

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри _____ Є. О. Давиденко
підпис

«__» _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

**Програмне забезпечення моніторингу успішності здобувачів вищої освіти
Спеціальність «Інженерія програмного забезпечення»**

121 – КРБ.1 – 409.21810927

Студент _____ М. М. Янковий
підпис
«__» _____ 2022 р.

Керівник канд. техн. наук, доцент. _____ Г. В. Горбань
підпис
«__» _____ 2022 р.

Консультант канд. техн. наук, доцент _____ А. О. Алексєєва
підпис
«__» _____ 2022 р.

Миколаїв – 2022

ЗМІСТ

| | |
|---|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ | 4 |
| ВСТУП..... | 5 |
| 1 АНАЛІТИЧНА ЧАСТИНА. ОБГРУНТУВАННЯ ПЛАНУ ВИКОНАННЯ ЗАВДАННЯ | 7 |
| 1.1 Системи управління навчанням..... | 7 |
| 1.2 Аналоги. Переваги та недоліки систем..... | 8 |
| 1.2.1 Canvas | 8 |
| 1.2.2 Google Class | 13 |
| 1.2.3 Moodle..... | 17 |
| 1.3 Специфікація вимог | 22 |
| Висновки до розділу 1 | 23 |
| 2 МОДЕЛЮВАННЯ ОБ'ЄКТУ ТА ПРЕДМЕТУ ДОСЛІДЖЕННЯ..... | 24 |
| 2.1 Проєктування візуальних моделей..... | 24 |
| 2.1.1 Встановлення та сумісність..... | 25 |
| 2.1.2 Ключові особливості..... | 25 |
| 2.2 Користувальницький інтерфейс | 29 |
| Висновки до розділу 2 | 34 |
| 3 АРХІТЕКТУРА, МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ..... | 35 |
| 3.1 Використані технології..... | 35 |
| 3.1.1 Vue.CLI..... | 35 |
| 3.1.2 Materialize.CSS..... | 37 |
| 3.1.3 Firebase | 38 |

| | |
|---|-----------|
| 3.1.4 VSCode | 39 |
| 3.2 Сценарії використання системи..... | 41 |
| 3.3 Діаграми класів, послідовності, компонентів | 45 |
| Висновки до розділу 3 | 48 |
| 4 КОДУВАННЯ, ТЕСТУВАННЯ (ВЕРИФІКАЦІЯ, ВАЛІДАЦІЯ) ПЗ. АНАЛІЗ РЕЗУЛЬТАТІВ РОЗРОБКИ | 49 |
| 4.1 Створення застосунку з Vue CLI 2 | 49 |
| 4.2 Реалізація сторінок та Layouts | 51 |
| 4.2.1 Сторінка «Студенти» | 54 |
| 4.2.2 Сторінка «Курси»..... | 55 |
| 4.2.3 Сторінка «Новий запис» | 57 |
| 4.2.4 Сторінка «Огляд курсу» | 59 |
| 4.2.5 Сторінка «Історія записів» | 60 |
| 4.3 Firebase | 63 |
| 4.4 Захист роутів..... | 65 |
| 4.5 Динамічна локалізація (ua/en)..... | 66 |
| Висновки до розділу 4 | 67 |
| ВИСНОВКИ..... | 68 |
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ..... | 69 |

Кафедра інженерії програмного забезпечення
Програмне забезпечення моніторингу успішності здобувачів вищої освіти

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

ЗВО – заклад вищої освіти (institution of higher education)

ЗВО – здобувач вищої освіти (higher education seeker)

LMS – Learning Management Systems (системи управління навчанням)

СУО – Системи управління навчанням

ВСТУП

Система забезпечення та підвищення якості освіти у ЗВО повинна будуватися на свідомому, спільному створенні культури якості всіх представників академічної спільноти. Досягнення високої якості освіти нерозривно пов'язане з автономією, традиціями та гідністю академічної спільноти.

Законодавчою базою формування системи внутрішнього забезпечення якості виступає Закон України «Про вищу освіту». За вимогами Закону система внутрішнього забезпечення якості є одним з трьох елементів системи забезпечення якості вищої освіти. Аналіз процедур та заходів системи внутрішнього забезпечення якості у ЗВО дозволив виявити ті елементи, що вже сформовані та ті, що потребують впровадження та (або) удосконалення:

- визначення принципів та процедур забезпечення якості вищої освіти;
- здійснення моніторингу та періодичного перегляду освітніх програм;
- щорічне оцінювання здобувачів вищої освіти, науково-педагогічних і педагогічних працівників ЗВО та регулярне оприлюднення результатів таких оцінювань на офіційному вебсайті ЗВО, на інформаційних стендах та в будь-який інший спосіб;
- забезпечення підвищення кваліфікації педагогічних, наукових і науково-педагогічних працівників;
- забезпечення наявності необхідних ресурсів для організації освітнього процесу, для самостійної роботи здобувачів за кожною освітньою програмою;
- забезпечення наявності інформаційних систем для ефективного управління освітнім процесом;
- забезпечення публічності інформації про освітні програми, ступені вищої освіти та кваліфікації;
- забезпечення дотримання академічної доброчесності працівниками закладів вищої освіти та здобувачами вищої освіти, у тому числі створення і забезпечення функціонування ефективного системи запобігання та виявлення академічного плагіату;

– інші процедури і заходи.

Об'єктом кваліфікаційної роботи є методи моніторингу успішності здобувачів вищої освіти.

Предметом кваліфікаційної роботи є технології створення програмного забезпечення моніторингу успішності ЗВО.

Актуальність теми кваліфікаційної роботи полягає в створенні комфортних умов для навчання здобувачів вищої освіти та полегшенню ведення навчального процесу у ЗВО.

Метою кваліфікаційної роботи є поліпшення навчального процесу у ЗВО шляхом розробки програмного забезпечення для моніторингу успішності ЗВО.

Для досягнення мети треба вирішити наступні задачі:

- провести аналіз аналогічних програмних застосунків зі схожим функціоналом;
- обрати найкращі технології для розробки застосунку;
- сформулювати вимоги застосунку для оптимізації на всіх платформах;
- розробити архітектуру бази даних;
- реалізувати застосунок.

1 АНАЛІТИЧНА ЧАСТИНА. ОБГРУНТУВАННЯ ПЛАНУ ВИКОНАННЯ ЗАВДАННЯ

1.1 Системи управління навчанням (LMS)

Програмні сервіси та системи управління навчанням (СУО, англ. Learning Management Systems, LMS) допомагають освітнім установам надавати та реалізовувати освітні курси або навчальні програми[1].

Щоб програмний продукт був включений до категорії програмних систем управління навчанням, система повинна задовольняти критеріям:

- дозволяти утримувати курси та навчальні матеріали у централізованій системі, доступній для здобувачів з метою навчання;
- зберігати звіти про перебіг навчання окремих учасників та про продуктивність навчальних програм загалом;
- давати можливість налаштовувати програми навчання відповідно до індивідуальних потреб;
- надавати можливості побудови планів та розкладів, та відстеження проходження навчальних курсів та дисциплін;
- пропонувати можливості розробки курсів або інтеграції зі сторонніми програмами для розробки курсів.

Методи і технології моніторингу вельми різноманітні. Багато в чому вони залежать від кваліметричної підготовки персоналу та можливостей технічних систем. В якості основних слід виділити наступні **методи моніторингових досліджень**:

- метод порівняльних оцінок (порівняння з нормами якості освіти як базами оцінки);
- метод експертних оцінок;
- аналіз документальних джерел, наприклад, змістовний і контент-аналіз освітніх програм та інших документів;
- аналіз результатів предметних тестів і даних успішності;

- збір і аналіз статистичних даних про різні види діяльності освітніх систем;
- аналіз баз і банків даних;
- імітаційне моделювання на електронно-обчислювальних машинах;
- соціологічні методи (соціологічне і соціометричне опитування, факторний і кореляційний аналіз, соціальне прогнозування).

Технології організації моніторингу можуть бути різними, наприклад, подання статистичних даних і звітів за певні періоди діяльності або накопичення оцінок і визначення рейтингу здобувачів в рейтингових системах обліку успішності. Так як дискретність збору інформації в ході моніторингу може бути дуже різною (протягом уроку, щодня, щотижня, протягом чверті, півріччя, навчального року), то неоднакові і обсяги цієї інформації. У будь-якому випадку повинен дотримуватися розумний баланс між необхідною періодичністю збору даних, їх обсягом і необхідністю подання до вищого органу управління.

Форми представлення результатів моніторингу:

- сформовані бази даних про функціонування освітніх систем різного рівня, які постійно поповнюються;
- статистичні та наукові звіти.

Дані про моніторинг можуть розміщуватися на їх сайтах, публікуватися в шкільних і муніципальних виданнях, в таких формах вони доступні для батьків, потенційних роботодавців, можливих соціальних партнерів.

1.2 Аналоги. Переваги та недоліки систем

1.2.1 Canvas

Canvas – це система управління навчанням, що допомагає організувати навчальний процес з урахуванням нових інформаційних технологій [2].

Програмний продукт Canvas (укр. Канвас) від компанії Instructure призначений для дистанційного навчання. Система Канвас застосовується академічними та освітніми установами (школами, коледжами, інститутами,

Кафедра інженерії програмного забезпечення
Програмне забезпечення моніторингу успішності здобувачів вищої освіти
університетами), а також навчальними підрозділами у компаніях. Це програмне
забезпечення може застосовуватися з використанням веббраузерів, так само як і
встановлено на мобільні пристрої Android та iOS (рис. 1.1).

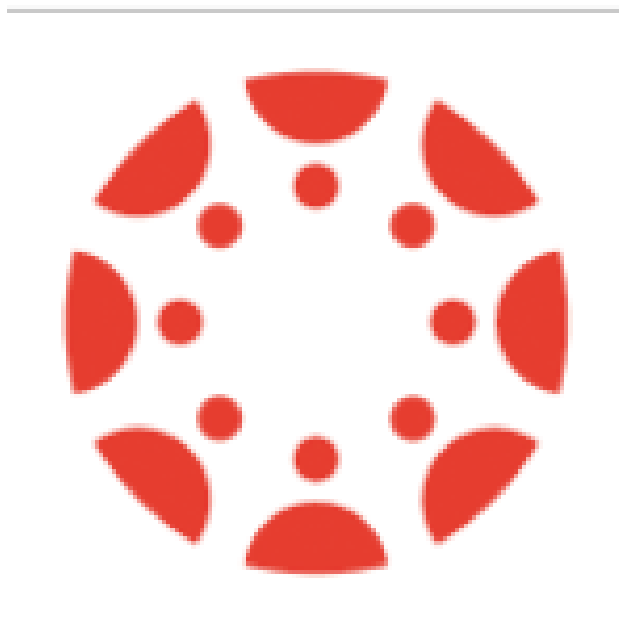


Рисунок 1.1 – Логотип Canvas

Програмна система Canvas реалізує основні функції управління навчанням, у тому числі управління класом, управління випробуваннями (тестові будівлі, заліки), спільної роботи та організації навчальних матеріалів.

Підходить для: Середній бізнес, Спеціаліст, Корпорація.

Розгортання: Мобільний пристрій, Сервер підприємства.

Графічний інтерфейс: Веб-браузер, iOS, Android.

Тарифування: Безкоштовно.

Підтримка мов: Російська, Англійська, Арабська, Угорська та ще 14.

Демо версія: Так, ознайомитися безкоштовно.

Пробна версія (тріал): Так.

Інформаційну панель зображено на рис. 1.2:

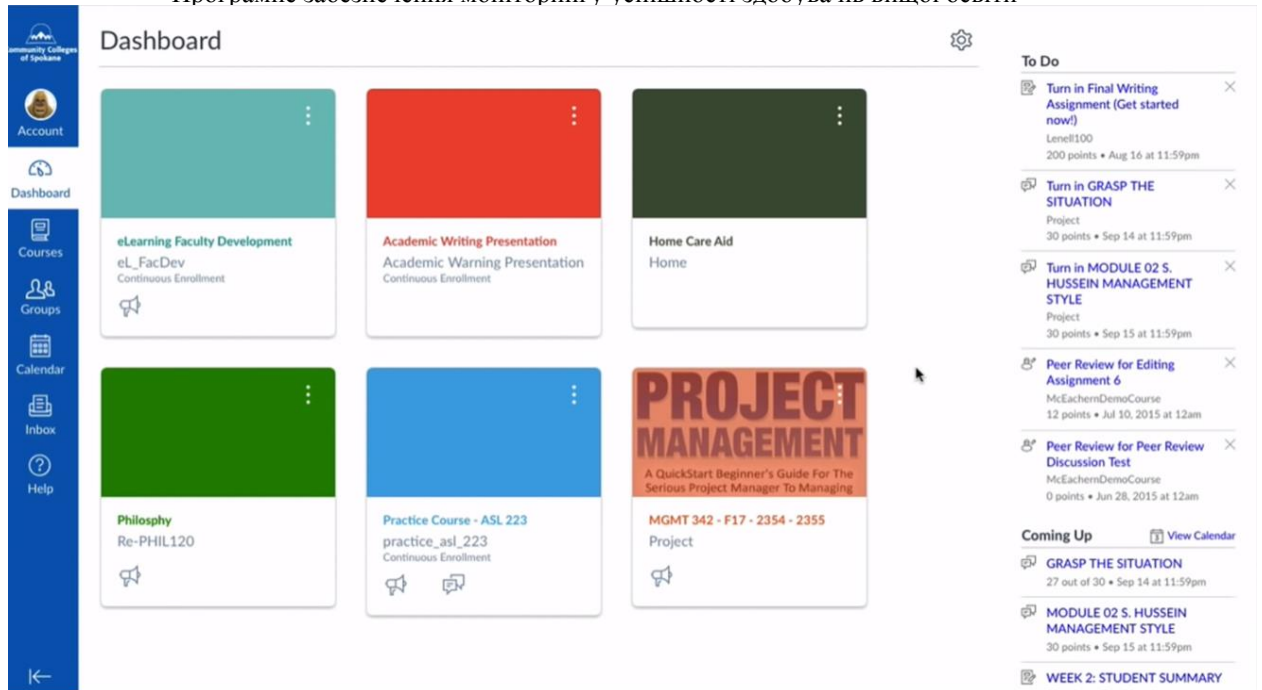


Рисунок 1.2 – Інформаційна панель

Функції Canvas

- прокторинг;
- управління класом;
- розробка матеріалів курсу;
- робота з журналом оцінок;
- імпорт/експорт даних;
- відеоконференції;
- адміністрація;
- наявність API;
- звітність та аналітика;
- розрахований на багато користувачів доступ;
- відповідність SCORM;
- відеоуроки;
- ігрофікація;
- синхронне навчання;
- управління базою знань;

- тестування та оцінка знань;
- асинхронне навчання;
- індивідуальне керування навчанням.

Програмне забезпечення Canvas включає наступні основні можливості:

- створення багатого та привабливого вмісту курсів, в т. ч. через текст, зображення, рівняння, відео тощо;
- імпорт раніше розроблених навчальних курсів;
- організація онлайн-дискусій та обговорень, що дозволяють учням глибше занурюватися у зміст курсів;
- Вимірювання та відстеження викладачем прогресу учнів для прийняття своєчасних та обґрунтованих навчальних рішень;
- організація та участь у синхронних онлайн-комунікаціях (вебінари, конференції);
- доступ батьків/керівників до даних про прогрес навчання для своєчасної допомоги дитині/підлеглому.

Систему оцінювання можна переглянути на рис. 1.3:

The screenshot shows the 'Grades' page for the course 'Project Management'. The page title is 'Grades for Skitch the Bigfoot'. The course is set to 'Project Management' and is arranged by 'Due Date'. A table lists various assignments with their due dates, scores, and maximum possible scores. A summary on the right shows a total score of 90%.

| Name | Due | Status | Score | Out of |
|---|-------------------|--------|-------|--------|
| GRASP THE SITUATION | Sep 14 by 11:59pm | | 27 | 30 |
| MODULE 02 S. HUSSEIN MANAGEMENT STYLE | Sep 15 by 11:59pm | | - | 30 |
| WEEK 2: STUDENT SUMMARY | Sep 18 by 1pm | | - | 50 |
| Your Opinion on Key Influencers | Sep 22 by 12pm | | - | 30 |
| WEEK 3: STUDENT SUMMARY | Sep 25 by 1pm | | - | 50 |
| PROJECT LIFECYCLE - 5 PHASES | Sep 29 by 12pm | | - | 30 |
| Enterprise Environmental Factors versus Organizational Process Assets | Sep 29 by 11:59pm | | - | 30 |
| MODULE 7 QUIZ | Oct 1 by 11:59pm | | - | 20 |
| MODULE 8 QUIZ | Oct 1 by 11:59pm | | - | 20 |
| MODULE 09 QUIZ | Oct 8 by 11:59pm | | - | 20 |
| WEEK 6 STUDENT SUMMARY | Oct 8 by 11:59pm | | - | 50 |

Summary:
Total: 90%
Assignments are weighted by group:
Group | Weight
Student Discussion | 60%
Student Summary | 10%
Assignments | 20%
Imported Assignments | 0%
Total | 90%

Calculate based only on graded assignments
You can view your grades based on What-If scores so that you know how grades will be affected by upcoming or resubmitted assignments. You can test scores for an assignment that already includes a score, or an assignment that has yet to be graded.

Рисунок 1.3 – Виставлення оцінок у програмній системі

Кафедра інженерії програмного забезпечення
Програмне забезпечення моніторингу успішності здобувачів вищої освіти

Перелік навчальних курсів та інформацію про розробника можна переглянути на рис. 1.4 – 1.5:

| Course | Nickname | Term | Enrolled as | Published |
|---------------------------------------|-------------------------------|-----------------------|-------------|-----------|
| ☆ A New and Exciting Shell | | Continuous Enrollment | Student | No |
| ☆ Development Course - Ruth Perkins | | Winter 2013 | Student | No |
| ☆ Development Course - Greg Cain | | Continuous Enrollment | Student | No |
| ★ eLearning Faculty Development | | Continuous Enrollment | Student | Yes |
| ☆ 2017 eLearning Academy | | Continuous Enrollment | Student | Yes |
| ★ Academic Writing Presentation | Academic Writing Presentation | Continuous Enrollment | Student | Yes |
| ☆ demo101 EVAL | | | Student | Yes |
| ☆ Development Course - Mark Lenell | | Continuous Enrollment | Student | Yes |
| ☆ GESLO Critical Thinking 2016/2017 | | | Student | Yes |
| ☆ Hogan master6 | | | Student | Yes |
| ★ Home Care Aid | | | Student | Yes |
| ☆ Learning Mastery Gradebook Practice | | Continuous Enrollment | Student | Yes |

Рисунок 1.4 – Перелік навчальних курсів

| | |
|---------------------------------|---|
| Розробник системи Canvas | |
| Компанія-розробник | Веб-сайт (продукт) |
| Інструкція | https://www.canvaslms.com |
| Збоку | Веб сайт |
| США | https://www.instructure.com |
| Головний офіс | Соціальні мережі |
| Солт-Лейк-Сіті, Юта | Facebook YouTube Twitter Instagram LinkedIn |

Рисунок 1.5 – Інформація про розробника

Переваги:

- наявність сучасних інструментів для створення курсів;
- відкритий API і, завдяки йому, – підтримка численних сторонніх розширень та інтеграцій, що дозволяє розширити функціонал навіть без наявності ІТ-знань;
- зручність використання вкладок і контекстного меню в інтерфейсі;
- посилена безпека;

– можливість для мобільного навчання.

Недоліки:

– Відсутність гейміфікації.

Однак компанія-розробник Instructure також представила Arc – інклюзивну відеоплатформу для онлайн-навчання, що легко може використовуватися з Canvas для завантаження та обміну відео в цифровому класі, і дозволяє студентам і викладачам коментувати відео, створюючи бесіди.

1.2.2 Google Class

Google Клас – це освітній інструмент, який допомагає вчителям швидко створювати та керувати навчальними завданнями, забезпечувати зворотний зв'язок та спілкуватися зі своїми класами [3].

Програмний продукт Google Клас (англ. Google Classroom, укр. Google Клас) об'єднує в одному інструменті послуги компанії Google для використання з метою навчання на комп'ютері (рис. 1.6):

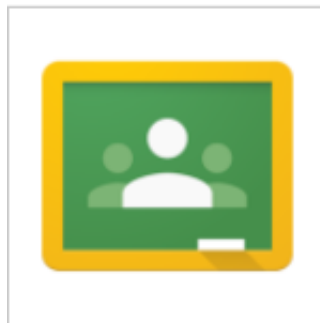


Рисунок 1.6 – Лого Google Class

На програмній платформі Google Classroom можна виконувати основні функції керування навчанням (рис. 1.7):

- Створення та управління класами;
- Створення курсів;
- Організація запису учнів на курс;
- Надання доступу до навчальних матеріалів для учнів;
- Публікація завдань для учнів (рис. 1.8);

- Оцінка завдань учнів, проведення тестів та відстеження прогресу навчання;
- Організація взаємодії учасників навчального процесу.

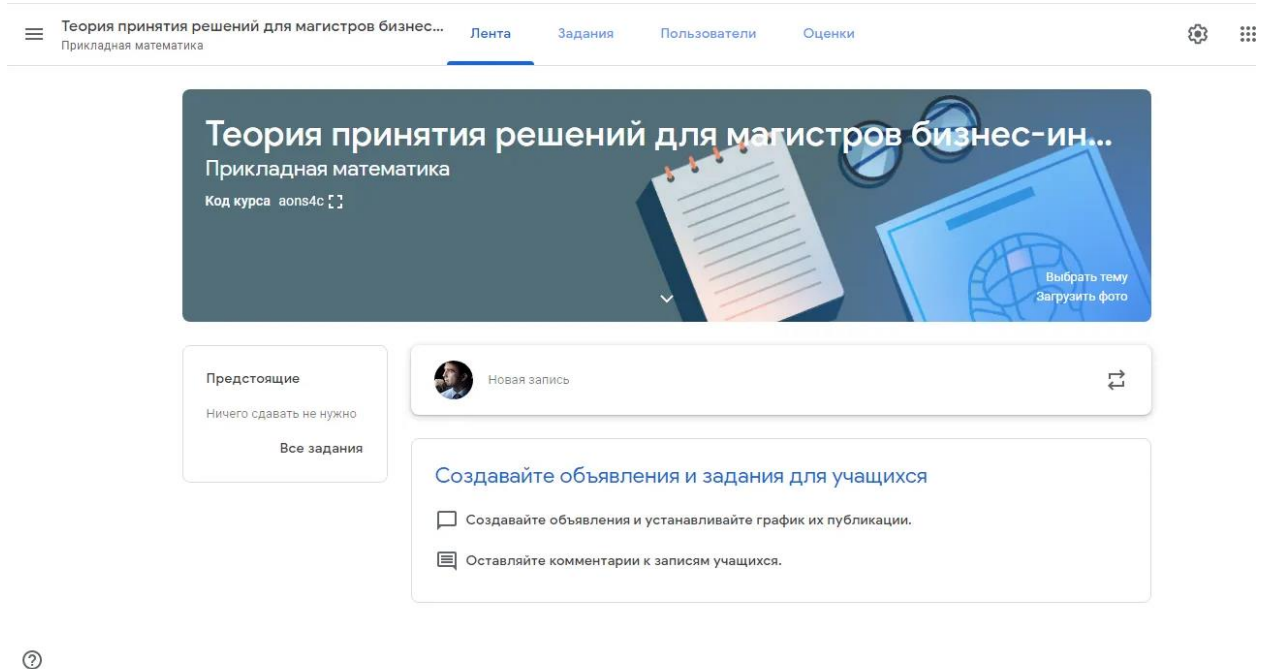


Рис. 1.7 – Перегляд стрічки подій в освітній системі

Підходить для: Фрілансер, Середній бізнес, ІП, Фахівець, Корпорація, Малий бізнес.

Розгортання: Хмара (SaaS).

Графічний інтерфейс: Веб-браузер.

Тарифікація: Безкоштовно.

Підтримка мов: Українська.

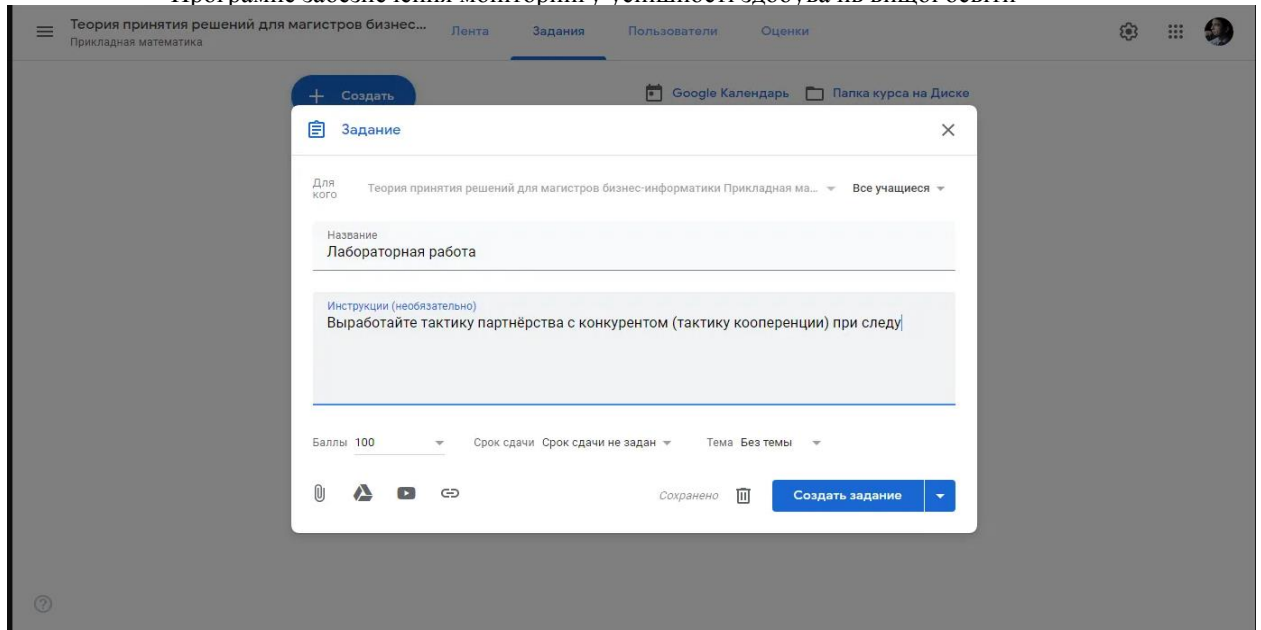


Рисунок 1.8 – Створення завдань

Функції Google Клас

- імпорт/експорт даних;
- управління класом;
- відео-конференції;
- розробка матеріалів курсу;
- робота з журналом оцінок;
- асинхронне навчання;
- відеоуроки;
- адміністрація;
- синхронне навчання;
- прокторинг;
- наявність API;
- відповідність SCORM;
- управління базою знань;
- звітність та аналітика;
- розрахований на багато користувачів доступ;
- ігрофікація;

Кафедра інженерії програмного забезпечення
Програмне забезпечення моніторингу успішності здобувачів вищої освіти

- індивідуальне керування навчанням;
- тестування та оцінка знань.

Можливість додавання викладача до курсу та інформація про розробника зображена на рис. 1.9 – 1.10:

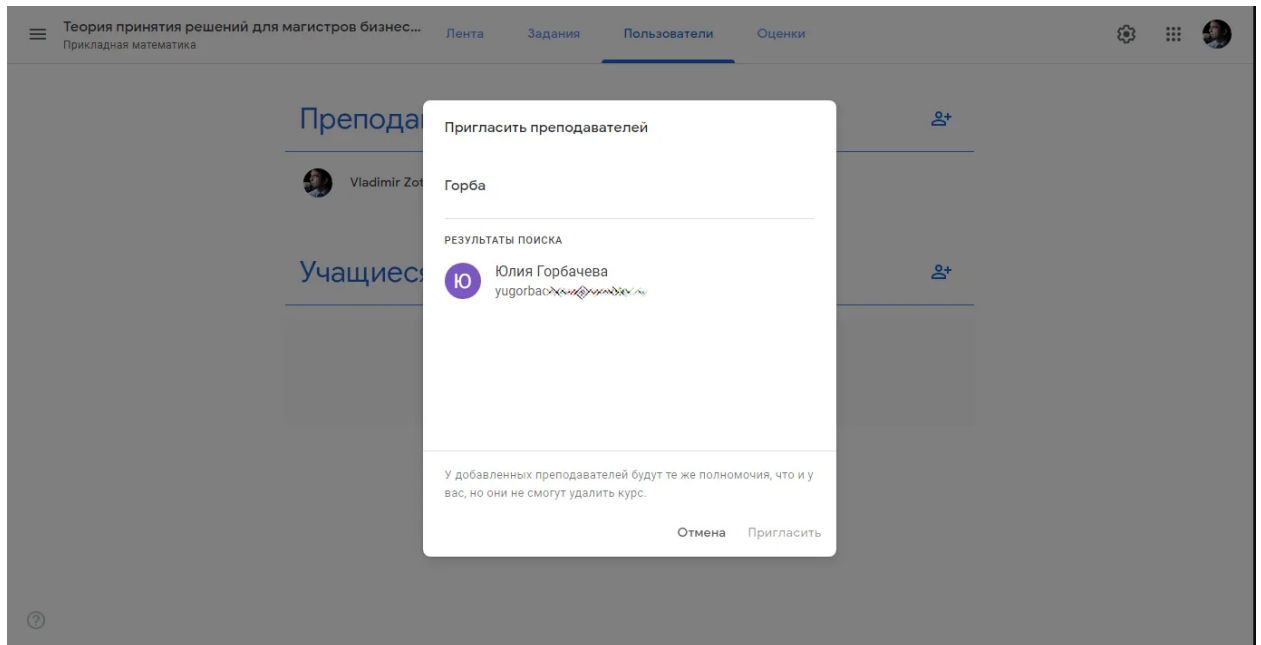


Рисунок 1.9 – Додавання викладача

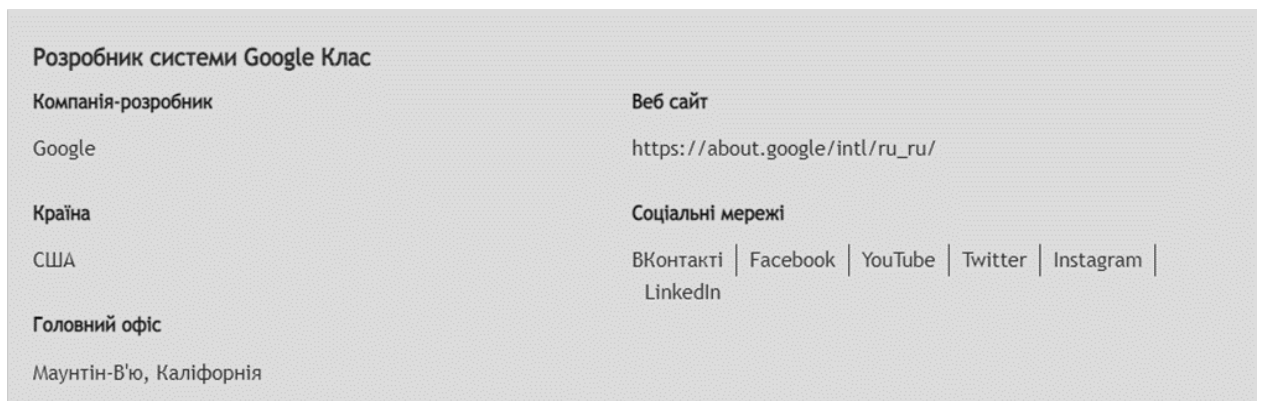


Рисунок 1.10 – Інформація про розробника

Переваги:

- сервіс безкоштовний, якщо не ведеться комерційна діяльність, тобто не здійснюється продаж своїх курсів;
- інтуїтивно зрозумілий інтерфейс;

- сервіс сам підказує – як створити курс або тест, де подивитися результати учнів, – щоб швидше освоїти його можливості;
- декілька інструментів Google в одному. Classroom включає диск для зберігання файлів, Google Docs для публікації текстових лекцій, презентації, опитування, сервіс для відеозустрічей і календар для планування навчання;
- командну роботу над курсом можуть створювати одразу кілька вчителів: поки що один пише лекцію, інший збирає тест;
- регулярний зворотний зв'язок з учнями – викладач може коментувати кожне завдання та тест;
- батьки можуть бачити оцінки дітей, щоб контролювати їхню успішність – звіт із відмітками прийде їм на пошту.
- є мобільний застосунок для створення курсів та навчання.

Недоліки:

- не підтримує стандарти електронного навчання: SCORM, Tin Can (xAPI) та cmi5. Курси, зібрані за цими стандартами, не працюють у Classroom;
- у тестах немає захисту від списування. Не можна обмежити час на відповідь та кількість спроб, штрафувати за помилки або запустити перемішування питань, щоб у кожного учня вони відображалися у випадковому порядку. Самі тести більше нагадують опитування;
- повільна технічна підтримка англійською. Інженери відповідають на запитання протягом двох-трьох днів, як пощастить;
- вчитися у Classroom можуть лише користувачі пошти Google;
- можливості дизайну обмежені;
- у безкоштовній версії можна навчати лише 200 осіб.

1.2.3 Moodle

Moodle – це гнучка система дистанційного навчання, що налаштовується, з відкритим вихідним кодом, що допомагає задовольнити широкі потреби в організації навчання для освітніх закладів [4].

Програмний продукт Moodle (укр. Мудл) від однойменної австралійської компанії є однією з найбільш популярних зараз безкоштовних систем управління навчанням та систем дистанційного навчання(рисунок 1.11):



Рисунок 1.11 – Лого Moodle

Програмне забезпечення moodle надає викладачам у всьому світі рішення для електронного дистанційного навчання, яке масштабується, налаштовується і забезпечує безпеку з одним з найбільших наборів функцій. СУО Мудл підтримується та розвивається активною мережею сертифікованих партнерів та спільнотою розробників.

Підходить для: Фрілансер, Середній бізнес, Фахівець, Корпорація, Малий бізнес.

Розгортання: Мобільний пристрій, Персональний комп'ютер, Сервер підприємства, Хмара (SaaS).

Графічний інтерфейс: macOS, Веб-браузер, iOS, Windows, Android.

Тарифікація: Безкоштовно, Щорічна оплата.

Підтримка мов: Російська, Англійська, Азербайджанська, Арабська та ще 26.

Демо версія: Так.

Пробна версія (тріал): Так.

Вільне ПЗ: Так.

Навчальний календар зображено на рис. 1.12:

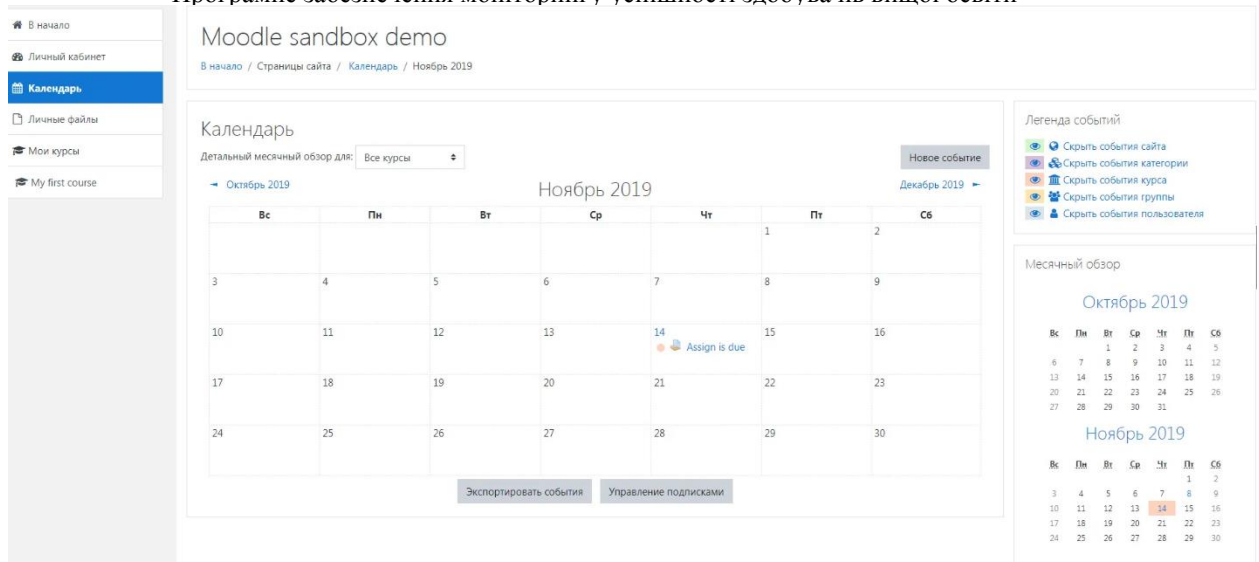


Рисунок 1.12 – Навчальний календар

Moodle є програмним забезпеченням з відкритим вихідним кодом, однак надається можливість використання з хмари компанії-розробника. У складі програмного комплексу є крім основної клієнт-серверної системи, програми для персонального комп'ютера та для мобільних пристроїв.

Програмне розширення Moodle Workplace дає особливі можливості для користувачів:

- Організація процесів навчання у компанії;
- Координація та управління знаннями;
- Управління компетенціями;
- Налаштування під процеси та організаційну структуру компанії.

Управління навчальними матеріалами зображено на рис. 1.13:

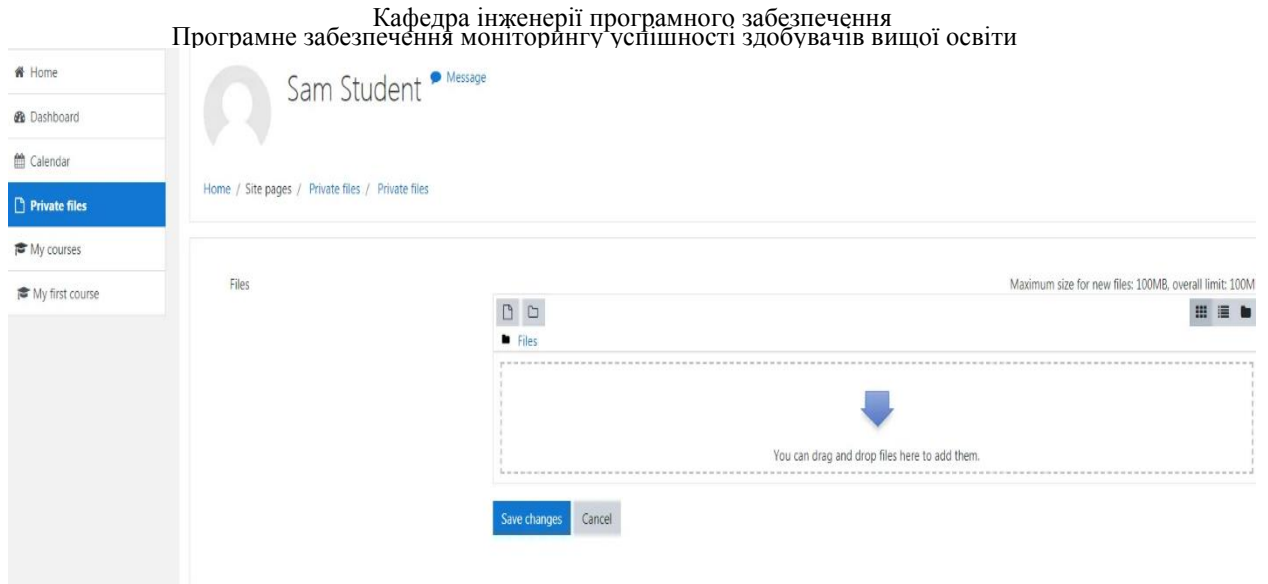


Рисунок 1.13 – Управління навчальними матеріалами

Функції Moodle:

- звітність та аналітика;
- управління класом;
- ігрофікація;
- відповідність SCORM;
- робота з журналом оцінок;
- прокторинг;
- відео-конференції;
- асинхронне навчання;
- наявність API;
- управління базою знань;
- розробка матеріалів курсу;
- відеоуроки;
- синхронне навчання;
- розрахований на багато користувачів доступ;
- індивідуальне керування навчанням;
- імпорт/експорт даних;
- тестування та оцінка знань.

Переваги:

- Moodle має вбудований конструктор курсів (рис. 1.14).
- Підтримує, серед іншого, асинхронне і змішане навчання, гейміфікацію, мобільне навчання, стандарти SCORM, синхронне навчання, відеоконференції і багато іншого.
- Безліч плагінів.
- Відкритий вихідний код.

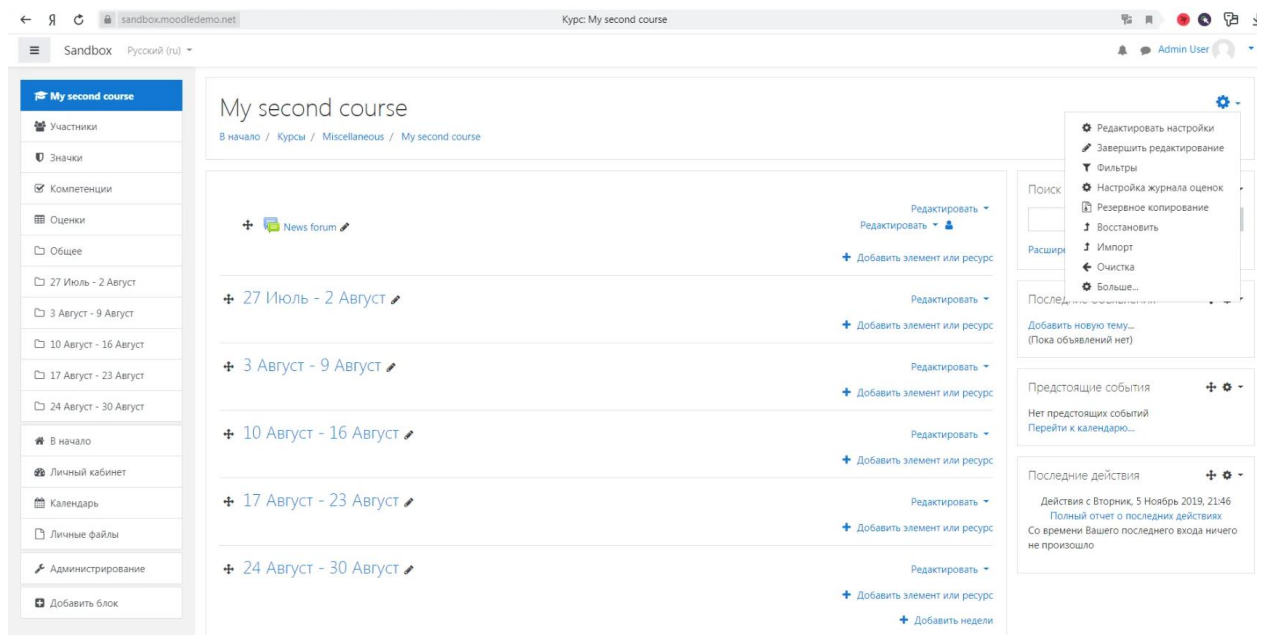


Рисунок 1.14 – Створення навчального курсу

Недоліки:

- Архітектура системи дуже складна і допрацювати самостійно Moodle можна, але в якийсь момент ви зрозумієте, що легше було написати з нуля.
- Інтерфейс (UI) системи залишає бажати кращого.

1.3 Специфікація вимог

Призначення застосунку: поліпшення навчального процесу у ЗВО.

Сфера застосування: застосунок може бути використаний у всіх навчальних закладах освіти.

Характеристики користувачів: застосунок призначений для усіх учасників навчального процесу, зокрема спрямований на викладачів, для поліпшення роботи.

Склад системи: Вхід/Реєстрація користувача, головна сторінка (Студенти), вкладка «Профіль», сторінки: «Історія записів», «Огляд курсу», «Новий запис», «Курси».

Вимоги до способів організації, збереження та введення інформації: інформація повинна швидко знаходити та повинна бути захищена від стороннього втручання шляхом шифрування.

Джерела і зміст вхідної інформації (даних): всі дані зберігаються в БД на хмарі.

Основні функції системи:

1. Створення та редагування курсів(дисциплін).
2. Додавання нового запису з кількістю робіт та прізвищем студента.
3. Огляд створених курсів та динамічне заповнення кількості їх робіт.
4. Моніторинг створених записів та відображення у вигляді графіка.
5. Детальний перегляд записів у таблиці.
6. Відображення списку студентів групи та фільтрація за певними параметрами.
7. Динамічна локалізація інтерфейсу.

Навички: для правильного користування застосунком необхідні базові знання комп'ютера та доступ до мережі Інтернет.

Рекомендації: оновлений браузер до останньої версії.

Інтерфейс користувача: застосунок розроблений з використанням новітніх технологій, що забезпечує простий користувальницький інтерфейс та зручність у користуванні.

Вимоги до системи: система повинна працювати без збоїв.

Висновки до розділу 1

У розділі було розглянуто приклади аналогів розроблюваного програмного застосунку. Наведено їх функціонал, інтерфейс та детальну інформацію.

Було проведено окрему роботу щодо визначення переваг та недоліків застосунків-аналогів, що допоможе використати це у розробці майбутнього застосунку.

Визначено основні функції кожного застосунку, які повинні бути взяті до уваги при розробці застосунку.

Досліджено:

- основні технології моніторингу;
- методи моніторингових досліджень;
- форми представлення результатів моніторингу.

Кафедра інженерії програмного забезпечення
Програмне забезпечення моніторингу успішності здобувачів вищої освіти

2 МОДЕЛЮВАННЯ ОБ'ЄКТУ ТА ПРЕДМЕТУ ДОСЛІДЖЕННЯ

2.1 Проектування візуальних моделей

BPWin – це спеціальна програма, яка забезпечує проектування візуальних моделей. У програмі можна створити шаблон для бізнес-плану та презентаційний документ. BPWin забезпечує створення моделей для «корпоративів» та презентацій [9].

Це програмне забезпечення створює моделі бізнес-процесів у «візуальній» оболонці для кращого сприйняття інформації. Візуальна модель полегшує розуміння різних процесів та зв'язку, які створені між ними. Графічна оболонка програми не складна та розрахована на недосвідчених користувачів. Інтерфейс BPwin зображено на рис 2.1:

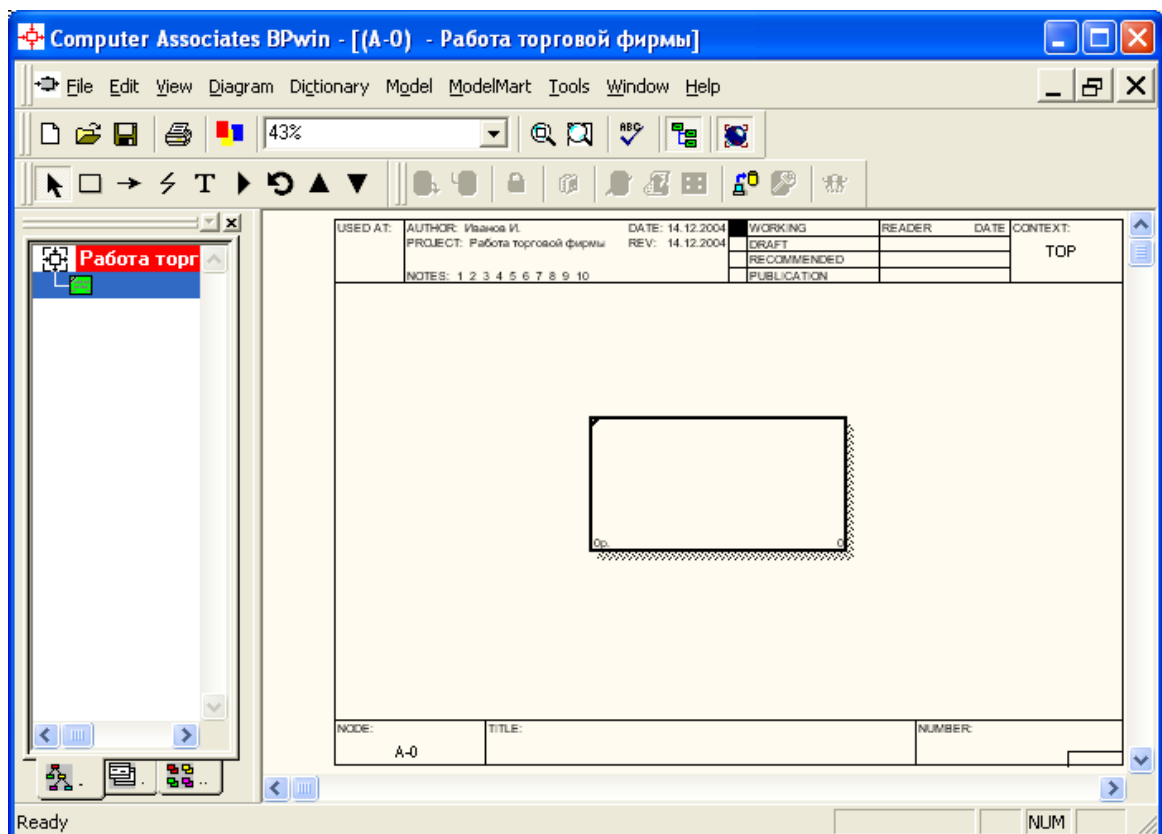


Рисунок 2.1 – Інтерфейс BPwin

BPWin – це одне з найстаріших CASE-рішень, яке використовує технологію IDEF0. Використовуючи інструменти, що входять до складу програмного

Кафедра інженерії програмного забезпечення
Програмне забезпечення моніторингу успішності здобувачів вищої освіти
забезпечення, створюється повноцінна віртуальна модель технологічних процесів.
Крім основного завдання, ця програма автоматично генерує звіти. Є можливість
роздруковувати готові документи.

Під час створення проєкту доступне використання різних функцій. Можна
скористатися масштабом, перевірити текст на помилки, керувати залежностями та
додавати діаграми. Робочі зони програми зображені на рис. 2.2:

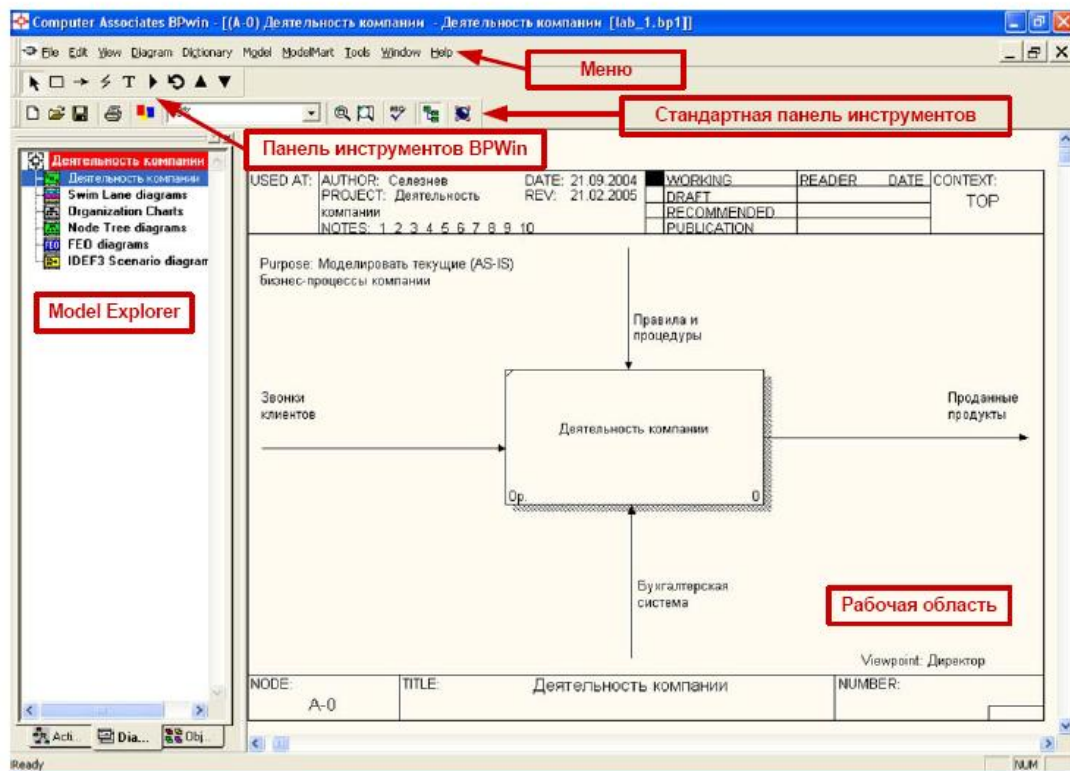


Рисунок 2.2 – Робочі зони BPWin

Використовуючи спеціальний майстер, можна працювати з діаграмами
набагато швидше. Запустивши його, потрібно вибрати тип діаграми, назву,
інформацію для відображення у комірках, а також вказати автора.

Графічна оболонка BPWin створена у примітивній оболонці без складних
панелей. Головне вікно програми містить панель інструментів та навігатора. Всі
допоміжні функції знаходяться в основному меню.

BPWin не оновлюється досить давно. Творці програмного забезпечення
зупинилися на версії 4.0, яка з'явилася у 2006 році. Встановлюючи програму, ви

зіткнетеся з проблемами сумісності, якщо на вашому комп'ютері встановлено нове збирання OS Windows. Запустіть інсталятор BPWin, використовуючи режим сумісності.

Якщо ви користуєтеся OS Windows 7, 8 та 10, то встановіть аналогічну програму – ERWin. Вона містить більшу кількість функцій, і вона сумісна з новими збірками OSWindows.

Ключові особливості:

- безліч інструментів, що забезпечують моделювання бізнес-процесів;
- графічне середовище створено простому інтерфейсі;
- у програмі вбудована функція, що генерує звіти в автоматичному режимі;
- програма оснащена майстром для створення діаграм.

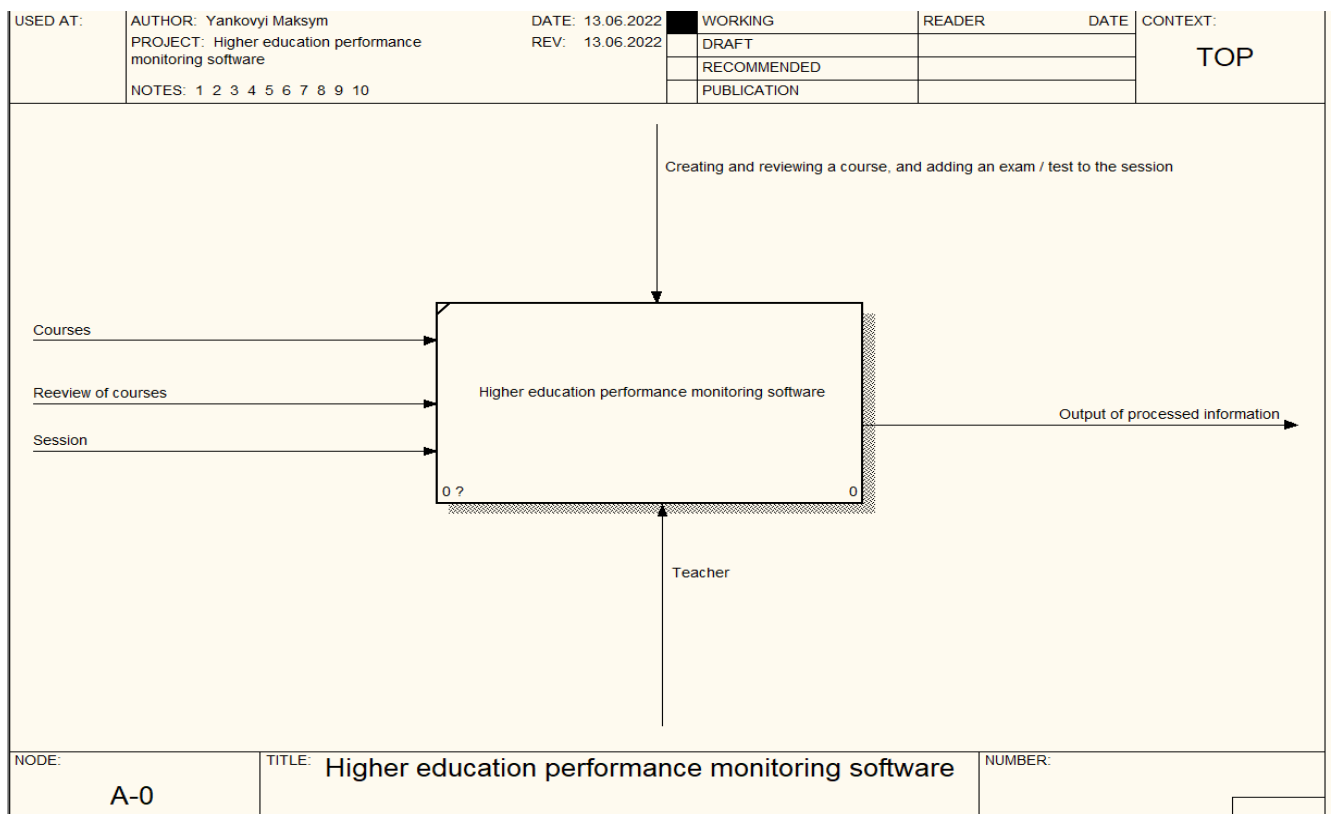


Рисунок 2.3 – Діаграма IDEF0

Суб'єктом бізнес-процесу є «Програмний застосунок для моніторингу успішності здобувачів вищої освіти» (рис. 2.3).

На вхід надходять дані:

- курси;

- огляд курсу;
- студенти;

Механізмами виконання процесу вважається:

- викладач.

Керуванням є:

- створення та перегляд курсу в огляді.
- створення та редагування нового запису.

На виході отримуємо:

- виведення обробленої інформації.

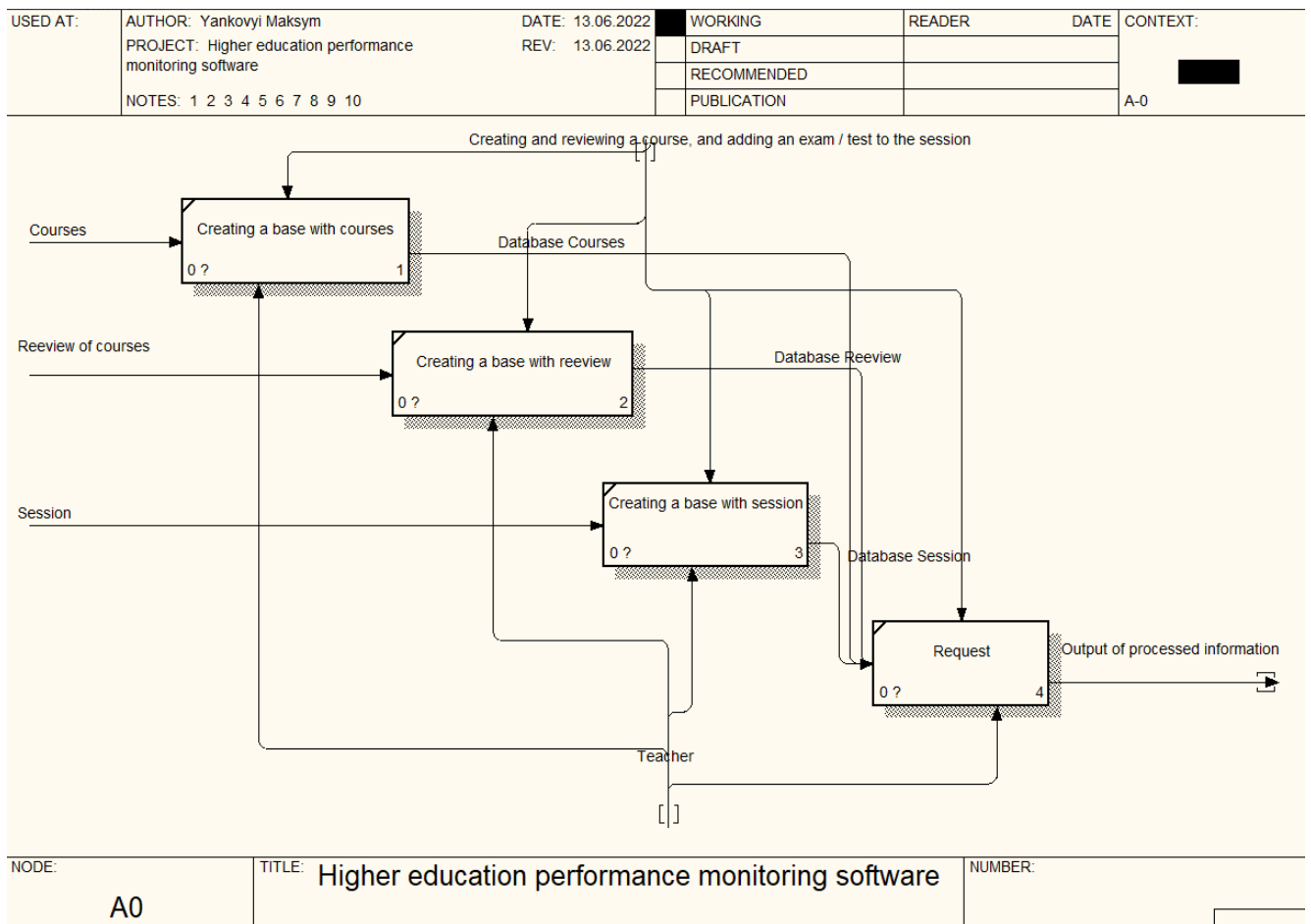


Рисунок 2.4 – Повна діаграма IDEF0

Провідними бізнес-функціями процесу «створення та редагування курсу» стануть:

1. Створення БД з студентами.

2. Створення БД з курсами.
3. Створення БД з оглядом курсу.
4. Формування запиту.

2.2 Користувацький інтерфейс

Одним із інструментів для створення вебдизайну у сфері ІТ є Figma (рис. 2.5), популярність якого зростає останніми роками. Це так званий сервіс, який дозволяє організувати спільну роботу в режимі реального часу. Даним редактором можна користуватись як онлайн через браузер, так і, встановивши програму на ПК [10].

З допомогою Figma можна розробляти:

- прототипи майбутніх продуктів;
- дизайн сайтів для ПК, моб пристроїв, планшетів;
- логотипи, іконки;
- векторні зображення.



Рисунок 2.5 – Лого Figma

Одними з особливостей та, водночас, переваг редактора є хмарне зберігання файлів, можливість спільного доступу до проєкту та роботи над ним. Тобто, принцип подібний до Google Docs. Інтерфейс зображено на рис. 2.6:

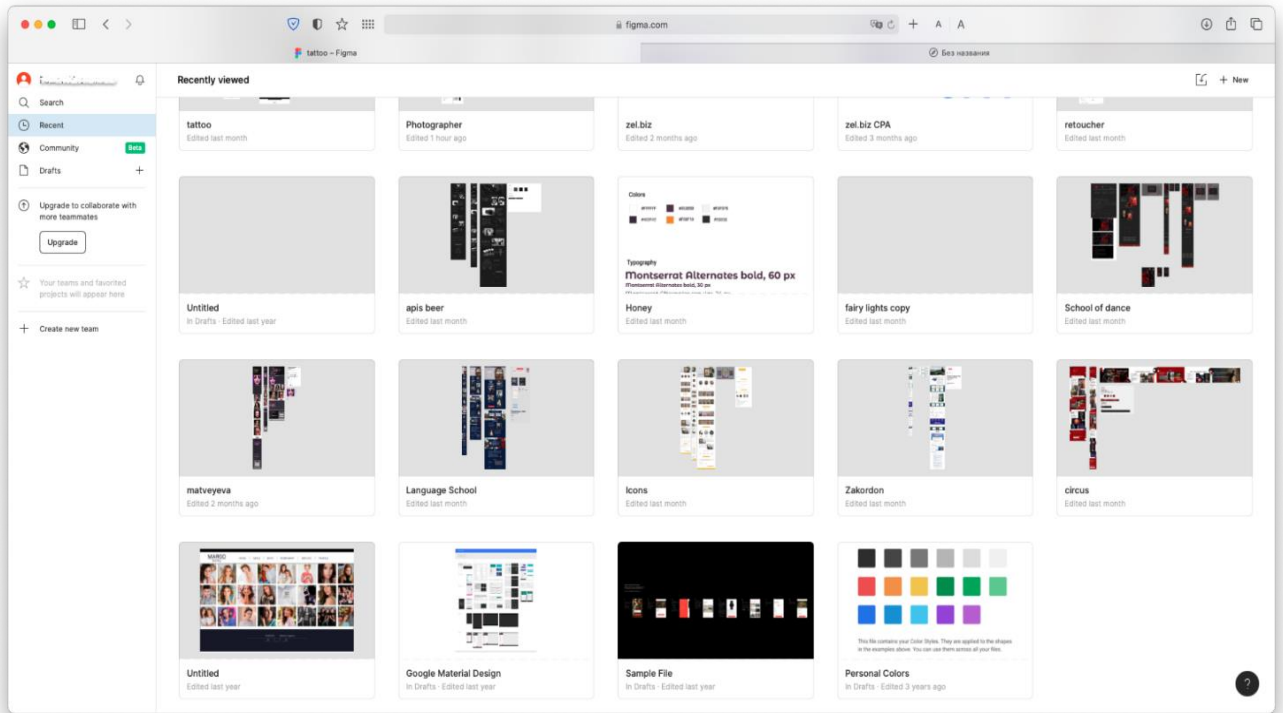
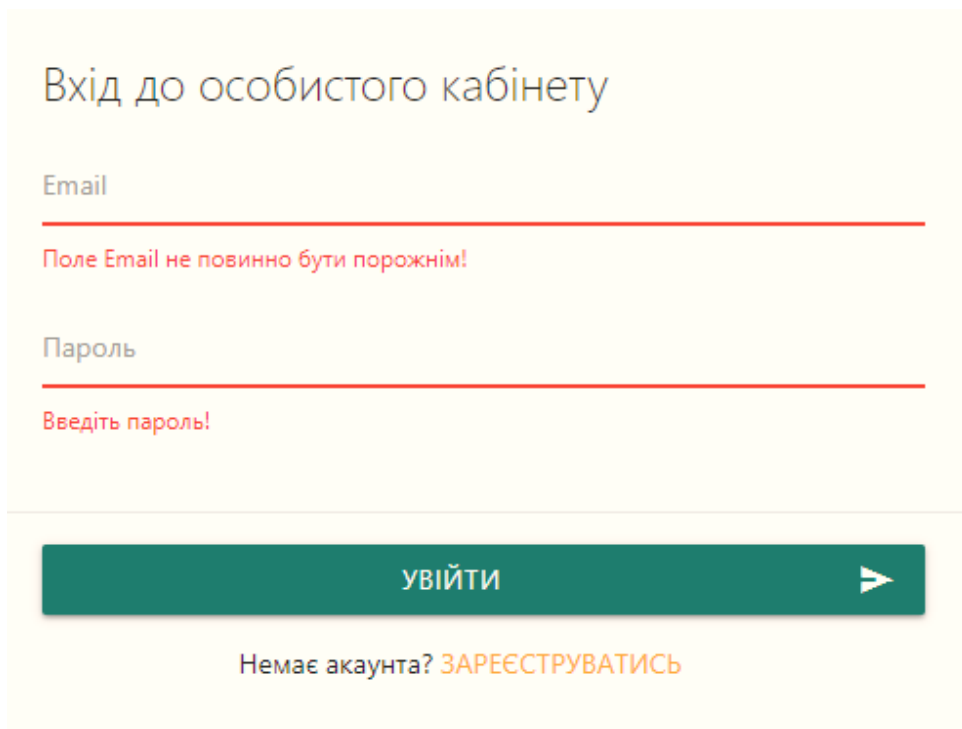


Рисунок 2.6 – Інтерфейс програми

За допомогою застосунку Figma було розроблено користувальницький інтерфейс майбутніх сторінок програмного застосунку. Форму входу та реєстрації зображено на рис. 2.7 – 2.8:



Реєстрація користувача

Email
 Поле Email не повинно бути порожнім!

Пароль
 Введіть пароль!

Ім'я
 Введіть ім'я!

Згоден з правилами

ЗАРЕЄСТРУВАТИСЬ ➤

Вже маєте аккаунт? [УВІЙТИ!](#)

Рисунок 2.8 – Форма реєстрації

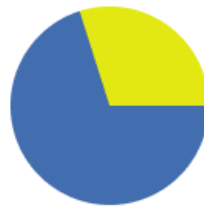
Користувач має можливість переглянути таблицю студентів на сторінці «Студенти» та фільтрувати її. Результат зображено на рис. 2.9:

| Ім'я | Прізвище | Email | Дата народження | Група(Підгрупа) | Студентський квиток |
|-----------|----------------|------------|-----------------|-----------------|---------------------|
| Стефанія | Бондаренко | @gmail.com | 01/04/2001 | 409(1) | 21810904 |
| Максим | Васев | @gmail.com | 01/04/2001 | 409(1) | 21810905 |
| Данило | Васильковський | @gmail.com | 01/04/2001 | 409(1) | 21810906 |
| Надія | Воронцова | @gmail.com | 01/04/2001 | 409(1) | 21810907 |
| Олександр | Гранченко | @gmail.com | 01/04/2001 | 409(1) | 21810909 |

Кафедра інженерії програмного забезпечення
 Програмне забезпечення моніторингу успішності здобувачів вищої освіти
 Рисунок 2.9 – Москур-сторінка «Студенти»

На сторінку «Історія записів» передається вся інформація, після додавання керівником нового запису з відповідними параметрами. Сторінка зображена на рис. 2.10:

Історія записів



| # | Дата | Кількість робіт | Курс | Тип | Список |
|---|-----------------|-----------------|----------|----------------|-------------------|
| 1 | 01.01.01, 00:00 | 5 | Курс "А" | Вчасно | → |
| 1 | 01.01.01, 00:00 | 2 | Курс "Б" | Із запізненням | → |

Рисунок 2.10 – Москур «Історія записів»

Користувач має можливість створювати та редагувати уже існуючі курси, які він проходить. Сторінка зображена на рис. 2.11:

Рисунок 2.11 – Сторінка «Курси»

Всі створенні користувачем курси відображаються на сторінці «Огляд курсу». Сторінку зображено на рис. 2.12:

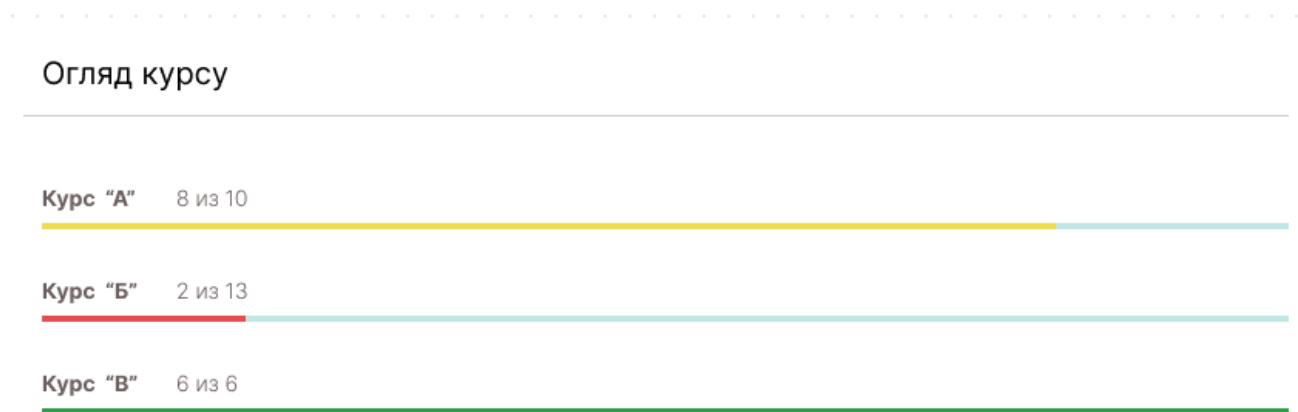


Рисунок 2.12 – Мокіур сторінка «Огляд курсу»

Кожен курс має шкалу заповнення із 3х кольорів (зеленого, жовтого, червоного) відповідно до вказаного ліміту при створенні.

Додавання нового запису із вказанням кількості робіт та прізвищем студента відбувається на сторінці «Новий запис». Сторінку зображено на рис. 2.13:

Новий запис

Оберіть курс

Вчасно

Із запізненням

Кількість робіт

Введіть студента!

СТВОРИТИ

Рисунок 2.13 – Москур сторінка «Новий запис»

June 19, 2022, 11:36:57 PM

Yankovyi Maksym

Students

History

Review

Record

Courses

Profile

Name

Yankovyi Maksym

English Українська

UPDATE >

Рисунок 2.14 – Москур вкладка «Профіль»

Вибравши вкладку «Профіль» (рис. 2.14), користувач має змогу використати динамічну локалізацію, змінивши повністю відображення застосунку на іншій мові.

Висновки до розділу 2

У другому розділі кваліфікаційної роботи бакалавра було змодельовано об'єкт та предмет дослідження. Проектування візуальних моделей відбувалось за допомогою програмного застосунку BPWin.

Проаналізовано: ключові особливості застосунку BPWin та визначено сумісність з операційною системою.

Створено: модель бізнес-процесу «Програмного застосунку моніторингу успішності здобувачів вищої освіти»; скорочену і повну діаграму IDEF0.

Розроблено: користувальницький інтерфейс за допомогою застосунку Figma. Кожна сторінка майбутнього застосунку представлена та описана у розділі.

3 АРХІТЕКТУРА, МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Використані технології

3.1.1 Vue CLI

Vue CLI (рис. 3.1) - повноцінна система для швидкої розробки на Vue.js, що надає [11]:

Інтерактивне створення проекту через @vue/cli.

Швидке прототипування через @vue/cli+ @vue/cli-service-global без конфігурації.

Runtime-залежність (@vue/cli-service) надає:

- можливість поновлення;
- створено поверх webpack, з оптимальними параметрами за замовчуванням;
- налаштування за допомогою конфігураційного файлу у проекті;
- розширюваність плагінами

- велика колекція офіційних плагінів, що інтегрують найкращі інструменти екосистеми фронтенду.
- повноцінний графічний інтерфейс для створення та управління проектами Vue.js.



Рисунок 3.1 – Лого Vue

Vue CLI прагне стати стандартним інструментом для екосистеми Vue. Він забезпечує безперебійну роботу різних інструментів складання, встановлює розумні значення за замовчуванням, тому ви зможете зосередитись на розробці програми, а не проводити дні за налаштуванням. У той же час залишається гнучкість налаштування конфігурації кожного інструменту без необхідності вилучення конфігурації в окремий файл.

Компоненти системи

Vue CLI складається з декількох складових частин - якщо ви подивитесь на вихідний код, то виявите, що це монорепозиторій, в якому кілька окремо опублікованих пакетів.

CLI (@vue/cli) — це npm-пакет, що встановлюється глобально та надає команду `vue` в терміналі. Він дозволяє швидко створити новий проект командою `vue create`, або миттєво прототипувати ваші нові ідеї через `vue serve`. Також можна керувати проектами у графічному інтерфейсі через `vue ui`. Ми розглянемо, що він може робити у наступних розділах керівництва.

Сервіс CLI

CLI (`@vue/cli-service`) — це залежність для розробки. Це npm-пакет, встановлюваний локально кожен проєкт, створений з допомогою `@vue/cli`.

Сервіс CLI побудований на основі `webpack` та `webpack-dev-server`. Він містить:

- ядро сервісу, яке завантажує інші плагіни CLI;
- внутрішню конфігурацію `webpack`, оптимізовану більшості додатків;
- бінарний файл `vue-cli-service` всередині проєкту, який дозволяє

використовувати команди `serve`, `build` та `inspect`.

Якщо ви знайомі з `create-react-app`, то `@vue/cli-service` подібний до `react-scripts`, хоча набір можливостей і відрізняється.

Плагіни для CLI

Плагіни для CLI - це npm-пакети, які надають додаткові можливості для ваших проєктів, створених через Vue CLI, такі як транспіляція Babel/TypeScript, інтеграція ESLint, модульне тестування та E2E-тестування. Легко визначати плагіни для Vue CLI, оскільки їх імена починаються з `@vue/cli-plugin-`(для вбудованих плагінів) або `vue-cli-plugin-`(для плагінів спільноти).

Коли ви запускаєте бінарний файл `vue-cli-service` усередині проєкту, він автоматично визначає та завантажує всі плагіни CLI, зазначені у файлі `package.json` проєкту.

`align="justify">` Плагіни можуть додаватися як частина проєкту на етапі його створення або їх можна додати в проєкт пізніше. Вони можуть бути також згруповані в пресети налаштувань. Ми обговоримо їх докладніше у розділі плагіни та пресети.

3.1.2 Materialize.css

Materialize - це CSS-фреймворк для створення сайтів, що базується на принципах `material design`. Якщо сказати простіше, це набір інструментів для верстки і готові компоненти, наприклад: сітка, каруселі, акордеон, адаптивне

Кафедра інженерії програмного забезпечення
Програмне забезпечення моніторингу успішності здобувачів вищої освіти

меню. Загалом є всі необхідні компоненти, виконані в красивому та сучасному дизайні для верстки сайтів [12]. Логотип зображено на рис. 3.2:



Рисунок 3.2 – Лого Materialize

Економія часу розробки нескладного багатосторінника чи лендингу, з допомогою готових компонентів і стилів, просто колосальна. Витрачений час на вивчення даного фреймворку з лишком окупиться на першому ж проекті. Потрібно знати основи HTML та CSS. Вся суть роботи з фреймворком полягає у постановці потрібних класів та атрибутів у потрібні вам елементи.

Що таке material design?

CSS-стилі підтримують принципи популярного напрямку у веб-дизайні – material design, колись розробленого компанією Google. Найближчий дизайн, із якими часто порівнюють material design, це flat design (плоский дизайн). Плоский дизайн намагається сфокусувати на легкій взаємодії з сайтом та дозволяє спростити його функціональність. А також наголошує на важливості контенту, виділяючи його на сторінці. Часто матеріальний дизайн роблять у стилі мінімалізму, з мінімальним набором елементів дизайну. Буває так, що іконки та логотип – єдині нефункціональні елементи на сторінці. Ніякого глянцю та глибоких тіней на кнопках не повинно бути.

3.1.3 Firebase

Firestore (рис. 3.3) – це комплексна платформа для розробки вебпрограм від компанії Google [13].

Завдяки Firebase можна створювати такі системи, які оновлюватимуть дані в режимі реального часу на всіх пристроях, які будуть цією системою користуватися.

В першу чергу Firebase розроблявся для мобільних застосунків, але можна використовувати його і для вебзастосунків.



Рисунок 3.3 – Лого Firebase

Один користувач користується програмою на одному пристрої, інший на іншому. Один повзаємодіяв з цією системою, щось змінив і в режимі реального часу, як він там щось змінив, інший користувач, не перезавантажуючи сторінку (в цьому основний сенс), дані у нього автоматично оновлюються. У традиційному застосунку вирішення такого завдання є проблемним.

Realtime database - це зберігання та синхронізація даних у реальному часі. Це документоорієнтована база даних, всередині якої дані нас зберігаються у форматі JSON. Дані зберігаються у хмарі. Ми можемо на будь-якому пристрої перейти в нашу базу даних і щось змінити.

3.1.4 VSCode

Visual Studio Code — безкоштовний та дуже популярний редактор коду від Microsoft [13].

Редактор допомагає в роботі веброзробникам та верстальникам усіх рівнів. З одного боку, він підходить новачкам, тому що його інтерфейс інтуїтивно

Кафедра інженерії програмного забезпечення
Програмне забезпечення моніторингу успішності здобувачів вищої освіти
простий і зрозумілий. З іншого боку, в VS Code вбудовано багато можливостей, які цікаві досвідченим розробникам.

Редактор підтримує багато мов програмування та легко налаштовується під користувача. Інтерфейс редактора зображено на рис. 3.4:

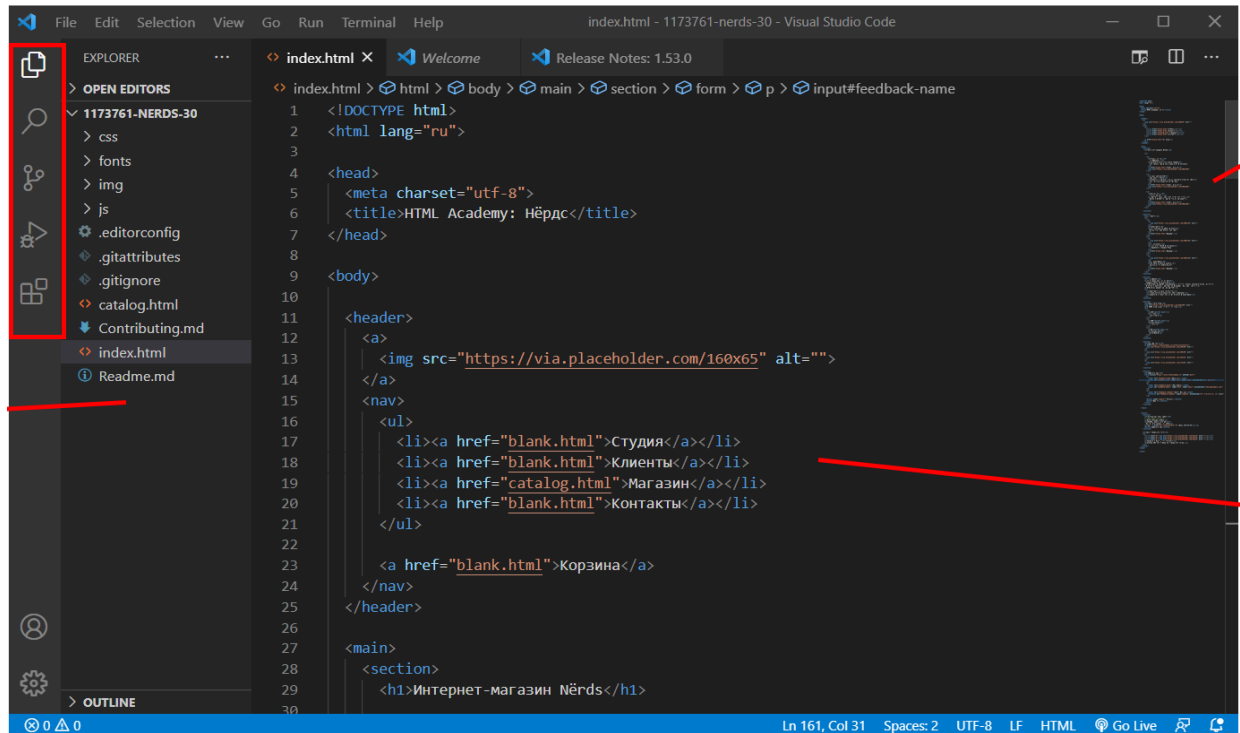


Рисунок 3.4 – Інтерфейс редактора

Наступний пункт – пошук (рис. 3.5). У полі можна ввести потрібне значення, і редактор покаже, де воно знаходиться. За бажання можна змінити значення через поле заміни.

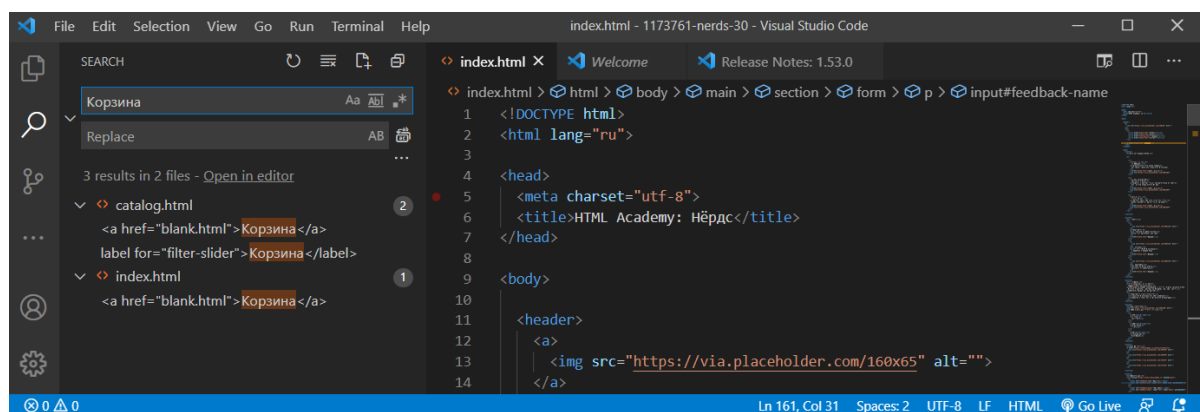


Рисунок 3.5 – Пошук у застосунку

VS Code із коробки підтримує роботу із системою контролю версій Git. Можна виконати базові операції та переглянути, як змінилися файли з моменту останнього комміту.

VS Code відрізняє від конкурентів наявність вбудованого відладчика. Після невеликого налаштування можна шукати баги в коді прямо з редактора - наприклад, поставити точку зупинки та спостерігати за виконанням конкретної ділянки коду. Крім цього редактор має вбудовану консоль, в яку може виводитися результат роботи або повідомлення про помилку, якщо щось пішло не так. Відладчик можна налаштувати під різні мови та різні завдання.

VS Code використовується розробниками в компаніях різного розміру, а також дуже зручний під час навчання на професійних курсах HTML Academy. У ньому є всі функції, за допомогою яких можна написати, протестувати та успішно захистити випускний проект.

3.2 Сценарії використання системи

StarUML – це програмний інструмент візуального моделювання з відкритим вихідним кодом, який підтримує стандартизовану мову графічного опису UML (Unified Modeling Language) для моделювання систем та програмного забезпечення [14].

Програмний продукт StarUML від розробника MKLabs призначений для створення та застосування графічних моделей у нотації UML. Система заснована на UML 2.0 та надає одинадцять різних видів діаграм, активно підтримуючи таким чином підхід побудови архітектур на базі моделей (англ. Model Driven Architecture, MDA). Система може ефективно застосовуватися системними аналітиками, проектувальниками та архітекторами систем, інженерами-програмістами. Логотип застосунку зображено на рисунку 3.6:



Рисунок 3.6 – Лого StarUML

Програмне забезпечення StarUML відрізняється високою налаштованістю відповідно до користувача середовищем і високою розширюваністю у своїй функціональності. Діаграма варіантів використання (рис. 3.7) має одну дійову особу – Користувач. Наведено всі можливості зареєстрованого користувача.

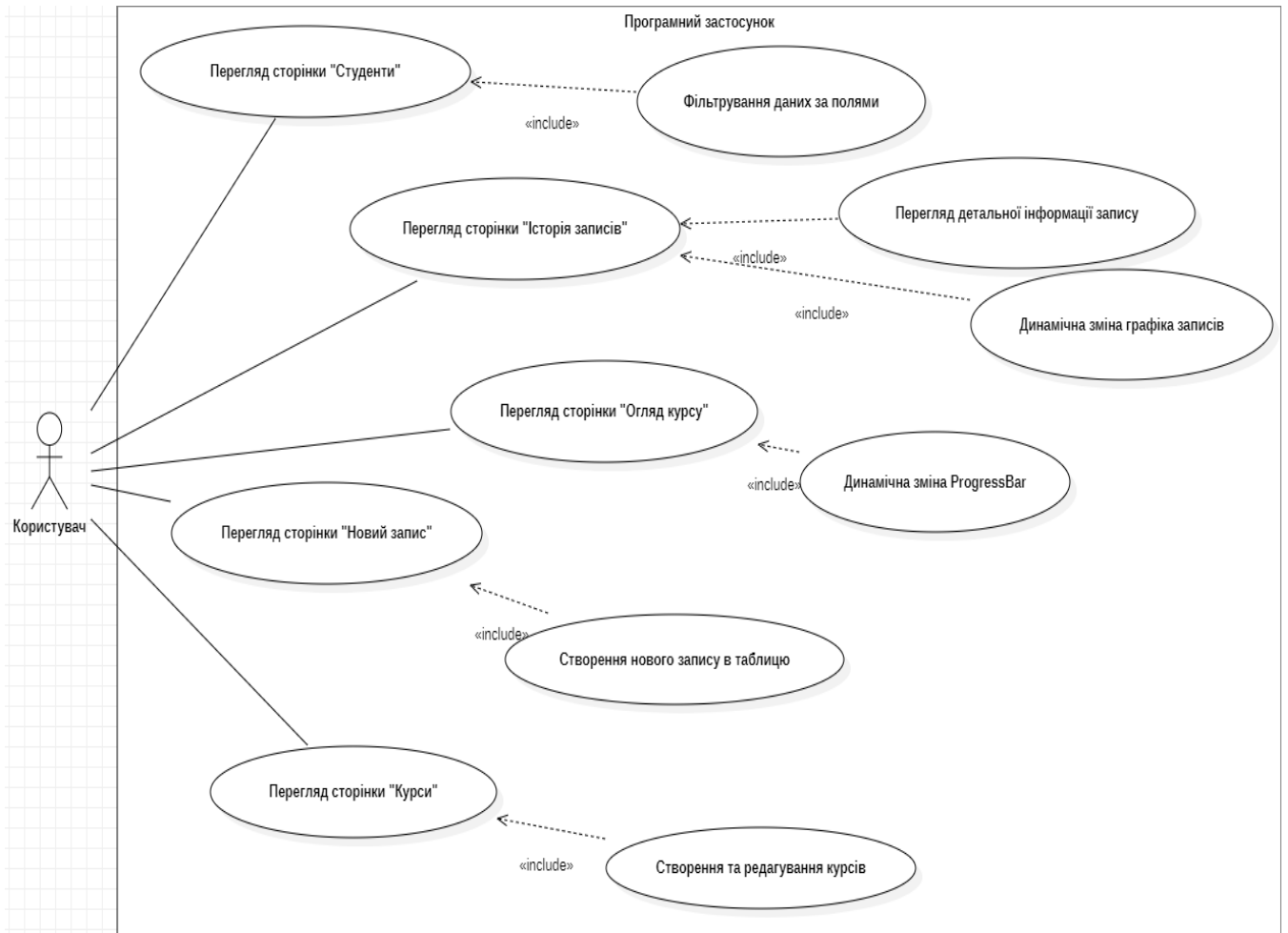


Рисунок 3.7 – Діаграма варіантів використання

Сценарій 1. Вхід до системи(рисунок 3.8).

Користувач потрапляє на сайт і йому одразу пропонують виконати вхід під своїм логіном та паролем. Після введення даних, система звіряється з базою даних на предмет збігу та дозволяє виконати вхід. Після цього користувач буде переміщений на головну сторінку сайту.



Рисунок 3.8 – Авторизація користувача

Сценарій 2. Альтернативні сценарії(рисунок 3.9).

Головний сценарій.(Успішний)

Користувач виконує вхід під своїм логіном та паролем. Після вводу, переходить на головну сторінку сайту.

Альтернативні сценарії.

1. Немає доступу. Введено неправильний логін або пароль.
2. Сервер не відповідає. Завантаженість системи, або помилка мережі.



Рисунок 3.9 – Альтернативні сценарії

Сценарій 3. Внесення даних до системи.

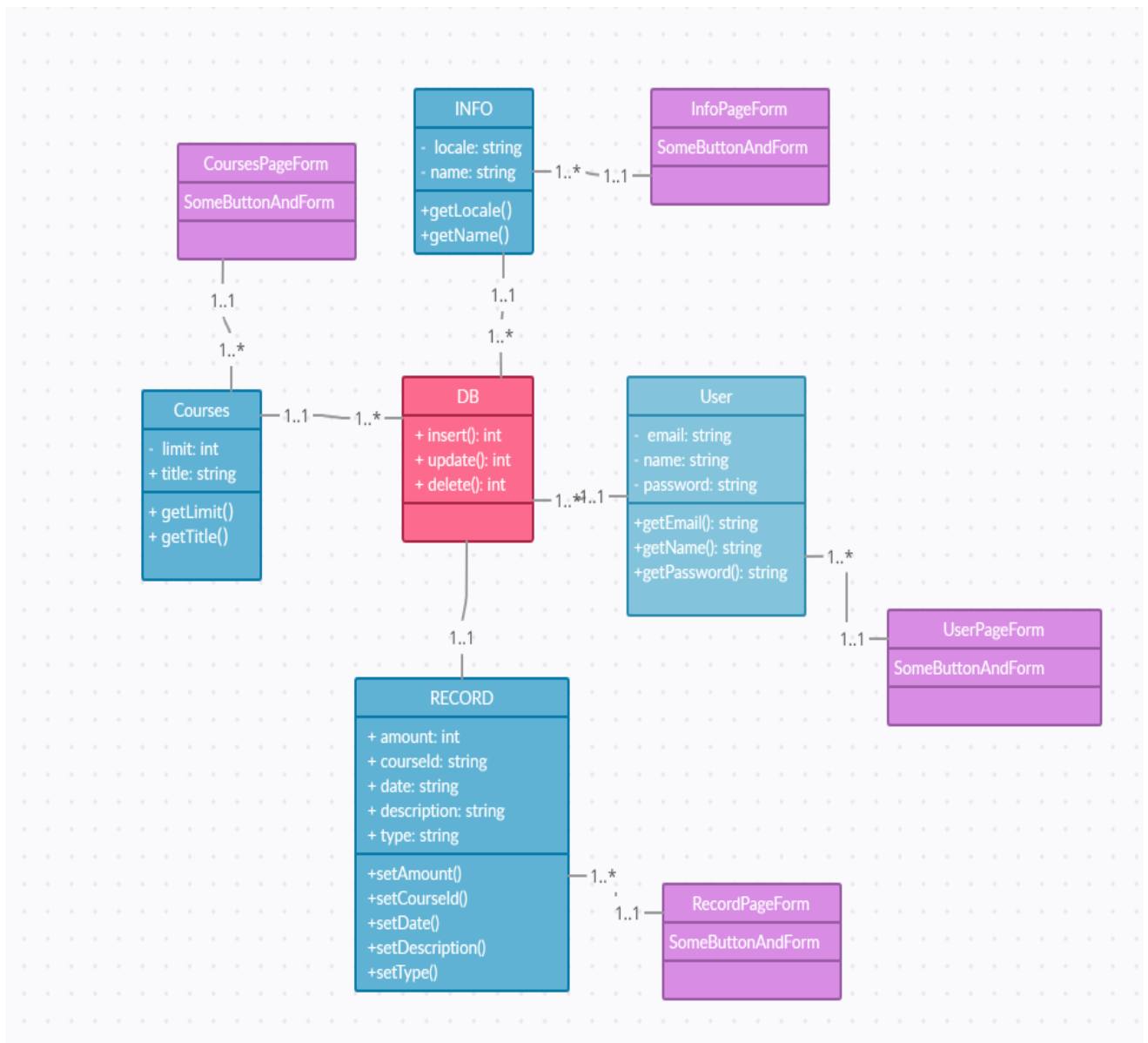
Таблиця 3.3 – Система моніторингу успішності та створення дисципліни на сторінці «Курси»

| | |
|----------------------------|---|
| Scope | Система моніторингу успішності здобувачів вищої освіти |
| Level | Додати нову дисципліну на сторінці «Курси» |
| Primary Actor | Викладач |
| Stakeholders and interests | Викладач: зацікавлений у швидкому введенні інформації та доступності її для перегляду/редагування. |
| Precondition | Викладач виконав вхід до системи |
| Main Success Scenario | <ol style="list-style-type: none"> 1. Викладач здійснює пошук необхідного курсу. 2. Потрапляє на сторінку «Огляд курсу» та переглядає його. 3. Переходить на сторінку «Курси». 4. Натискає на кнопку «Створити» або «Редагувати». 5. Додає новий курс до переліку, або редагує уже існуючий. |
| Extensions | <ol style="list-style-type: none"> 1. Викладач може відкрити сторінку «Огляд курсу» та переглянути усі його курси. 2. Помилка системи. 3. Відмовлено у доступі до сайту. |
| Special Requirements | Пристрій з доступом до інтернету та який задовольняє мінімальні системні вимоги браузера. |

| | |
|-------------------------|-------------------------------------|
| Frequency of Occurrence | Система може працювати безперервно. |
|-------------------------|-------------------------------------|

3.3 Діаграми класів, послідовності, компонентів

Діаграма класів – статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм. Діаграма класів служить для представлення статичної структури моделі системи в термінології класів об'єктноорієнтованого програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.



User – клас, який відповідає за представлення користувача;

- string email – поле, містить email користувача;
- string name – поле, містить ім'я користувача;
- string password – поле, містить пароль користувача;

Опис класу Courses:

Courses – клас, який відповідає за створення Курсів

- int limit – поле, містить кількість робіт;
- string title – поле, містить назву предмету;

Опис класу Info:

Info – клас, який відповідає за локалізацію

- string locale – поле, містить вибрану мову;
- string name – поле, містить ім'я користувача;

Опис класу Record:

Record – клас, який відповідає за створення нових записів

- int amount – поле, містить кількість зданих робіт;
- string courseId – поле, містить назву курсу;
- string date – поле, містить дату створення запису;
- string description – поле, містить прізвище студента;
- string type – поле, містить один із checkbox;

Опис інтерфейсу DB:

DB – інтерфейс, класи які його реалізують мають можливість взаємодіяти з

БД

- int insert() – метод, за допомогою якого виконується запис даних до БД;
- int update() – метод, за допомогою якого виконується зміна даних у БД;
- int delete() – метод, за допомогою якого виконується видалення даних з БД;

Діаграма послідовності - різновид діаграми в UML. Діаграма послідовності

відображає взаємодії об'єктів впорядкованих за часом. Зокрема, такі діаграми відображають задіяні об'єкти та послідовність надісланих повідомлень.

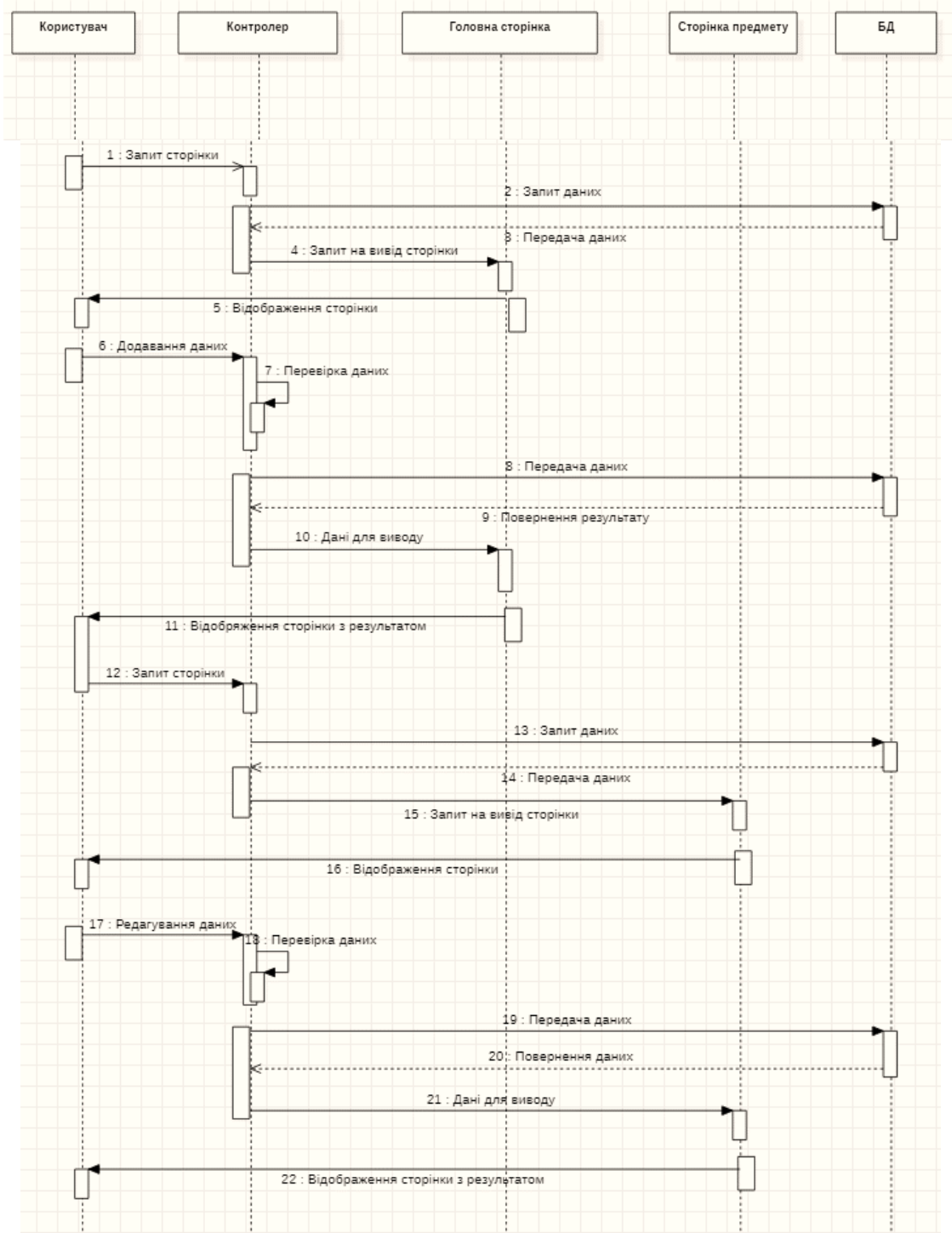


Рисунок 3.11 – Діаграма послідовності

Визначені стандартом UML 2.0 діаграми послідовностей мають ті ж можливості, що і визначені стандартом UML 1.x, і підтримують додаткові можливості зміни стандартного порядку повідомлень.

- Контролер робить запит до бази даних. База даних повертає дані.
- Контролер робить запит до сторінки після цього користувач бачить відображення сторінки.
- Користувач переглядає свої курси. Контролер перевіряє введені дані, відправляє до бази даних, отримує результат. Після цього, контролер передає дані до сторінки і уже йде відображає нових даних.

Діаграма компонентів - діаграма, на якій відображаються компоненти, залежності та зв'язки між ними. Діаграма компонентів відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись.

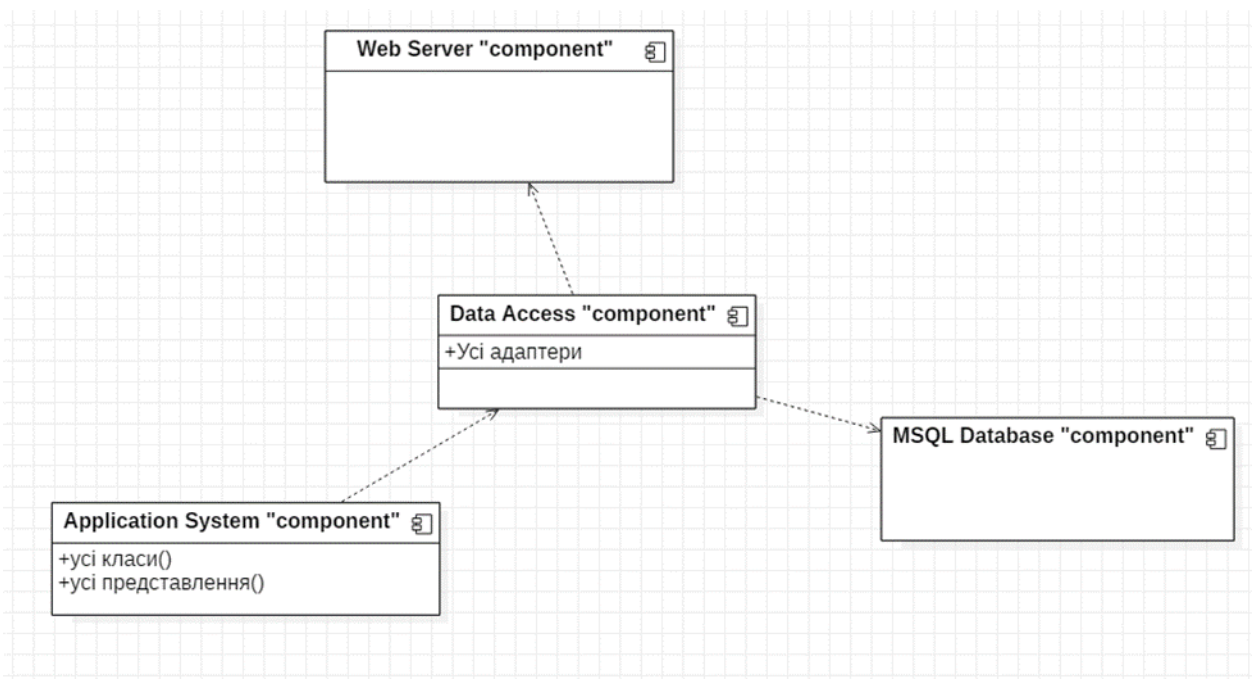


Рисунок 3.12 – Діаграма компонентів

Діаграма компонентів відображає лише структурні характеристики, для відображення окремих екземплярів компонент слід використовувати діаграму розгортання.

Компоненти об'єднуються, разом використовуючи структурні зв'язки, щоб об'єднати інтерфейси двох компонент. Це ілюструє зв'язок типу «клієнт-сервер».

Структурна взаємодія — «зв'язок двох компонент, який передбачає, що один з них надає послуги, потрібні іншому компоненту.»

При використанні діаграми компонентів, щоб показати внутрішню структуру компонента, клієнтські та серверні інтерфейси можуть утворювати пряме з'єднання з внутрішніми. Таке з'єднання називається з'єднанням делегації.

Висновки до розділу 3

У третьому розділі кваліфікаційної роботи бакалавра було здійснено моделювання та проектування програмного забезпечення.

Наведено список використаних технологій та короткий опис кожної з них.

Визначено середовище розробки майбутнього програмного застосунку.

За допомогою програмного інструменту StarUML , візуально змодельовано діаграми використання, класів, послідовності та компонентів.

Описано головний та альтернативні сценарії.

Кафедра інженерії програмного забезпечення
Програмне забезпечення моніторингу успішності здобувачів вищої освіти

4 КОДУВАННЯ, ТЕСТУВАННЯ (ВЕРИФІКАЦІЯ, ВАЛІДАЦІЯ) ПЗ.

АНАЛІЗ РЕЗУЛЬТАТІВ РОЗРОБКИ

4.1 Створення застосунку з Vue CLI 2

Для створення майбутнього проєкту необхідно встановити останню версію NODE.JS з офіційного сайту, який несе у собі пакетний менеджер NPM (рис. 4.1):



Рисунок 4.1 – Офіційний сайт Node.js

Команди в консолі для перевірки наявності встановлення та версій зображено на рис. 4.2:

```
C:\Users\maksy>node -v
v18.3.0

C:\Users\maksy>npm -v
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
8.11.0
```

Рисунок 4.2 – Консоль cmd

Кафедра інженерії програмного забезпечення
Програмне забезпечення моніторингу успішності здобувачів вищої освіти
Встановлюємо інструмент VUE CLI за допомогою команди, яка зображена

на рис. 4.3:

```
npm install -g @vue/cli
# OR
yarn global add @vue/cli
```

Рисунок 4.3 – Install @vue/cli

У менеджері проєктів Vue створюємо проєкт (рис. 4.4):

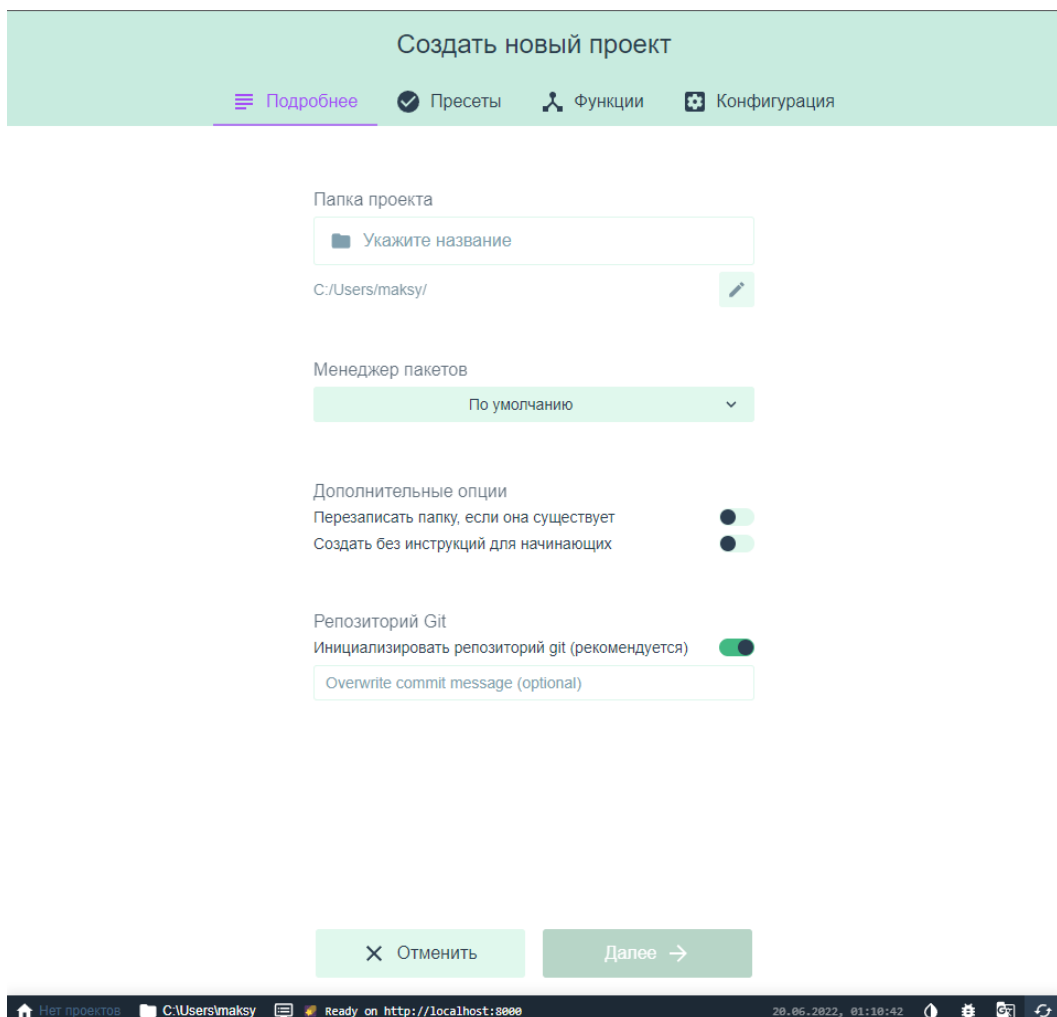


Рисунок 4.4 – Створення нового проєкту

Відкриваємо створений проєкт у редакторі коду VSCode та встановлюємо необхідні розширення (рис. 4.5):

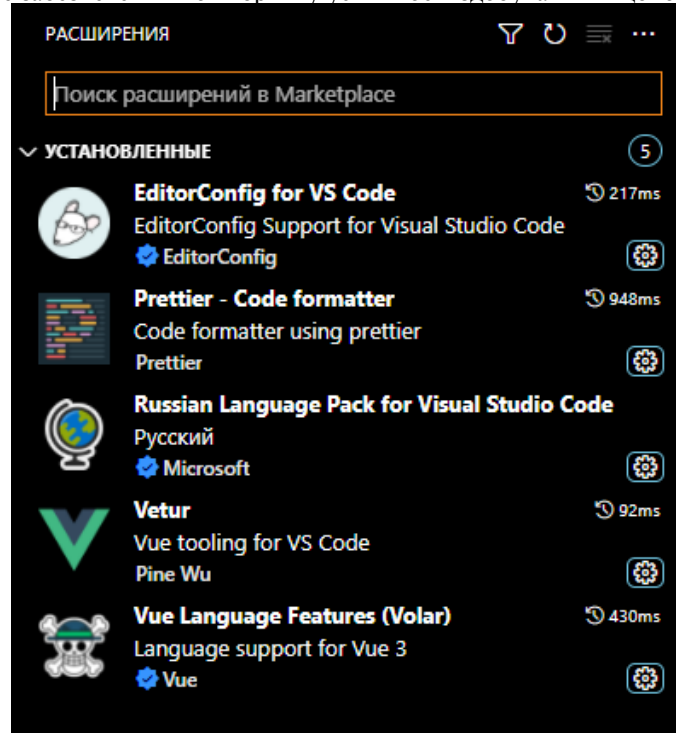


Рисунок 4.5 – Розширення для VSCode

Всі встановлені розширення значно полегшують написання коду та допомагають при роботі з бібліотекою.

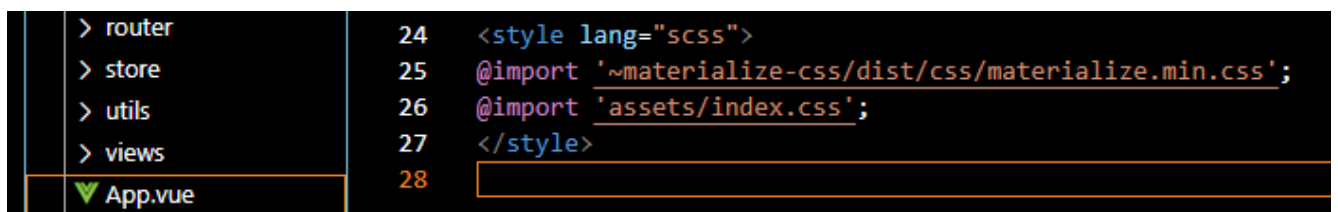
4.2 Реалізація сторінок та Layouts

Спочатку додамо пакет `materialize` до нашого проекту (рис. 4.6):

```
PS C:\Users\maksy\Desktop\vue-diplom> npm install materialize-css@next
```

Рисунок 4.6 – Підключення `materialize-css`

В головному компоненті `App.vue` імпортуємо файли CSS (рис. 4.7):

Рисунок 4.7 – Імпорт `css`-файлів

Створемо папку `layouts`, в якій розмістимо файли `EmptyLayout` та `MainLayout`.

`EmptyLayout` (рис. 4.8) будуть використовувати лише 2 файли `login.vue`(форма входу)та `register.vue`(форма реєстрації):



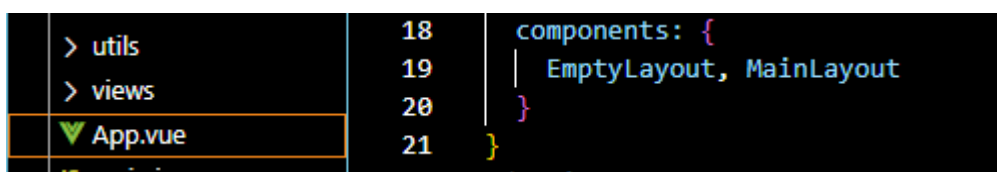
```

1 <template>
2   <div class="grey darken-1 empty-layout">
3     <router-view />
4   </div>
5 </template>
6
7 <script>
8   import messages from '@/utils/messages'
9
10  export default {
11    computed: {
12      error() {
13        return this.$store.getters.error
14      }
15    },
16    watch: {
17      error(fbError) {
18        this.$error(messages[fbError.code] || 'Щось пішло не так...')
19      }
20    }
21  }
22 </script>
23

```

Рисунок 4.8 – файл `EmptyLayout.vue`

Необхідно також створити компоненти `layout` в файлі `App.js` (рис. 4.9):



```

18 components: {
19   EmptyLayout, MainLayout
20 }
21

```

Рисунок 4.9 – `Components` в `App.js`

Для решти файлів майже з одноковою формою використаємо файл `MainLayout`. Створимо 2 файли `Sidebar` і `Navbar` з наступною реалізацією, в яких розмістимо основний макет для керування(рисунок 4.10):

```

42 <script>
43 export default {
44   data: () => ({
45     date: new Date(),
46     interval: null,
47     dropdown: null,
48   }),
49   methods: {
50     async logout() {
51       await this.$store.dispatch('logout')
52       this.$router.push('/login?message=logout')
53     }
54   },
55   computed: {
56     name() {
57       return this.$store.getters.info.name
58     }
59   },
60   mounted() {
61     this.interval = setInterval(() => {
62       this.date = new Date()
63     }, 1000)
64     this.dropdown = M.Dropdown.init(this.$refs.dropdown, {
65       constrainWidth: false
66     })
67   },
68   beforeDestroy() {
69     clearInterval(this.interval)
70     if (this.dropdown && this.dropdown.destroy) {
71       this.dropdown.destroy()
72     }
73   }
74 }
75 </script>
76
  
```

Рисунок 4.10 – Navbar.vue

```

1 <template>
2 <ul class="sidenav app-sidenav" :class="{open: value}">
3
4   <router-link
5     v-for="link in links"
6     :key="link.url"
7     tag="li"
8     active-class="active"
9     :to="link.url"
10    :exact="link.exact"
11   >
12     <a href="#" class="waves-effect waves-orange pointer">{{link.title}}
13   </router-link>
14 </ul>
15 </template>
16
17 <script>
18 import localizeFilter from '@/filters/localize.filter'
19 export default {
20   props: ['value'],
21   data: () => ({
22     links: [
23       {title: localizeFilter('Menu_Session'), url: '/', exact: true},
24       {title: localizeFilter('Menu_History'), url: '/History'},
25       {title: localizeFilter('Menu_Review'), url: '/Review'},
26       {title: localizeFilter('Menu_Record'), url: '/Record'},
27       {title: localizeFilter('Menu_Courses'), url: '/Courses'}
28     ]
29   })
30 }
31 </script>
32
  
```

Рисунок 4.11 – Navbar.vue

4.2.1 Сторінка «Студенти»

На головній сторінці реалізована таблиця, до якої занесені дані про студентів. Користувач у змозі фільтрувати дані, шляхом пошуку необхідної інформації. Першим кроком буде імпорт необхідних елементів у файл Main.js для роботи з таблицею (рис. 4.12-4.13):

```
import { TableComponent, TableColumn } from 'vue-table-component'
```

Рисунок 4.12 – Import

```
Vue.component('table-column', TableColumn);
```

Рисунок 4.13 – Vue.component

Запис даних у таблицю наведено на рис. 4.14:

```
<table-component
  :data="{
    { firstName: 'Стефанія', lastName: 'Бондаренко', group: '409(1)', birthday: '01/04/2001', number: 21810904, email: '@gmail.com'},
```

Рисунок 4.14 – Запис у таблицю

Реалізація створення колонок та їх заповнення, а також сортування наведена на рис. 4.15:

```
sort-by="number"
sort-order="asc"

<table-column show="firstName" label="Ім'я"></table-column>
<table-column show="lastName" label="Прізвище"></table-column>
<table-column show="email" label="Email" data-type="email"></table-column>
<table-column show="birthday" label="Дата народження" data-type="date:DD/MM/YYYY"></table-column>
<table-column show="group" label="Група(Підгрупа)" data-type="numeric"></table-column>
<table-column show="number" label="Студентський квиток" data-type="numeric"></table-column>
```

Рисунок 4.15 – Column and Sort

Кінцевий варіант сторінки наведено на рис. 4.16:

| Ім'я | Прізвище | Email | Дата народження | Група(Підгрупа) | Студентський квиток |
|-----------|----------------|------------|-----------------|-----------------|---------------------|
| Стефанія | Бондаренко | @gmail.com | 01/04/2001 | 409(1) | 21810904 |
| Максим | Васев | @gmail.com | 01/04/2001 | 409(1) | 21810905 |
| Данило | Васильковський | @gmail.com | 01/04/2001 | 409(1) | 21810906 |
| Надія | Воронцова | @gmail.com | 01/04/2001 | 409(1) | 21810907 |
| Олександр | Гранченко | @gmail.com | 01/04/2001 | 409(1) | 21810909 |

Рисунок 4.16 – Сторінка «Студенти»

4.2.2 Сторінка «Курси»

Знаходячись на сторінці «Курси» користувач має можливість створити курс, попередньо вказавши Назву та Кількість робіт, яку буде включати даний курс (рис. 4.17):

Курси

Створити

Назва

Введіть назву курсу!
Лабораторних робіт

0

Мінімальна кількість 1

СТВОРИТИ ➤

Рисунок 4.17 – Створення курсу

Для обох полів реалізуємо валідацію, щоб форма не створювала Курси без назви та вкажемо мінімальну кількість лабораторних робіт – 1 (рис. 4.18):


```

<form @submit.prevent="submitHandler">
  <div class="input-field">
    <input
      id="name"
      type="text"
      v-model="title"
      :class="{invalid: $v.title.$dirty && !$v.title.required}"
    >
    <label for="name">Назва</label>
    <span
      v-if="$v.title.$dirty && !$v.title.required"
      class="helper-text invalid"
    >
      Введіть назву курсу!
    </span>
  </div>

  <div class="input-field">
    <input
      id="limit"
      type="number"
      v-model.number="limit"
      :class="{invalid: $v.limit.$dirty && !$v.limit.minValue}"
    >
    <label for="limit">Лабораторних робіт</label>
    <span
      v-if="$v.limit.$dirty && !$v.limit.minValue"
      class="helper-text invalid"
    >
  </div>
</form>

export default {
  data: () => ({
    title: '',
    limit: 1
  }),
  validations: {
    title: {required},
    limit: {minValue: minValue(1)}
  },
  mounted() {
    M.updateTextFields()
  },
  methods: {
    async submitHandler() {
      if (this.$v.$invalid) {
        this.$v.$touch()
        return
      }
    }
  }
}

```

Рисунок 4.18 – Програмний код створення Курсу

Після успішного створення курсу на сторінці з'являється додаткове поле Редагування. Поле містить select, у якому зберігаються і динамічно додаються нові курси, що створюються. Після натискання кнопки Оновити, буде перезаписано курс із новими даними. Результат наведено на рис. 4.19:

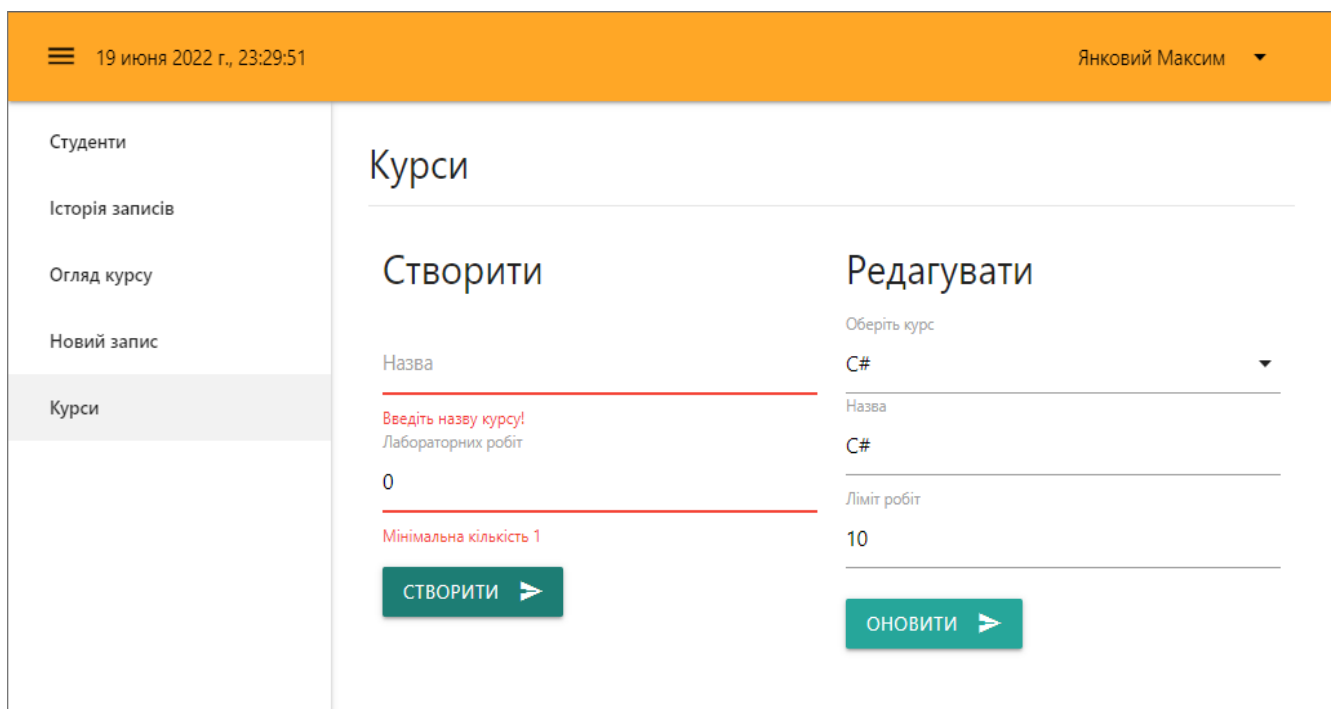


Рисунок 4.19 – Сторінка «Курси»

4.2.3 Сторінка «Новий запис»

В процесі навчання користувач має можливість створити запис, в якому він вказує кількість виконаних робіт, прізвище та в select обирає курс. В кінці обравши тип Вчасно або Із запізненням, користувач створює запис (рис. 4.20):

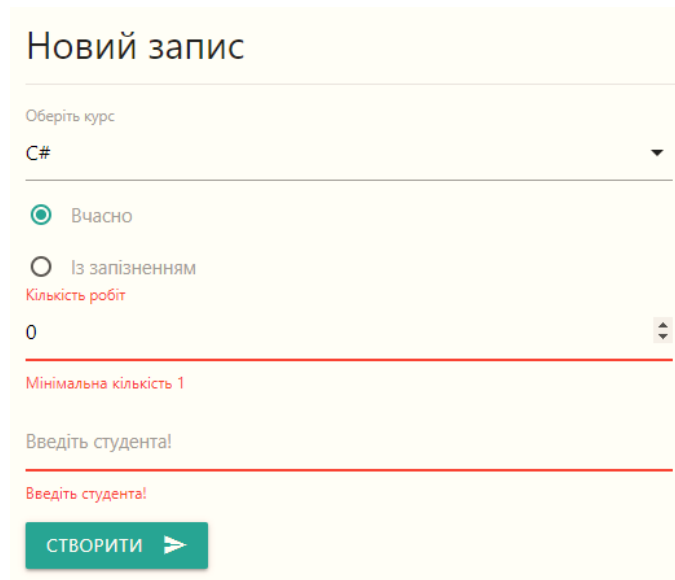


Рисунок 4.20 – Створення нового запису

Також присутня валідація для полів, що унеможливорює створення запису з 0 робіт, а також не вказавши студента. Частина коду реалізації сторінки зображені на рис. 4.21-4.22:

```
<script>
import {required, minValue} from 'vuelidate/lib/validators'
import {mapGetters} from 'vuex'

export default {
  name: 'record',
  data: () => ({
    loading: true,
    select: null,
    courses: [],
    course: null,
    type: 'status1',
    amount: 1,
    description: ''
  }),
  validations: {
    amount: {minValue: minValue(1)},
    description: {required}
  }
}
```

Рисунок 4.21 – Тег script

```

<div class="input-field">
  <input
    | id="amount"
    | type="number"
    | v-model.number="amount"
    | :class="{invalid: $v.amount.$dirty && !$v.amount.minValue}"
  >
  <label for="amount">Кількість робіт</label>
  <span
    v-if="$v.amount.$dirty && !$v.amount.minValue"
    class="helper-text invalid"
  >
  Мінімальна кількість {{ $v.amount.$params.minValue.min }}
  </span>
</div>

<div class="input-field">
  <input
    | id="description"
    | type="text"
    | v-model="description"
    | :class="{invalid: $v.description.$dirty && !$v.description.required}"
  >
  <label for="description">Введіть студента!</label>
  <span
    v-if="$v.description.$dirty && !$v.description.required"
    class="helper-text invalid"
  >
  Введіть студента!
  </span>
</div>

```

Рисунок 4.22 – Реалізація та валідація полів

Кінцевий варіант сторінки зображено на рис. 4.23:

Рисунок 4.23 – Сторінка «Новий запис»

4.2.4 Сторінка «Огляд курсу»

Всі створені курси можна переглянути на сторінці «Огляд курсу». Для моніторингу успішності здачі робіт по кожному курсу, реалізовано ProgressBar (рис. 4.24). Цей елемент має 3-х кольорове заповнення(червоний, жовтий, зелений).

```
export default {
  name: 'review',
  data: () => ({
    loading: true,
    courses: []
  }),
  async mounted() {
    const records = await this.$store.dispatch('fetchRecords')
    const courses = await this.$store.dispatch('fetchCourses')

    this.courses = courses.map(cat => {
      const spend = records
        .filter(r => r.courseId === cat.id)
        .filter(r => r.type === 'status1' || r.type === 'status2')
        .reduce((total, record) => {
          return total += +record.amount
        }, 0)

      const percent = 100 * spend / cat.limit
      const progressPercent = percent > 100 ? 100 : percent
      const progressColor = percent < 60
        ? 'red'
        : percent < 100
        ? 'yellow'
        : 'green'

      const tooltipValue = cat.limit - spend
      const tooltip = `${tooltipValue < 0 ? 'Більше на' : 'Залишилось'} ${Math.abs(tooltipValue)}`

      return {
        ...cat,
        progressPercent,
        progressColor,
        spend,
        tooltip
      }
    })
  }
}
```

Рисунок 4.24 – Реалізація ProgressBar

При наведенні на рядок, користувач отримує число робіт, яке йому залишилось здати для успішного завершення курсу, або 0 коли студент виконав всі роботи (рис. 4.25):

JS 10 из 11

CSS 8 из 8

Залишилось 1

Рисунок 4.25 – ProgressBar

Кафедра інженерії програмного забезпечення
Програмне забезпечення моніторингу успішності здобувачів вищої освіти
Кінцевий варіант сторінки зображено на рис. 4.26:

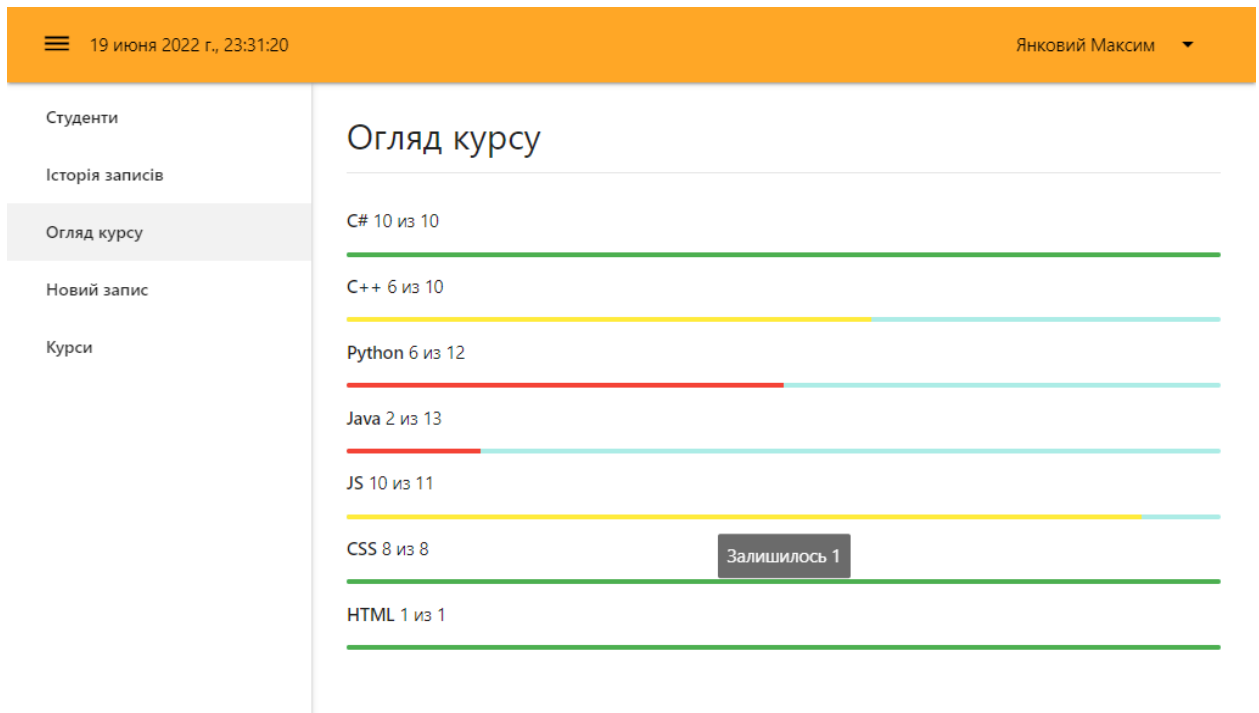


Рисунок 4.25 – Сторінка «Огляд курсу»

4.2.5 Сторінка «Історія записів»

На сторінці користувач має змогу переглянути всі створені Записи з детальною інформацією щодо них (рис. 4.26):

| # | Дата | Кількість робіт | Курс | Тип | Детальніше |
|---|----------------------------|-----------------|--------|----------------|-------------------|
| 1 | June 18, 2022, 03:06:18 PM | 8 | C# | Вчасно | ↗ |
| 2 | June 18, 2022, 03:06:39 PM | 6 | C++ | Вчасно | ↗ |
| 3 | June 18, 2022, 03:07:13 PM | 1 | HTML | Вчасно | ↗ |
| 4 | June 18, 2022, 03:11:21 PM | 1 | Python | Із запізненням | ↗ |
| 5 | June 18, 2022, 03:37:55 PM | 2 | Java | Із запізненням | ↗ |

Назад 1 2 Вперед +

Рисунок 4.26 – Записи

Кафедра інженерії програмного забезпечення
 Програмне забезпечення моніторингу успішності здобувачів вищої освіти
 Реалізацію таблиці зображено на рис. 4.27:

```

<table>
<thead>
<tr>
  <th>#</th>
  <th>Дата</th>
  <th>Кількість робіт</th>
  <th>Курс</th>
  <th>Тип</th>
  <th>Детальніше</th>
</tr>
</thead>

<tbody>
<tr v-for="(record, idx) of records" :key="record.id">
  <td>{{ idx + 1 }}</td>
  <td>{{ record.date | date('datetime') }}</td>
  <td>{{ record.amount }}</td>
  <td>{{ record.courseName }}</td>
  <td>
    <span
      :class="[record.typeClass]"
      class="white-text badge">{{record.typeText}}</span>
  </td>
  <td>
    <button
      class="btn-small btn"
      v-tooltip="'Переглянути'"
      @click="$router.push('/detail/' + record.id)"
    >
      <i class="material-icons">open_in_new</i>
    </button>
  </td>
</tr>
</tbody>

```

Рисунок 4.27 – Програмний код HistoryTable

На сторінці для комфортного перегляду даних в таблиці, реалізовано пагінацію. Спочатку встановлено плагін (рис. 4.28), та за допомогою бібліотеки Lodash створено реалізацію (рис. 4.29):

```
$ npm install vuejs-paginate --save
```

Рисунок 4.28 – Плагін paginate

```

import _ from 'lodash'

export default {
  data() {
    return {
      page: +this.$route.query.page || 1,
      pageSize: 5,
      pageCount: 0,
      allItems: [],
      items: []
    }
  },
  methods: {
    pageChangeHandler(page) {
      this.$router.push(`${this.$route.path}?page=${page}`)
      this.items = this.allItems[page - 1] || this.allItems[0]
    },
    setupPagination(allItems) {
      this.allItems = _.chunk(allItems, this.pageSize)
      this.pageCount = _.size(this.allItems)
      this.items = this.allItems[this.page - 1] || this.allItems[0]
    }
  }
}

```

```

<Paginate
  v-model="page"
  :page-count="pageCount"
  :click-handler="pageChangeHandler"
  :prev-text="'Назад'"
  :next-text="'Вперед'"
  :container-class="'pagination'"
  :page-class="'waves-effect'"
/>

```

Рисунок 4.29 – Програмний код pagination.mixins

Дані з таблиці також відображаються у вигляді графіку, якого можна динамічно змінювати на сторінці (рис. 4.30):

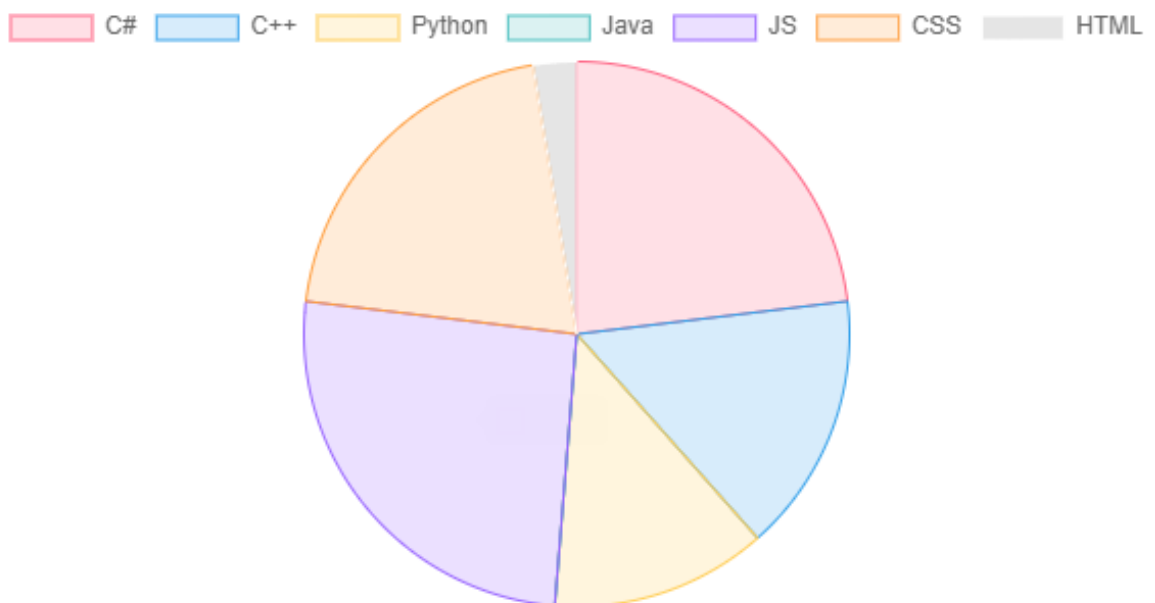


Рисунок 4.30 – графік vue-chartjs

4.3 Firebase

У терміналі проєкту встановлюємо firebase за допомогою команди (рис. 4.31):

```
PS C:\Users\maksy\Desktop\vue-diplom> npm i firebase
```

Рисунок 4.31 – Встановлення firebase

Імпортування відбувається у головному файлі Main.js. Import зображено на рис. 4.32:

```
import firebase from 'firebase/compat/app'  
import 'firebase/compat/auth'  
import 'firebase/compat/database'
```

Рисунок 4.32 – Import firebase

Далі з офіційного сайту копіюємо та додаємо до проєкту ініціалізацію у файл Main.js (рис. 4.33):

```
firebase.initializeApp({  
  apiKey: "AIzaSyB1KawiEP95-dEC5zHng1WPwf207D6FufE",  
  authDomain: "vue-diplom-aa5b4.firebaseio.com",  
  projectId: "vue-diplom-aa5b4",  
  storageBucket: "vue-diplom-aa5b4.appspot.com",  
  messagingSenderId: "1074480998703",  
  appId: "1:1074480998703:web:5f303583f5a55d613799da",  
  measurementId: "G-SZBBX5SY75"  
})
```

Рисунок 4.33 – Ініціалізація firebase

На офіційному сайті створимо нову базу даних Realtime, у якій будуть зберігатися всі дані, що обробляються та записуються на сторінках застосунку (рис. 4.34):

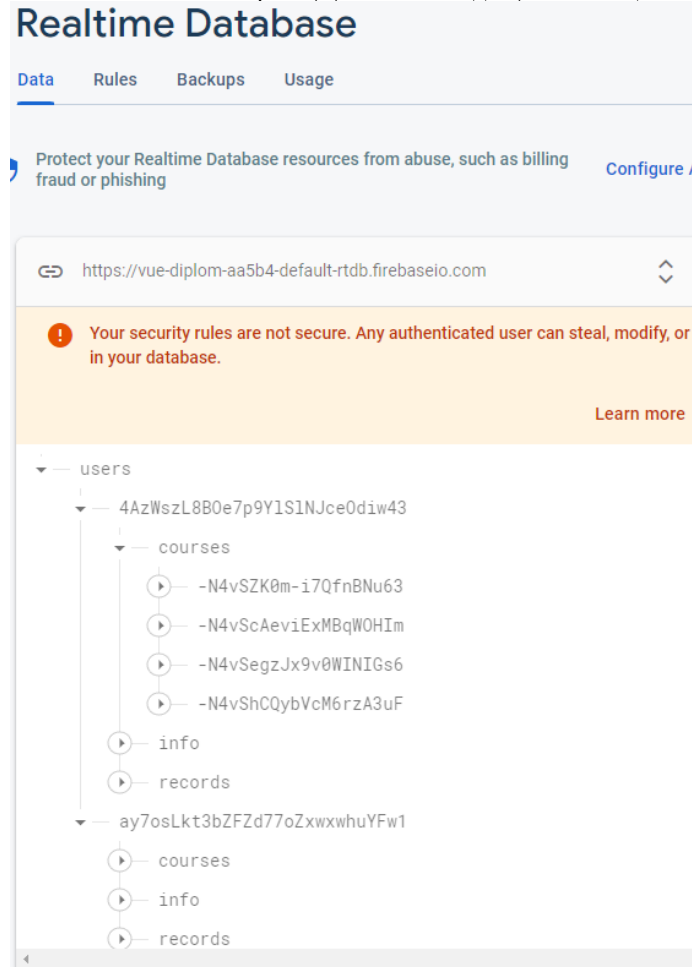


Рисунок 4.34 – БД Realtime

В окремій вкладці Authentication на сайті firebase, є можливість моніторингу зареєстрованих користувачів та відображення їх UID (рис. 4.35):

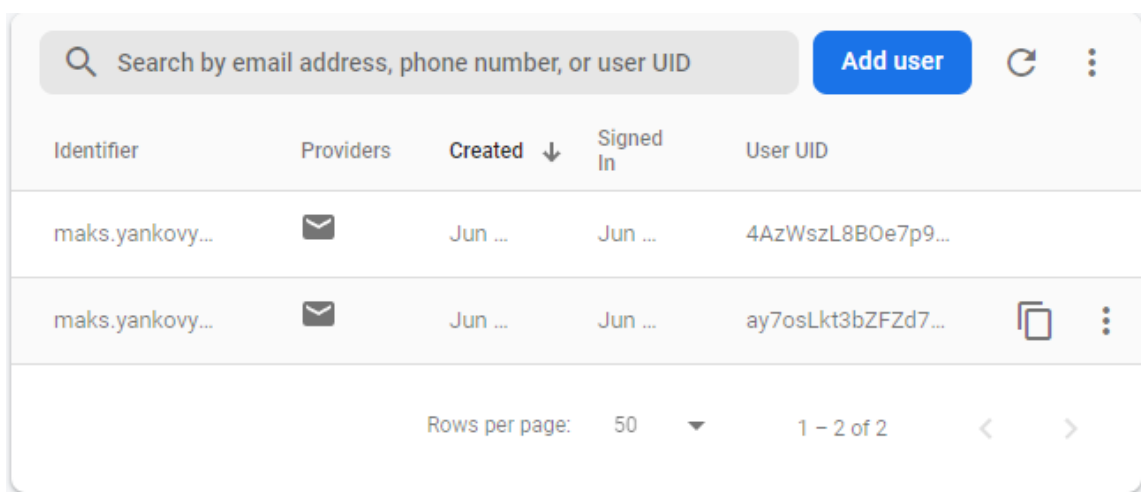


Рисунок 4.35 – Вкладка Authentication

4.4 Захист роутів

Для захисту даних від незареєстрованих користувачів було реалізовано захист роутів, які повертають користувача на певну сторінку за її посиланням. Імпортуємо firebase у файлі router.js (рис. 4.36):

```
3 import firebase from 'firebase/compat/app'  
4
```

Рисунок 4.36 – Імпорт firebase до router.js

Використаємо нове поле auth: true для кожної сторінки окрім login та register, яке буде вимагати наявності авторизації для відображення сторінки (рис. 4.37):

```
meta: {layout: 'main', auth: true},
```

Рисунок 4.37 – Поле Auth

Перевірка наявності авторизації та реалізація перенаправлення адресу зображено на рис. 4.38:

```
router.beforeEach((to, from, next) => {  
  const currentUser = firebase.auth().currentUser  
  const requireAuth = to.matched.some(record => record.meta.auth)  
  
  if (requireAuth && !currentUser) {  
    next('/login?message=login')  
  } else {  
    next()  
  }  
})  
export default router
```

Рисунок 4.38 – Метод BeforeEach

4.5 Динамічна локалізація (ua/en)

Створимо новий файл `localize.filter.js` в якому розмістимо наступну реалізацію (рис. 4.39):

```
import store from '../store'
import ua from '../locales/ua.json'
import en from '../locales/en.json'

const locales = {
  'ua-UA': ua,
  'en-US': en
}

export default function localizeFilter(key) {
  const locale = store.getters.info.locale || 'ua-UA'
  return locales[locale][key] || `[Localize error]: key ${key} not found`
}
```

Рисунок 4.39 – Файл `localize.filter.js`

Розділимо у 2 файла переклад усіх ключів, які будуть викликатися у застосунку та змінимо їх виклик на сторінках де вони знаходяться (рис. 4.40):

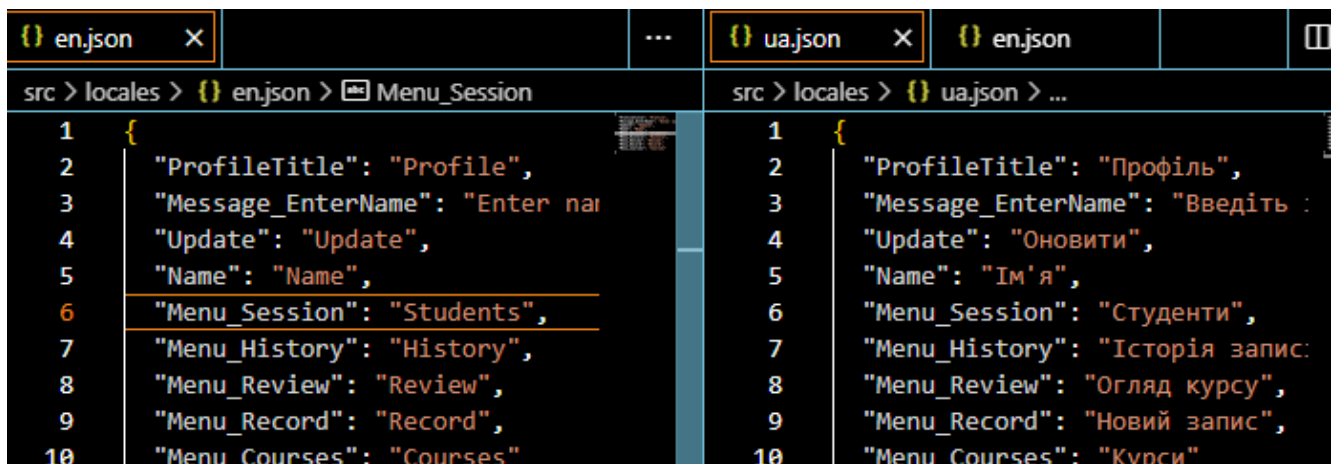


Рисунок 4.40 – Файл `en.json/ua.json`

Тепер на вкладці профіля користувач має можливість змінити власне Ім'я та мову відображення застосунку, перевівши `checkbox` в іншу сторону та натиснувши кнопку Оновити/Update (рис. 4.40):

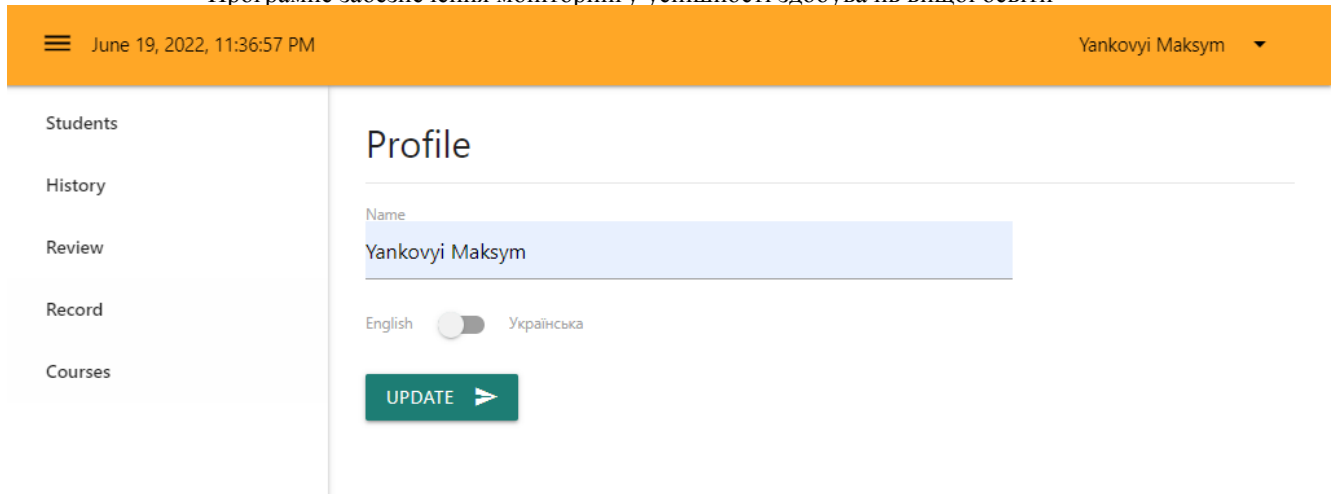


Рисунок 4.40 – Результат динамічної локалізації

Під динамічну локалізацію потрапляють всі сторінки даного застосунку, поле «Дата», Меню та Ім'я користувача.

Висновки до розділу 4

У четвертому розділі кваліфікаційної роботи бакалавра було відображено процес розробки програмного застосунку для моніторингу успішності здобувачів вищої освіти.

Наведено кодування файлів проєкту та їх реалізація і перелік необхідних розширень у середовищі розробки.

Виконано тестування програмного застосунку (верифікація, валідація) та перевірка функціоналу кожної сторінки проєкту. Створено та підключено базу даних, яка приймає всі необхідні дані при роботі застосунку.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи бакалавра було проаналізовано систему що розробляється шляхом порівняння її з аналогами, а саме: Canvas, Google Class та Moodle. Проаналізовані програмні аналоги, виявлені їх переваги та недоліки для розробки системи. З урахуванням досліджених аналогів, на їх основі побудована специфікація вимог.

Розглянуто та обґрунтовано вибір інструментів для моделювання бізнес процесів. Застосовані методології аналізу бізнес-процесів з побудовою логічних моделей.

За допомогою UML-діаграм була спроектована система, визначено користувача та його діяльність в рамках системи. Використовуючи UML-діаграми класів спроектована система, взаємодія між компонентами та створена база даних.

За допомогою програмного застосунку Figma були розроблені mockups користувацького інтерфейсу, на основі яких був побудований користувацький інтерфейс.

Спираючись на специфікацію вимог, були розглянуті та підібрані технології для клієнтської та серверної частини, виявлені їх переваги та недоліки при використанні у розробці системи. Підібрані інструменти та розширення у середовищі розробки для роботи з обраними технологіями.

Побудована та відлагоджена система для моніторингу здобувачів вищої освіти.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Stud: вебсайт. URL: <https://stud.com.ua/> (дата звернення 19.05.2022).
2. Soware/Canvas: вебсайт. URL: <https://soware.ru/products/canvas> (дата звернення 15.05.2022).
3. Soware/GoogleClass: вебсайт. URL: <https://soware.ru/products/google-classroom> (дата звернення 15.05.2022).
4. Soware/Moodle: вебсайт. URL: <https://soware.ru/products/moodle> (дата звернення 15.05.2022).
5. Kerivnyk: вебсайт. URL: <https://kerivnyk.info/> (дата звернення 16.05.2022).
6. Evergreen: вебсайт. URL: <https://evergreens.com.ua/> (дата звернення 16.05.2022).
7. Instructure: вебсайт. URL: <https://www.instructure.com/> (дата звернення 16.05.2022).
8. Moodle: вебсайт. URL: <https://moodle.org/> (дата звернення 19.05.2022).
9. Softdroids: вебсайт. URL: <https://softdroids.com/343-brwin/> (дата звернення 06.06.2022).
10. ArmadSoft: вебсайт. URL <https://armedsoft.com/ua/blog/grafichnyu-redaktor-figma> (дата звернення 05.06.2022).
11. Cli.vuejs: вебсайт. URL: <https://cli.vuejs.org/ru/guide/> (дата звернення 06.06.2022).
12. Myrusakov: вебсайт. URL: <https://myrusakov.ru/css-materialize.html/> (дата звернення 07.06.2022).
13. Avada-media: вебсайт. URL: <https://avada-media.ua/ua/services/firebase/> (дата звернення 08.06.2022).
14. HTMLAcademy: вебсайт. URL: <https://htmlacademy.ru/blog/articles/vscode> (дата звернення 09.06.2022).
15. Soware: вебсайт. URL: <https://soware.ru/products/staruml/> (дата звернення 10.06.2022).