована архітектура.

Дана система бухгалтерського обліку для об'єднань співвласників будинків розроблена як клієнт-серверна система. Клієнт-серверна система характеризується наявністю двох взаємодіючих самостійних процесів клієнта і сервера, які, в загальному випадку, можуть виконуватися на різних комп'ютерах, обмінюючись даними по мережі. За такою схемою можуть бути побудовані системи обробки даних на основі СУБД, поштові та інші системи.

Додаток на робочій станції «відповідає за все» – за формування інтерфейсу користувача, логічну обробку даних і за безпосереднє маніпулювання даними. Файловий сервер надає послуги тільки найнижчого рівня – відкриття, закриття і модифікацію файлів. Таким чином, безпосереднім маніпулюванням даними займається кілька незалежних і неузгоджених між собою процесів [3].

3.2 Визначення загального паттерна розробки додатку

На сьогоднішній день існують два основних паттерна розробки веб-додатків – SPA (англ. Single Page Application – односторінковий додаток) і МРА (англ. Multi Page Application – багатосторінковий додаток).

МРА – це традиційний веб-додаток. При такому підході кожен раз, коли додаток запитує дані або відправляє їх на сервер, він змушений отримувати нову сторінку в повному обсязі, а потім візуалізувати її в браузері.

Для відносно простих додатків такий підхід цілком підходить і не викликає будь-яких незручно. Але коли мова йде про створення додатка з багатим для користувача інтерфейсом, з високим ступенем інтерактивності, то на сторінці з'являється безліч користувальницьких інструментів. Все це веде до того, що розмір трафіку помітно збільшується, і при багатосторінковому підході неминучий з'являються проблеми з продуктивністю.

Генерація складних сторінок на сервері, передача їх клієнту та візуалізація в браузері займають тривалий час і тому значно зручніше використання ресурсу, збільшуючи час очікування.

Крім цього, будь-які дії користувача, такі як: доступ до потрібної інформації, відправки різних запитів и даних форм, редагування, збереження і видалення інформації, вимагають перезавантаження всієї сторінки сайту. При роботі з подібними програмами, користувач відчував незручності, так як після кожної дії, йому доводиться очікувати генерації та завантаження всієї сторінки цілком.

Альтернативним підходом є SPA. У цьому випадку з сервера завантажується єдина найпростіша веб-сторінка, до якої підключається JavaScript бібліотека (в більш складному випадку набір бібліотек), яка містить в собі програмний код, здатний динамічно змінювати DOM сторінки, і таким чином, перемальовувати контент сторінки в реальному часу.

Даний програмний код виконується на протязі всього сеансу роботи користувача з додатком. При цьому додаткові веб-сторінки не завантажуються, а періодично завантажується лише необхідна для перемальовування окремих ділянок єдиної сторінки інформація (як правило, в форматі JSON). При цьому перемальовуються тільки ті частини сторінки, які цього вимагають, наприклад, в разі втрати актуальності відображених на них даних. Під сторінками в такому додатку маються на увазі «віртуальні» сторінки в межах програми.

На рисунку 3.1 схематично представлені обидва патерна розробки. Пунктирна лінія умовно поділяє клієнтське і серверне середовища виконання. Стрілки показують напрямки руху даних між клієнтською і серверною стороною.

У випадку з МРА видно, що серверна сторона повністю бере на себе всі функції по формуванню уявлення, в той час як клієнтська сторона

займається виключно рендером сформованого статичного контенту. Варто зауважити, що в цьому випадку реалізація «товстого» клієнта зі своєю бізнес логікою та зберіганням складного стану важка, або взагалі неможлива.

У випадку з SPA сервер не формує готове представлення, а замість цього, протягом роботи програми, обмінюється з клієнтом інформаційними пакетами (в даному прикладі в форматі JSON). При цьому, клієнт може мати власний шар обробки бізнес логіки, більш того, клієнт може зберігати на своєму боці якісь дані, від яких буде залежати стан клієнта. В цьому випадку можна говорити, що клієнт може бути реалізований як товстий клієнт.

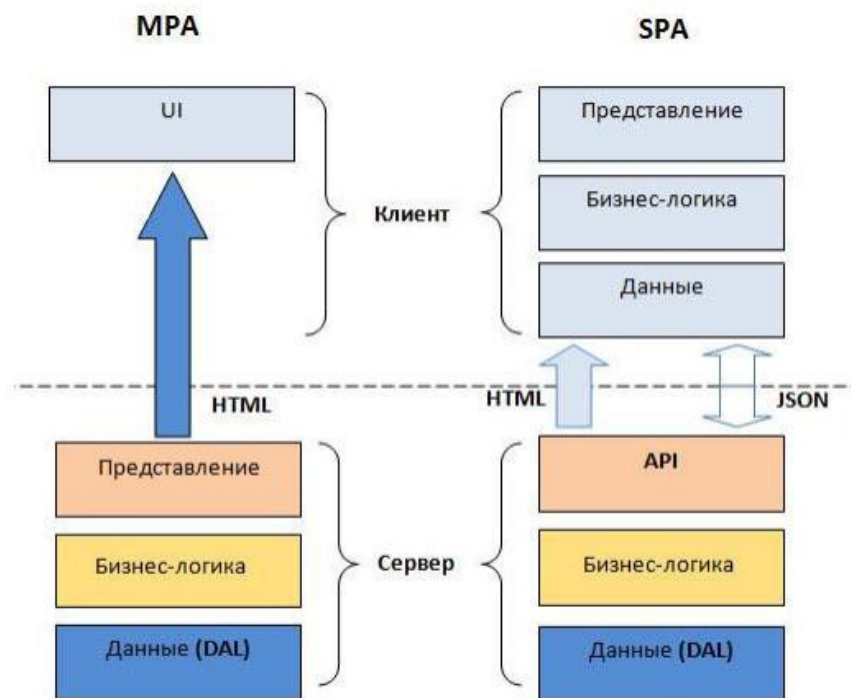


Рисунок 3.1 – Порівняльні схеми патернів розробки за технологією SPA та MPA

Як і люба інша технологія, SPA має певні недоліки та переваги відносно іншої технології. Недоліки SPA по відношенню до MPA:

- клієнт представляє з себе досить важкий front-end додаток, який потрібно завантажувати на кожен клієнт. Однак наявність функції кешування

ресурсів сучасними веб-браузерами зводять негативний ефект від цього недоліку до мінімуму;

- SEO – SPA ускладнює оптимізацію сайту під пошукові движки. Оскільки більша частина веб-сторінки будується на стороні клієнта, боти пошукових систем бачать сторінку зовсім інакше, ніж користувач. Для цього розробляється вимога підтримки SEO відсутня, так як основна функціональність програми полягає в наданні різного роду сервісу, при цьому контент, який необхідно індексувати пошуковими системами відсутній.

Переваги SPA по відношенню до МРА:

- більш висока швидкість роботи програми та завантаження сторінок – це перевага задовольняє третьому пункту не функціональних вимог щодо розроблюваного додатка;

- високий рівень інтерактивності призначеного для користувача інтерфейсу, а також поліпшене сприйняття призначеного для користувача інтерфейсу, оскільки завантаження даних з сервера відбувається у фоновому режимі – ця перевага задовольняє другому пункту не функціональних вимог щодо розроблюваного додатка;

- поділ програми на клієнтську (front-end, зовнішнє уявлення) і серверну (back-end) частини, що дозволяє реалізовувати безліч різних зовнішніх уявлень для різних платформ, в тому числі мобільних, взаємодіючих за єдиним API з серверною частиною програми – це перевага задовольняє першому і четвертому пункту не функціональних вимог щодо розроблюваного додатка;

- можливість реалізації «товстого» клієнта, що дозволить забезпечити більшу динаміку на стороні клієнта.

З огляду на пред'явлені вимоги до розробляється з додатком, а також беручи до уваги всі переваги і недоліки SPA, в якості загального паттерна розробки вибирається SPA.

«Товстий» або «тонкий» клієнт: у випадку з МРА, клієнт, як правило являється «тонким», так як більша частина бізнес логіки обробляється виключно серверною стороною. Класичний приклад такого клієнта – це браузер, який займається виключно рендером отриманих з сервера HTML сторінок із застосуванням до них CSS стилів. У цьому випадку на клієнті можуть бути елементи бізнес логіки, але, як правило дуже прості. Наприклад, перевірка коректності призначених для користувача форм за допомогою скриптів на мові JavaScript. Подібна логіка часто дублюється серверною стороною. Крім цього, «тонкий» клієнт зазвичай не зберігає стану програми. Замість цього він може зберігати тільки ідентифікатор користувача сесії, який передається серверу з кожним запитом.

В якості загального паттерна розробки був обраний МРА, і як вже було підмічено раніше, додаток може бути реалізовано як «товстий» клієнт. Реалізація SPA програми у вигляді товстого клієнта дозволить домогтися більшої динаміки і швидкості роботи програми. Окреслені переваги будуть досягатися завдяки перенесенню логіки, пов'язаної з формуванням відображення, з серверної сторони на клієнтську, при цьому клієнт буде зберігати в собі поточний стан відображення. У даного підходу є ще одна перевага – це збереження працездатності додатку після втрати зв'язку з серверною стороною.

Таким чином, з огляду на переваги товстого клієнта, зупиняємося на загальному паттерні розробки програми SPA у вигляді «товстого» клієнта.

3.3 Вимоги до клієнт-серверної архітектури системи

Після того, як було визначено загальний патерн розробки веб-додатку, можна спроектувати клієнт-серверну архітектуру системи бухгалтерського обліку для об'єднань співвласників будинків з застосуванням технології SPA:

– односторінковий веб-додаток передбачає поділ на серверну – back-end і клієнтську front-end частини. Клієнтська частина буде виконуватися на пристрої користувача, в той час як серверна – на окремому фізичному вузлі. Саме клієнтську програму буде завантажуватися з сервера при першому до нього зверненні;

– архітектурний стиль взаємодії між клієнтом і сервером – REST API з використанням стандарту JSON в якості даних;

– відповідно до одного з функціональних вимог про те, що програма має бути здатна брати на себе деякі функції клієнта і взаємодіяти з сервером безпосередньо в реальному часі, back-end додатки будуть взаємодіяти з сервером за допомогою спеціального бінарного TCP протоколу для віддаленого управління сервером – Server Management API. Даний протокол дозволяє взаємодіяти з сервером безпосередньо в реальному часі;

– Server Management API не дозволяє виробляти всі можливі операції над сервером. Існують варіанти використання програми, наприклад, такі як «Реєстрація аккаунта», «Відображення статистики» і т.д., яким не обов'язково безпосередньо взаємодіяти з сервером. В цьому випадку з додатком може знадобитися з'єднання з MySQL базою даних. Таким чином, back-end повинен вміти взаємодіяти з MySQL базою даних, при цьому back-end повинен мати доступ до відповідного вузла;

– відповідно до одного з функціональних вимог про те, що дані, які не пов'язані з предметною областю, додаток повинен зберігати у власній MySQL базі даних, потрібен окремий фізичний вузол з розгорнутою на ній MySQL базою даних. Серверна частина веб-програми повинна вміти взаємодіяти з MySQL базою даних, при цьому повинна мати доступ до відповідного вузла.

3.4 Вибір програмного каркасу для реалізації системи

Патерн розробки SPA дозволяє реалізовувати безліч різних зовнішніх уявлень призначеного для користувача інтерфейсу для різних платформ і пристроїв, проте, в якості першого і основного зовнішнього подання буде використовуватися веб-браузер, що інтерпретує додаток на мові JavaScript.

Написання повноцінного односторінкового додатку на чистому JavaScript можливо, однак отриманий вихідний код вийде дуже об'ємним, а отримана програма не буде мати достатню гнучкість і масштабованість, при цьому його буде складно тестувати. Саме тому виникає необхідність в програмному каркасі. Серед різних інформаційних технологій, веб-технології є одними з найбільш стрімко розвиваючих, в даний час пропонується велика кількість програмних каркасів для написання SPA. Розглянемо деякі з них.

Vue – популярний фреймворк, широко використовуваний для створення і підтримки складних веб-додатків. Vue має відкритий вихідний код (MIT ліцензія) за підтримки Vue-співтовариства, засновником якого є Єван Ю. Vue – MVVM фреймворк. Однією з його сильних особливостей є двостороннє зв'язування між моделями і поданням. Ця прив'язка даних дозволяє проводити автоматичне оновлення по обидва боки кожного разу, коли відбувається зміна даних. Це дозволяє створювати багаторазово використовувані компоненти уявлення, що забезпечує легкий обмін даними між серверної і клієнтської частиною.

Також у фреймворк вбудований движок маршрутизації і сервіс для зручної роботи з REST API. Vue можливо використовувати при побудові складного веб-додатку, в тому випадку, коли є потреба в єдиному модульному фреймворку.

Крім цього, Vue безперервно розвивається. Навколо фреймворка організувалося величезне співтовариство розробників, які періодично

пропонують зручні рішення у вигляді модулів для вирішення широкого спектра завдань.

3.5 Технологія для управління пакетами

Фреймворк Vue складається з окремих модулів. Причому, кожен модуль має свою версію. У процесі розробки програми всі модулі бажано підтримувати в актуальному стані. Крім цього, як вже говорилося раніше, для Vue фреймворка реалізована велика кількість сторонніх бібліотек, які можуть виявитися корисними для розроблюваного додатка. Всі використовувані сторонні пакети також необхідно підтримувати в актуальному стані.

Очевидно, що розробнику неможливо постійно стежити за актуальністю такої великої кількості бібліотек. Тому виникає необхідність в автоматичному пакетному менеджері. Для управління пакетами додатками добре підійде пакетний менеджер NPM, що входить до складу Node.js.

Node.js – програмна платформа, заснована на движку V8 (здійснює трансляцію JavaScript в машинний код), що перетворює JavaScript з вузькоспеціалізованої мови в мову загального призначення. Для використання пакетного менеджера досить в корінь проекту помістити JSON файл «package.JSON», в якому будуть перераховані всі необхідні пакети. Для управління пакетним менеджером використовуються різні команди, виконувані в командній оболонці поточного оточення.

Керування Node.js та npm відбувається через консоль. Отже перш за все необхідно відкрити консоль та перейти до теки з проектом. Для першого запуску, або після внесення змін у файл «package.JSON», треба виконати процедуру відновлення залежностей в ході якою будуть завантажені потрібні версії всіх необхідних залежностей. Для виконання цієї процедури необхідно в консолі ввести команду «npm install». Цей процес зображений на рисунку 3.2.

Після успішного запуску серверу веб-сайт буде доступним за адресами, які виведені у консолі. Також за додатковими адресами, які теж виведені у консолі, буде доступний графічний інтерфейс керування сервером, який зображений на рисунку 3.4.

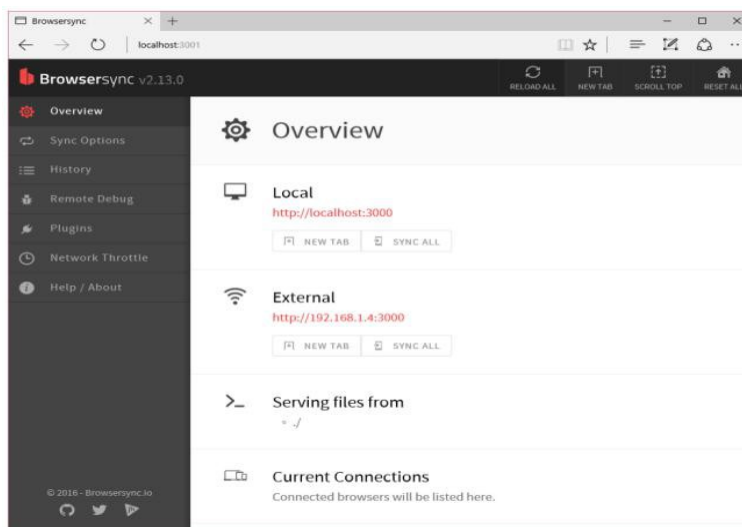


Рисунок 3.4 – Графічний інтерфейс керування сервером

3.6 Застосування технологій Gulp і Webpack для збірки проекту

Front-end додаток, написаний із застосуванням Vue фреймворка в початковому вигляді являє собою безліч окремих JavaScript модулів. Крім модулів самого додатка, безпосередньо реалізованих розробником програми існує також велика кількість сторонніх JavaScript модулів і бібліотек. У процесі складання проекту все JavaScript модулі повинні бути об'єднані в єдиний модуль, який надалі буде завантажений клієнтом і підключений до єдиної html-сторінки, щоб в подальшому інтерпретуватися браузером.

Для управління процесами збірки, налагодження і тестування Front-end частини проекту добре підійде система збирання проектів Gulp.

Gulp – потокова система збирання проектів, написаних на JavaScript. Інтегрована в багато середовищ розробки [4]. Дозволяє гнучко управляти складанням проектів, уникнути трудомістких процесів складання, спрощує тестування, покращує структуру програми та зменшує кількість коду. Gulp поширюється через менеджер пакетів NPM під MIT ліцензією. Виконанням збирача також займається програмна платформа Node.JS. Gulp дозволяє створювати і гнучко налаштовувати окремі сценарії для різних завдань від збірки проекту до тестування і розгортання.

При розробці програми використовувався збірщик модулів (англ. Module bundler) Webpack. Розроблюється, складається з різних модулів, що містять код на JavaScript, а також різні модулі CSS стилів. Поточні браузери не мають підтримки модульності, а підключення кожного файлу окремо не має сенсу, тому що в загальному випадку завантаження необхідних ресурсів одним HTTP - запитом відбувається в кілька разів швидше, ніж завантаження за допомогою декількох окремих запитів. Тому виникає необхідність генерації статичних ресурсів програми.

До того ж, розробка програми здійснюється з використанням стандарту ES6 мови JavaScript, який був випущений в 2015 році. Цей стандарт значно розширює синтаксис мови і прискорює розробку, проте на поточний момент не має повної підтримки серед браузерів, отже, необхідна трансляція вихідного коду в стандарт ES5, підтримуваний більшістю браузерів.

Webpack надає всі перераховані раніше переваги. За замовчуванням Webpack забезпечує тільки генерацію єдиного модуля додатка, однак він має підтримку розширень.

Всі необхідні розширення для даного проекту вже були розроблені спільнотою до моменту початку розробки.

у роботі використовуються наступні бібліотеки разом з їх розширеннями для Webpack:

- babel і babel-loader для трансляції коду зі стандарту ES6 в стандарт ES5;
- babel-preset-React для трансляції коду на мові JSX, використовуваному при описі призначених для користувача інтерфейсів засобами Angular в код на JavaScript;
- style-loader і css-loader для завантаження модулів з CSS стилями.

3.7 Обґрунтування вибору CSS-фреймворку Bootstrap

Клієнтом цього додатка є браузер. Завдяки наявності браузерів для практично будь-якої платформи, є можливість реалізувати підтримку як ПК, так і мобільних пристроїв. Як CSS фреймворку був обраний Bootstrap 3.

Bootstrap 3 – вільний набір інструментів для створення сайтів і веб-додатків. Включає в себе HTML і CSS шаблони оформлення для типографіки, веб-форм, кнопок, міток, блоків навігації та інших компонентів веб-інтерфейсів [7].

Основні переваги Bootstrap 3:

- економія часу – Bootstrap дозволяє заощадити час і зусилля, використовуючи шаблони дизайну і класи, і сконцентруватися на інших розробках;
- висока швидкість – динамічні макети Bootstrap масштабуються на різні пристрої і дозволу екрану без будь-яких змін в розмітці;
- простота у використанні;
- сумісність з браузерами – Bootstrap сумісний з Mozilla Firefox, Google Chrome, Safari, Internet Explorer і Opera;
- відкрите програмне забезпечення – особливість Bootstrap, яка передбачає зручність використання за допомогою відкритості вихідних кодів і безкоштовного завантаження.

Bootstrap надає кошти для створення адаптивного дизайну (англ. Responsive design), який, в свою чергу, дозволяє оптимізувати додаток під різні варіанти ширини екрану пристрою.

Завдяки цьому одна і та ж HTML-верстка буде відображатися по-різному в залежності від пристрою, з якого здійснюється перегляд.

3.8 Вибір серверних технологій для реалізації системи

Всі серверні дії буде виконувати додаток написаний за допомогою Node Js фреймворку NestJS, а також бібліотеки TypeORM для швидкої та простої роботи з PostgreSQL. Задача серверної частини – віддавати дані з бази даних у форматі JSON. Для простоти, гнучкості і ефективності був використан підхід REST для будування серверного API. Механізм авторизації використовувати JWT-токени.

Дізнаємось більше інформації про технології та підходи :

1) Фреймворк NestJS - це серверний фреймворк з відкритим кодом Node.js, заснований на Express.js, який використовує переваги TypeScript та JavaScript.

Вперше він був розроблений і запущений Камілем Мислівцем для розробки серверних додатків. В даний час NestJS налічує близько 23 тис. Зірок на Github, а середня швидкість завантажень у тиждень в хвилину становить близько 180 тис.

NestJS сумісний з прогресивним JavaScript і побудований з TypeScript і поєднує елементи цих трьох концепцій:

- ООП (об'єктно-орієнтоване програмування)
- FP (функціональне програмування)
- FRP (функціональне реактивне програмування).

Деякі чудові властивості Nest js:

- розширюваність: гнучкість використання інших бібліотек;
- універсальність: Адаптивна екосистема, повноцінна для всіх типів серверних додатків;
- прогресивність: додає функції JavaScript та шаблони дизайну до спільноти Node.

Nest.js дуже схожий на Angular.js багато в чому, це не здається дивним для тих, хто раніше працював над Angular. Як і Angular, NestJS постачається зі своїм власним контейнером для введення залежностей, утиліта здається забутою в інших рамках. Архітектура та синтаксис також збігаються з архітектурою Angular;

2) Бібліотека TypeORM - це ORM (Object-relational mapping), який може працювати в платформах NodeJS, Browser, Cordova, PhoneGap, Ionic, React Native, NativeScript, Expo та Electron і може використовуватися з TypeScript та JavaScript (ES5, ES6, ES7, ES8). Його мета - завжди підтримувати найновіші функції JavaScript і надавати додаткові функції, які допоможуть вам розробити будь-який тип додатків, що використовує бази даних - від невеликих додатків з кількома таблицями до великих корпоративних додатків з декількома базами даних.

TypeORM підтримує як шаблони Active Record, так і Data Mapper, на відміну від усіх інших ORM-кодів JavaScript, що існують на даний момент, а це означає, що ви можете писати високоякісні, вільно поєднані, масштабовані та ремонтпридатні програми найбільш продуктивним способом.

TypeORM знаходиться під сильним впливом інших ORM, таких як Hibernate, Doctrine та Entity Framework;

3) REST підхід проектування API - скорочення від Representational State Transfer, що можна перекласти як «передача репрезентативного стану». Це стиль проектування розподілених систем за допомогою обмежень. Центральною абстракцією в REST є ресурс, а головні обмеження виглядають так:

- взаємодія без збереження стану;
- логічний інтерфейс.

4) Логічний інтерфейс.

Більшість запитів до API будуть на виконання базових CRUD-операцій (створення-читання-оновлення-видалення). Ця таблиця ілюструє відповідність цих операцій в SQL та HTTP.

OPERATION	SQL	HTTP
CREATE	INSERT	POST
READ	SELECT	GET
UPDATE	UPDATE	PUT/PATCH
DELETE	DELETE	DELETE

Рисунок 3.5 – Схема утворення запиту на сервер.

Приклад базових CRUD-операцій с ресурсом :

- щоб отримати ресурс 1, ви повинні виконати GET/resource/1;
- щоб створити ресурс, ви повинні виконати POST/resource/1;
- щоб замінити ресурс 1 на інший, ви повинні виконати PUT/resource/1;
- щоб частково оновити інформацію про ресурс: PATCH/resource/1;
- щоб видалити ресурс 1: DELETE/resource/1;

5) JWT-токени.

JSON веб-токен (JWT) призначений для передачі підписаних «заявок» (claims) між службами (як зовнішніми, так і внутрішніми для вашого застосунку/сайту). «Заявки» – частина інформації, яку інші можуть переглядати

та/або перевіряти, але не змінювати.

Токени виглядають як рядки з «URL-безпечних» символів, що кодують інформацію. Вони складаються з трьох складових, розділених крапками:

1) Заголовок (Header)

Перша частина JWT — рядок, що закодує звичайний JavaScript об'єкт, який описує токен, а також використаний алгоритм хешування;

2) Корисне навантаження (Payload)

Друга частина JWT формує основу токена. Довжина корисного навантаження пропорційна кількості даних, збережених у JWT. Загальне правило: зберігати мінімум у JWT;

3) Підпис (Signature)

Третя і кінцева частина JWT — підпис, згенерований на основі заголовка та корисного навантаження. Використовується для перевірки JWT.



Рисунок 3.6 – Опис схеми JWT-токена

4 ПРОЕКТУВАННЯ ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ НАВЧАЛЬНИМ ПРОЦЕСОМ. ЕЛЕКТРОННИЙ ЖУРНАЛ

4.1 Основна концепція побудови системи

Для проектування системи і кращого орієнтування треба зробити план-карту сторінок і функціоналу всієї системи. Система ділиться на 3 модуля:

- модуль для адміністратора;
- модуль для вчителів;
- модуль для батьків.

Нижче приведені карти сторінок для кожного з модулів.

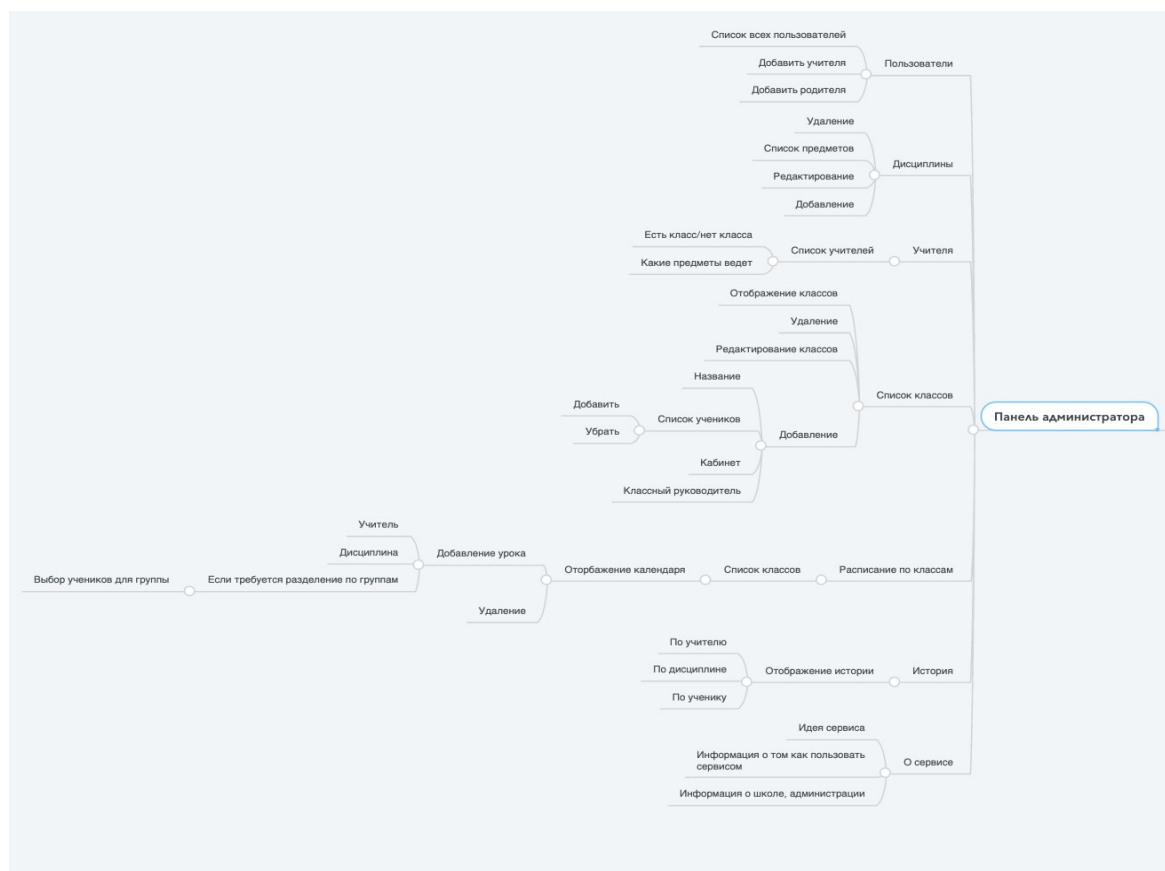


Рисунок 4.1 – План сторінок панелі адміністратора



Рисунок 4.2 – План сторінок модуля для батьків



Рисунок 4.3 – План сторінок модуля для учителя

4.2 Проектування бази даних системи

Для представлення логічної структури системи управління навчальним процесом обрана модель представлення даних «сутність-зв'язок».

Модель «сутність-зв'язок» ґрунтується на важливій семантичній інформації про реальний світ і призначена для логічного представлення даних. Вона визначає значення даних в контексті їх взаємозв'язку з іншими даними.

Сутність (entity) – це об'єкт, який може бути ідентифікований якимсь способом, що відрізняє його від інших об'єктів.

Сутність фактично являє собою безліч атрибутів, які описують властивості всіх членів даного набору сутностей. Безліч значень (область визначення) атрибуту називається доменом.

Зв'язок – це асоціація, встановлена між кількома сутностями.

Кожна сутність повинна містити атрибут або групу атрибутів, які будуть однозначно ідентифікувати кожен екземпляр сутності. Такий атрибут називають первинним ключем.

Виходячи з аналізу вимог до системи управління навчальним процесом необхідно створити базу даних, яка повинна містити в собі наступні дані:

- про користувача;
- про вчителя;
- про учня;
- про батьків;
- про предмети;
- про класи;
- про розклад;
- про коментарі;
- про історію урока.

Уявімо базу даних у вигляді моделі «сутність-зв'язок». Тип використовуваного зв'язку «один-до-багатьох». Діаграма «сутність-зв'язок» представлена на рис.

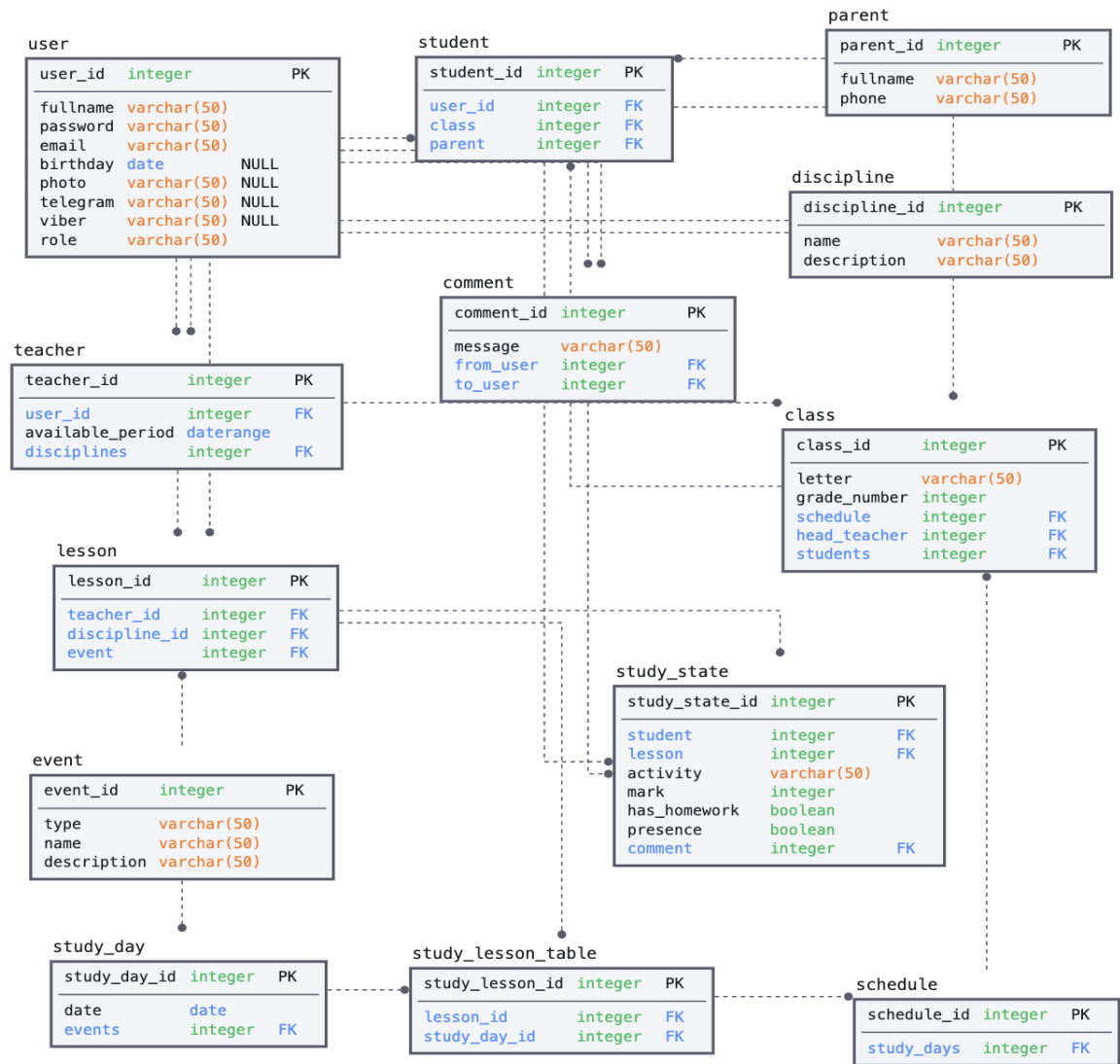


Рисунок 4.4 – Модель «сутність-зв'язок»

4.3 Керівництво користувача-вчителя

Головна сторінка – ця сторінка відображається при першому вході в систему управління навчальним процесом, коли роль користувача – вчитель. Її оформлення і зміст дає перше уявлення про Web-систему.

У верхній частині сторінки розміщена назва системи з логотипом, щоб повідомити користувачеві профіль, призначення та діяльність системи.

Також у верхній частині зображена навігація по сторінкам, список повідомленнями, а також фото користувача, нажавши на яке, він побачить меню з пунктами “Параметри” та “Вихід з системи”.

Домашня сторінка адаптивної системи управління навчальним процесом представлена на рисунку 4.5.

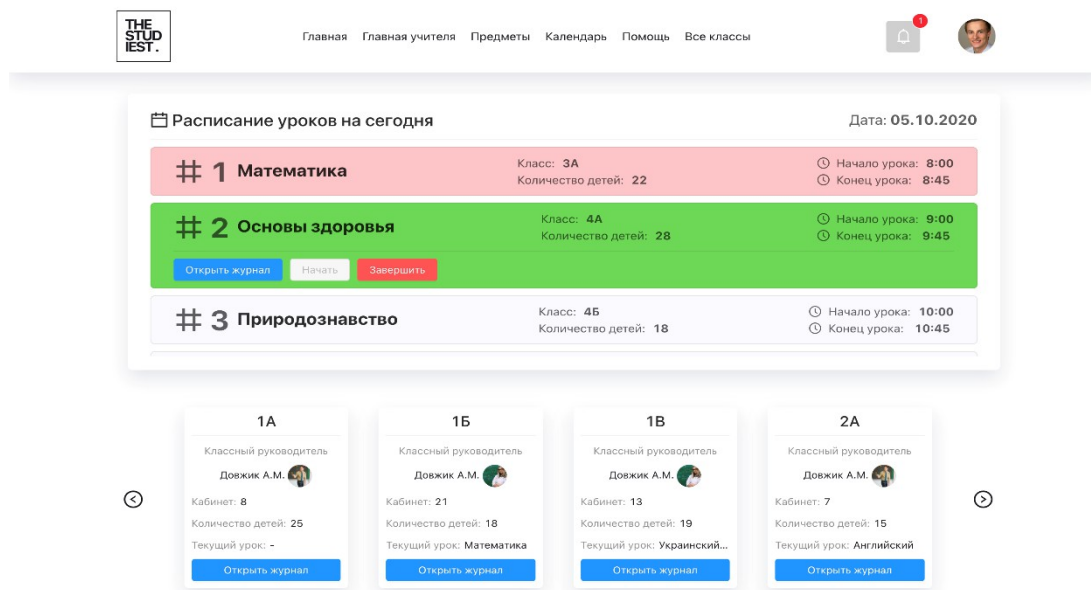


Рисунок 4.5 – Головна сторінка для користувача-вчителя

На сторінці присутній сьогоднішній розклад занять для вчителя. Вчитель бачить перед собою список предметів. Елемент з предметом має 3 стани:

– урок ще не почався;

- урок закінчився;
- урок в процесі.

Кожен стан відображено інтуїтивно-зрозумілим кольором. Натиснувши на кнопку “Відкрити журнал” вчитель потрапляє на сторінку з електронного журналу класа, де проходить заняття. На рисунку 4.6 зображена сторінка с електронним журналом одного з предметів.

The screenshot shows the 'Электронный журнал' (Electronic Journal) page. At the top, there is a navigation bar with links: Главная, Главная учителя, Предметы, Календарь, Помощь, and Все классы. A notification bell icon with a red '1' and a user profile picture are also present. Below the navigation bar, the page title is '← Электронный журнал' and 'Электронный журнал 4Б'. A breadcrumb trail reads: Главная / Список классов / Электронный журнал.

On the left side, there is a teacher profile card for 'Преподаватель' (Teacher) **Полищук Дмитрий Викторович** with email `teacher_email@gmail.com` and a circular profile picture.

The main content is a table with the following columns: ФИО (Full Name), возраст (Age), Адрес (Address), and three columns for grades: 01.0 5, 02.0 5, and 03.0 5. The last column is labeled 'Операции' (Operations) and contains a 'Действия' (Actions) link for each row.

ФИО	возраст	Адрес	01.0 5	02.0 5	03.0 5	Операции
Иванов Иван		Проспект Победы 56	10	10	10	Действия
Петров Петр		Академика Курчатова 12	10	10	10	Действия
Игорев Игорь		Георгиевская 3	10	10	10	Действия
Андреев Андрей		Защитников Украины 5	10	10	10	Действия
Володимиров Владимир		Советская 54	10	10	10	Действия
Борисов Борис		Сумская 73	10	10	10	Действия

At the bottom right of the table, there are pagination controls: '< 1 >'.

Рисунок 4.6 – Сторінка з електронним журналом класу

На цій сторінці, вчитель може побачити оцінки за період навчання, додаткову інформацію, а також залишить коментар, натиснувши на кнопку “Дії”.

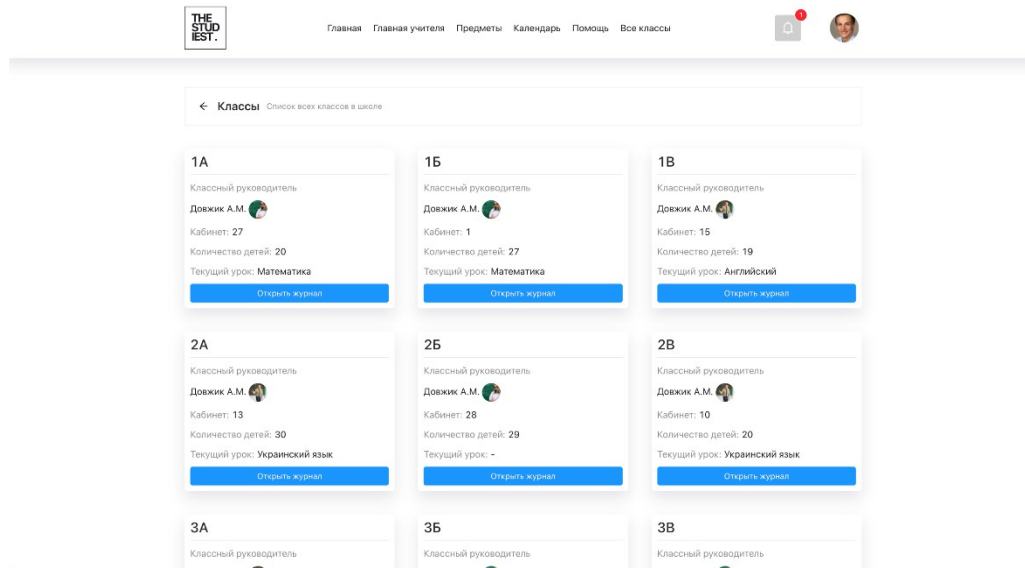


Рисунок 4.7 – Сторінка зі списком класів

На рисунку 4.7 зображена сторінка зі всіма класами, з якими хоча б якось взаємодіє вчитель. Він може перейти і переглянути один з електронних журналів, якогось з класів.

4.4 Керівництво користувачів-батьків

Авторизувавшись в системі, як користувач-батько він потрапляє на головну сторінку зображену на рисунку 4.8. Верхня частина така ж сама, як і у користувача-вчителя, за винятком різної кількості сторінок в навігації.

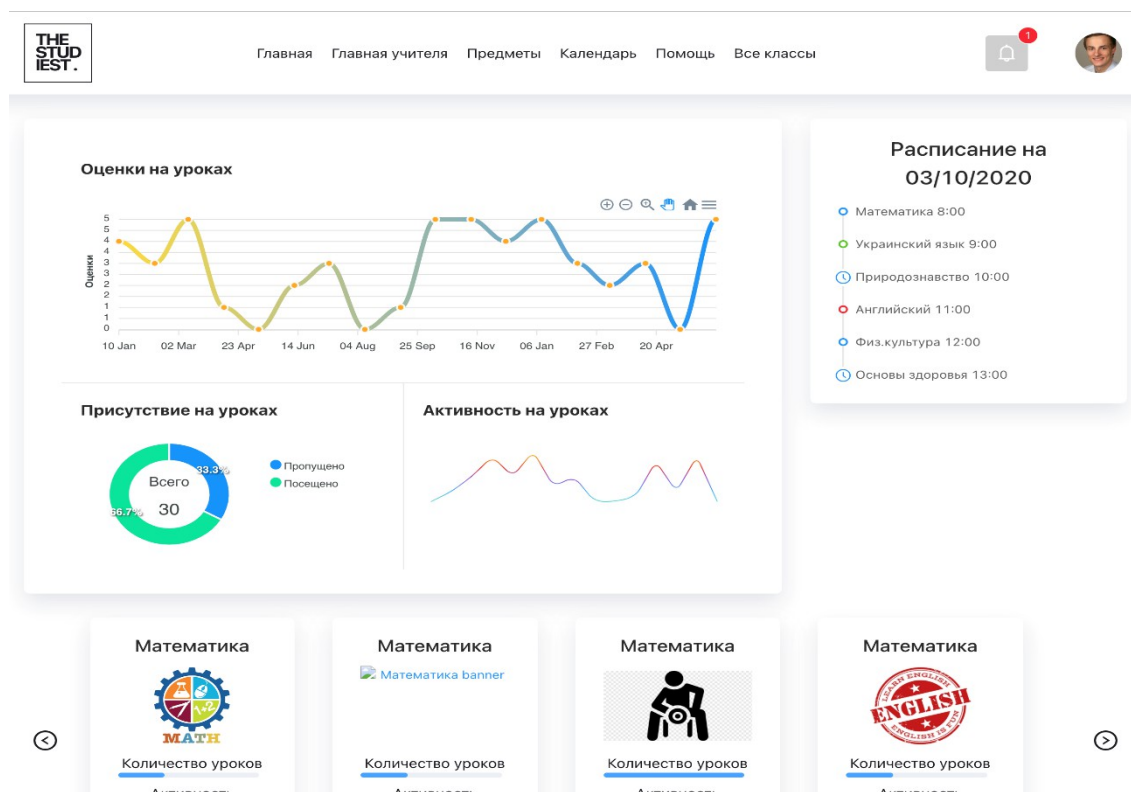


Рисунок 4.8 – Головна сторінка користувачів-батьків

На сторінці є 3 основних елемента:

1) Розклад занять. В ньому показано детальний розклад занять на сьогоднішній день. Завдяки якому батьки зможуть знати як взаємодіяти з дитиною під час навчального процесу;

2) Детальна статистика. В статистичному блоці є 3 метрики: присутність на заняттях, активність на уроках, оцінки на уроках. Батьки можуть завантажити данні в зручній для них формі та наприклад розпечатати її для детального обговорення з дитиною або вчителем;

3) Слайдер з уроками. В ньому зображені всі уроки, які є учня в даному семестрі. Натиснувши на урок він потрапить на сторінку зображену на рисунку 4.9.

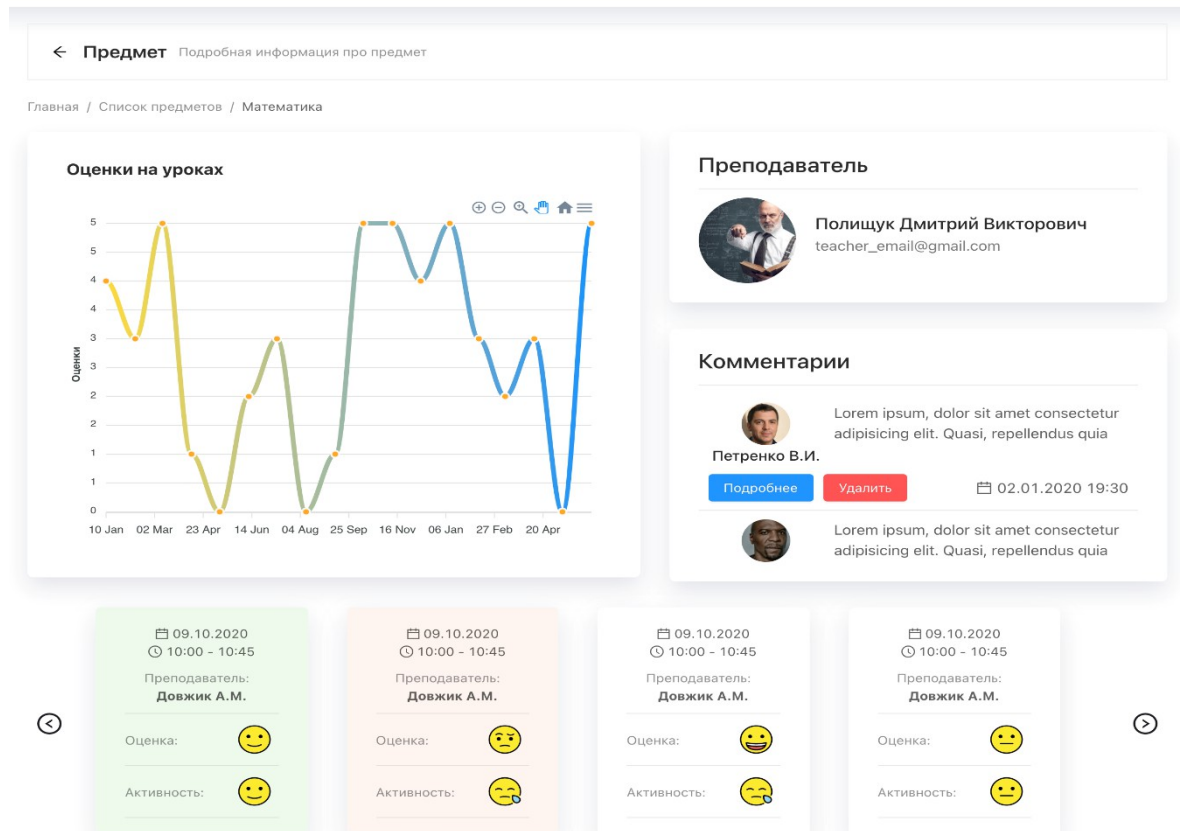


Рисунок 4.9 – Сторінка з детальною інформацією про предмет

На рисунку 4.9 зображена сторінка з детальною інформацією про предмет. На ній можна побачити графік з оцінками, хто викладає даний предмет, коментарі від викладачів, а також слайдер з заняттями, на яких зображені оцінки активності та успіхів у формі Емоїї.

Натиснувши на кнопку “Календарь” в панелі навігації, користувач перейде на сторінку з календарем в якому відображено розклад на семестр. Вона зображену на рисунку 4.10. Користувач може натиснути на один з днів та побачити детальну інформацію про день: наявність контрольних, тестів, відкритих уроків, тощо. Завдяки ній можна бачити весь розклад і не втрачати час на заповнення щоденників.



← Календарь

Календарь с расписанием

2020 ▼ груд ▼ Місяць Рік

пн	вт	ср	чт	пт	сб	нд
30	01	02	03	04	05	06
<ul style="list-style-type: none"> Зарубежная лит... Английский Украинская лите... Английский 	<ul style="list-style-type: none"> Физ.культура Украинская лите... Математика Зарубежная лит... 	<ul style="list-style-type: none"> Украинская лите... Зарубежная лит... Физ.культура Зарубежная лит... 	<ul style="list-style-type: none"> Английский Украинская лите... Украинская лите... Основы здоровья 	<ul style="list-style-type: none"> Английский Физ.культура Физ.культура Английский 		
07	08	09	10	11	12	13
<ul style="list-style-type: none"> Математика Украинская лите... Основы здоровья 	<ul style="list-style-type: none"> Зарубежная лит... Основы здоровья Украинская лите... Физ.культура 	<ul style="list-style-type: none"> Английский Украинская лите... Украинская лите... Основы здоровья 	<ul style="list-style-type: none"> Физ.культура Зарубежная лит... Украинская лите... Основы здоровья 	<ul style="list-style-type: none"> Основы здоровья Физ.культура Основы здоровья Основы здоровья 		
14	15	16	17	18	19	20
<ul style="list-style-type: none"> Английский Основы здоровья Основы здоровья Зарубежная лит... 	<ul style="list-style-type: none"> Английский Зарубежная лит... Украинская лите... Зарубежная лит... 	<ul style="list-style-type: none"> Физ.культура Украинская лите... Украинская лите... Зарубежная лит... 	<ul style="list-style-type: none"> Основы здоровья Украинская лите... Украинская лите... Украинская лите... 	<ul style="list-style-type: none"> Украинская лите... Зарубежная лит... Математика 		
21	22	23	24	25	26	27
<ul style="list-style-type: none"> Основы здоровья Английский Основы здоровья Основы здоровья 	<ul style="list-style-type: none"> Украинская лите... Математика Основы здоровья 	<ul style="list-style-type: none"> Математика Английский Украинская лите... Основы здоровья 	<ul style="list-style-type: none"> Математика Украинская лите... Основы здоровья Зарубежная лит... 	<ul style="list-style-type: none"> Основы здоровья Английский Физ.культура 		
28	29	30	31	01	02	03
<ul style="list-style-type: none"> Английский Математика Зарубежная лит... Физ.культура 	<ul style="list-style-type: none"> Физ.культура Украинская лите... Математика Физ.культура 	<ul style="list-style-type: none"> Английский Математика Английский 	<ul style="list-style-type: none"> Украинская лите... Английский Украинская лите... 	<ul style="list-style-type: none"> Зарубежная лит... Украинская лите... Зарубежная лит... Зарубежная лит... 		
04	05	06	07	08	09	10
<ul style="list-style-type: none"> Английский Основы здоровья Математика 	<ul style="list-style-type: none"> Зарубежная лит... Английский 	<ul style="list-style-type: none"> Физ.культура Украинская лите... Физ.культура 	<ul style="list-style-type: none"> Математика Математика Математика 	<ul style="list-style-type: none"> Математика Украинская лите... Физ.культура 		

Рисунок 4.10 – Сторінка з розкладом

ВИСНОВКИ

Проаналізован сучасний стан освіти у світі та Україні. Також проаналізован вплив пандемії COVID-19 на освітню сферу. Спроектowana та здійснена програмна реалізація системи з застосуванням сучасної та ефективної технології Single Page Application.

Актуальність роботи обґрунтована тим фактом, що під час карантинних заходів багато шкіл не змогли якісно, зручно та ефективно налагодити процес навчання із-за різних факторів, один з яких є відсутність доступних електронних журналів в Україні для повноцінної міграції з оффлайн в онлайн.

Використання сучасної технології SPA для розробки системи дозволило створити систему, яка підтримує кроссплатформенність та відрізняється швидкістю роботи. Кроссплатформенність та швидкість додатків, розроблених за допомогою технології Single Page Application, являється однією з найголовніших переваг відносно подібних додатків, розроблених з використанням інших технологій. Також у додатках, заснованих на такому підході, набагато простіше зберігати різну інформацію, управляти поданням інформації у системі, анімаціями.

В системі управління навчальним процесом, а саме електронним журналом реалізовані механізм аутентифікації користувачів і розмежування прав доступу до інформації. В системі доступні три види користувачів, такі як: адміністратор, користувач-вчитель та користувач-батько.

Завдяки використанню різних елементів управління, система має привабливий, функціональний, зручний і інтуїтивно-зрозумілий інтерфейс користувача з можливістю швидкої модернізації у разі потреби. Застосування технології Single Page Application, дозволило реалізувати достатньо швидко у

користуванні систему, що виділяє її з позиції ресурсоефективності та ресурсозбереження.

Підсумовуючи можна сказати що концепція односторінкових додатків має значні переваги і гарну перспективу розвитку у майбутньому. Це підтверджується значним використанням її при створенні нових веб-сайтів та великою кількістю існуючих. Кількість інструментів, фреймворків та технологій основним використанням яких є розробка односторінкових веб-додатків швидко росте. Це суттєво спрощує створення нових додатків, проте створення великого додатку сьогодні все ще залишається складною задачею. ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Порівняння популярних фреймворків [Електроний ресурс] – Режим доступу: http://cad.kpi.ua/attachments/093_2116d_Voronin.pdf
2. Клієнт-серверна архітектура [Електроний ресурс] – Режим доступу: <http://ukrefs.com.ua/169340-Proektirovanie-lokal-noiy-seti.html>
3. Потенційні споживачі результатів дослідження [Електроний ресурс] – Режим доступу: http://om.net.ua/11/11_13/11_133768_potentsialnie-potrebiteli-rezultatov-issledovaniya.html
4. Технологія QuaD [Електроний ресурс] – Режим доступу: http://om.net.ua/11/11_13/%2011-_133776_tehnologiya-QuaD.html
5. Технологічний аудит: передумови та детальна інструкція з проведення [Електроний ресурс] – Режим доступу: <http://blog.liga.net/user/ytrachuk/article/6072>
6. Відеоуроки з створення Інтернет-систем [Електроний ресурс] – Режим доступу: <http://www.ruseller.com/>
7. Нільсен, Хоа Лоранжер. Технологія Angular 5. – СПб.: Символ-плюс, 2009. – 512 с.
8. Бер Биббо. jQuery. Докладне керівництво по просунутому JavaScript. 2-ге вид. – СПб.: Символ Плюс, 2012. – 624 с.
9. Крістофер Косентіно. SPA технологія. Київ.: BHV, 2016. – 465с.

10. Федорчук А. Як створюються Web-системи за допомогою VueJS. – СПб.: Символ Плюс, 2015. – 224с.
11. Тереза Нейл, Білл Скотт. Проектування веб-інтерфейсів. – СПб.: Символ-Плюс, 2010. – 352 с.
12. Томасон Лаура. Розробка Web-додатків SPA.: Пер. С англ./Лаура Томсон, Люк Веллінг. 2-ге вид., випр. – СПб.: ООО ДияСофтЮП, 2016. – 672 с.
13. Попов Є.Ю. JavaScript та JQuery. Відеоуроки [Електроний ресурс] – Режим доступу: <http://www.evgeniypopov.ru/>
14. Кірсанов Д. Веб-дизайн: книга Дмитра Кірсанова. – СПб.: Символ-плюс, 2009. – 376 с.
15. Single Page Applications Fundamentals [Електроний ресурс] – Режим доступу: <http://www.evgeniypopov.ru/>