Міністерство освіти і науки України

Вище професійне училище № 21

м. Івано-Франківська

**ДИПЛОМНИЙ ПРОЄКТ**

**Тема:** Розробка інформаційної системи

«ОБЛІК УЧНІВ»

Спеціальність *112 Комп’ютерні науки*

Освітньо-професійна програма *Обслуговування програмних систем і комплексів*

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

ДП.КН-56.15.23

|  |  |
| --- | --- |
| Консультанти  з економічних питань  викладач \_\_\_\_\_\_\_\_\_\_\_\_\_\_Ніна БОЙЧУК  (підпис) (дата)  з охорони праці  викладач \_\_\_\_\_\_\_Евеліна КОЗУЛЬКЕВИЧ  (підпис) (дата) | Учня  КН-56\_\_\_\_­­­\_\_\_\_\_\_\_\_\_Віталій МІНАЄВ  (підпис) (дата)  Керівник проєкту  \_­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Наталія БОРТНИК  (підпис) (дата)  **Допускається до захисту**  Заступник директора з навчально-виробничої роботи  \_­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Олег БЕРЛАДИН  (підпис) (дата) |

2023

**ЗМІСТ**

[**ВСТУП 4**](#_Toc138092373)

[**1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ 5**](#_Toc138092374)

[**1.1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ 7**](#_Toc138092375)

[**1.2 АНАЛІЗ ІСНУЮЧИХ АНАЛОГІВ 9**](#_Toc138092376)

[**1.3 ПОСТАНОВКА ЗАДАЧІ 13**](#_Toc138092377)

[**2 ТЕХНІЧНЕ ЗАВДАННЯ 15**](#_Toc138092378)

[**3 ОБҐРУНТУВАННЯ ЗАСОБІВ РЕАЛІЗАЦІЇ 23**](#_Toc138092379)

[**3.1 ВИБІР МОВИ ТА СЕРИДОВИЩА ПРОГРАМУВАННЯ 26**](#_Toc138092380)

[**3.2 ВИБІР СИСТЕМИ УПРАВЛІННЯ БАЗАМИ ДАНИХ 27**](#_Toc138092381)

[**4 ОПИС ЕТАПІВ РЕАЛІЗАЦІЇ 29**](#_Toc138092382)

[**4 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕСПЕЧЕННЯ 31**](#_Toc138092383)

[**5 ОПИС ПРОГРАМНОГО ПРОДУКТУ 38**](#_Toc138092384)

[**5.1 РЕЗУЛЬТАТ РОБОТИ ПРОГРАМИ 40**](#_Toc138092385)

[**5.2 ІНСТРУКЦІЯ КОРИСТУВАЧА 42**](#_Toc138092386)

[**5.3 ІНСТРУКЦІЯ АДМІНІСТРАТОРА 43**](#_Toc138092387)

[**5.4 РОБОТА ПРОГРАМНОГО ПРОДУКТУ В МЕРЕЖІ 45**](#_Toc138092388)

[**6 ЕКОНОМІЧНА ЧАСТИНА 46**](#_Toc138092389)

[**7 ОХОРОНА ПРАЦІ 54**](#_Toc138092391)

[**ВИСНОВКИ 62**](#_Toc138092392)

[**ПЕРЕЛІК ПОСИЛАНЬ НА ДЖЕРЕЛА 63**](#_Toc138092393)

**ДОДАТКИ**

# **ВСТУП**

Програмні продукти є невід'ємною частиною сучасної освіти. Серед них можна виділити програми для обліку учнів, які дозволяють вчителям і адміністраторам шкіл та інших навчальних закладів ефективно вести облік учнівського складу, визначати успішність та прогрес кожного учня.

Існує велика кількість видів програмних продуктів, включаючи веб-додатки, настільні програми, мобільні додатки, ігри, системні програми та інші. Кожен тип продукту має свої унікальні вимоги та виклики при розробці.

Для створення програмних продуктів розробники використовують середовища розробки (IDE), які надають набір інструментів та функціональностей для покращення продуктивності та ефективності розробки. Ці інструменти можуть включати текстовий редактор, дебагер, компілятор, пакетний менеджер, систему контролю версій та інші.

Розробка програм для обліку учнів вимагає знання різних типів програм та інструментів розробки, таких як бази даних, мови програмування та фреймворки. У залежності від вимог та функціональності програми можуть бути розроблені як веб-додатки, настільні програми або мобільні додатки.

Середовища розробки для програм обліку учнів також мають свої особливості. Вони надають розробникам інструменти для створення і налагодження баз даних, редагування коду та тестування програм. Деякі популярні середовища розробки для цієї галузі включають Eclipse, NetBeans, Visual Studio та інші.

Актуальність теми курсового проєкту обліку учнів полягає в тому, що програми для обліку учнів стають все більш популярними в навчальних закладах. Вони допомагають зберігати інформацію про учнів, забезпечують швидкий та ефективний доступ до цих даних та дозволяють адміністраторам шкіл і вчителям легко аналізувати успішність учнів. Для студентів, які вивчають програмування, розробка програм для обліку учнів є важливою складовою процесу вивчення програмування та може допомогти їм отримати цінні знання та досвід в розробці програмних продуктів.

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

Дослідження предметної області для програми обліку учнів включає глибоке дослідження різних аспектів, пов'язаних з обліком учнів.

Аналіз потреб користувачів: Перший крок полягає у вивченні потреб користувачів, таких як учителі, адміністрація школи та інші стейкхолдери. Проведіть опитування, інтерв'ю та зберіть фідбек, щоб зрозуміти, які основні функції та можливості повинна мати програма обліку учнів. Врахуйте потреби у реєстрації учнів, веденні присутності, оцінюванні, звітах та іншому.

Огляд існуючих систем: Дослідіть існуючі системи обліку учнів, які використовуються в інших школах або установах освіти. Оцініть їхні переваги та недоліки, функціональність та можливості. Вивчіть, які технології та інструменти використовуються у цих системах.

Вивчення законодавства та нормативних вимог: Ознайомтеся з законодавством та нормативними вимогами, що стосуються обробки та зберігання особистих даних учнів. Розберіться з принципами конфіденційності, безпеки та етичними аспектами, що відносяться до обробки особистих даних учнів.

Визначення функціональних вимог: На основі отриманих від користувачів та аналізу існуючих систем визначте функціональні вимоги для програми обліку учнів. Складіть список функцій, які повинні бути присутніми в програмі, такі як реєстрація учнів, ведення присутності, оцінювання, статистика відвідувань, звіти про успішність, збереження особистих даних учнів та інші важливі функції.

Вивчення технічних аспектів: Розгляньте технічні аспекти розробки програми обліку учнів. Вивчіть можливі технології, мови програмування, бази даних та інфраструктуру, які можуть бути використані для реалізації програми. Розгляньте питання масштабованості системи, її ефективності та безпеки даних.

Проектування і прототипування: На основі зібраної інформації розробіть детальний план програми обліку учнів. Створіть макети, прототипи або діаграми, щоб проілюструвати функціональність та взаємодію з користувачем. Врахуйте зручність інтерфейсу, логіку роботи та потенційні проблеми, які можуть виникнути.

Тестування і оцінка: Проведіть тестування програми обліку учнів для перевірки її функціональності, надійності та безпеки. Заплануйте серію тестів, щоб перевірити правильність роботи програми у різних сценаріях. Оцініть результати тестів та здійсніть відповідні виправлення та вдосконалення.

Впровадження та навчання: Розробіть план впровадження програми обліку учнів. Передбачте процес перенесення даних зі старої системи, налаштування програми та навчання користувачів. Забезпечте документацію та підтримку для вчителів та адміністраторів школи.

Оцінка ефективності: Після впровадження оцініть ефективність програми обліку учнів. Зберіть фідбек від користувачів щодо використання програми, їхньої задоволеності та пропозицій щодо вдосконалення. Аналізуйте дані про використання програми, результати оцінювання та статистику відвідувань, щоб оцінити, наскільки ефективно програма виконує свої функції.

Цей процес дослідження предметної області для програми обліку учнів допоможе зрозуміти потреби користувачів, визначити функціональні вимоги, врахувати технічні аспекти розробки, протестувати та оцінити програму перед її впровадженням. Забезпечивши комплексний підхід до дослідження, ви зможете створити ефективну та корисну програму обліку учнів, яка відповідає потребам освітнього закладу.

1.1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

Програма обліку учнів призначена для автоматизації процесів збору, зберігання та обробки інформації про учнів у навчальному закладі, такому як школа або установа освіти. Вона забезпечує ефективну та зручну систему керування даними учнів, спрощує роботу вчителів, адміністраторів та батьків, та надає доступ до актуальної інформації про прогрес та успішність учнів.

Програма обліку учнів може мати такі основні функції:

Реєстрація учнів: Можливість додавання нових учнів до системи, включаючи введення особистих даних, контактної інформації та інших важливих даних.

Ведення присутності:

запис та моніторинг присутності учнів на заняттях, включаючи відсутність з певних причин та затримки.

Оцінювання:

введення та збереження оцінок за різними предметами, тестами, контрольними роботами тощо. Розрахунок середніх балів та статистики успішності учнів.

Електронний журнал:

ведення цифрового журналу, в якому зберігаються записи про присутність, оцінки, відсутності та інші важливі відомості про учнів.

Статистика та звіти:

генерація звітів та статистики щодо присутності, успішності, відсутностей, середніх балів тощо. Це може включати індивідуальні звіти для кожного учня, загальні звіти для класів, шкільні звіти та інші аналітичні дані.

Система сповіщень для батьків:

надання можливості надсилання сповіщень, повідомлень та інформації батькам щодо прогресу та успішності їх дітей. Це може включати повідомлення про відсутність учня, отримання нової оцінки, повідомлення про батьківські збори, події школи та інші повідомлення, які важливі для батьків.

Управління розкладом занять:

можливість створення та управління розкладом занять для класів та учнів. Включає можливість визначення годин проведення занять, розподіл предметів, аудиторій та вчителів, а також можливість змінити розклад у випадку потреби.

Інтеграція з системою електронного документообігу:

можливість інтеграції програми обліку учнів з існуючою системою електронного документообігу, що використовується в навчальному закладі. Це забезпечує зручний обмін даними та документами між системами та спрощує робочий процес для адміністраторів та вчителів.

Забезпечення безпеки даних:

забезпечення захисту конфіденційності та цілісності даних учнів. Це включає заходи для запобігання несанкціонованому доступу до інформації, використання шифрування для збереження та передачі даних, резервне копіювання та відновлення даних, а також виконання вимог законодавства щодо захисту особистих даних.

Ці завдання і функціональні можливості програми обліку учнів допомагають підвищити ефективність роботи навчального закладу, забезпечити точну та актуальну інформацію про учнів, спростити ком унікацію та спілкування між навчальним закладом, вчителями, адміністрацією та батьками учнів. Вона дозволяє покращити зв'язок та залученість батьків до освітнього процесу, забезпечує швидкий обмін інформацією та сприяє зростанню ефективності співпраці між всіма зацікавленими сторонами.

Узагальнюючи, програма обліку учнів є потужним інструментом, який допомагає автоматизувати та спростити багато рутинних завдань, пов'язаних з обліком та управлінням учнями. Вона забезпечує доступ до цінної інформації про прогрес, успішність та присутність учнів, полегшує комунікацію з батьками та сприяє покращенню роботи навчальних закладів.

1.2 АНАЛІЗ ІСНУЮЧИХ АНАЛОГІВ

При аналізі існуючих аналогів програми обліку учнів на сайті можна враховувати такі аспекти:

Функціональність: розгляньте, які функції та можливості надають існуючі аналоги. Це може включати реєстрацію учнів, збереження особистих даних, відомості про відвідування, академічні досягнення, статистику успішності, графіки занять тощо. Порівняйте ці функції з вимогами вашого навчального закладу та специфічними потребами.

Інтерфейс користувача: оцініть зручність та інтуїтивність інтерфейсу існуючих аналогів. Важливо, щоб користувачі з легкістю могли взаємодіяти з системою, швидко знаходити потрібну інформацію та виконувати необхідні дії.

Інтеграція: дізнайтесь, які можливості є для інтеграції існуючих аналогів з іншими системами, які використовуються у вашому навчальному закладі, наприклад, системою електронного навчання чи системою розкладу занять. Переконайтеся, що існуючі аналоги можуть взаємодіяти з іншими системами для обміну даними та уникнення дублювання інформації.

Безпека даних: врахуйте, які механізми захисту даних використовують існуючі аналоги. Переконайтеся, що існуючі аналоги забезпечують конфіденційність та цілісність особистих даних учнів, використовують шифрування, контроль доступу та аудит дій користувачів.

Технічна підтримка: оцініть, яку технічну підтримку надають розробники існуючих аналогів. Важливо мати можливість звертатися за допомогою, отримувати вчасні оновлення програмного продукту, виправлення помилок та консультації щодо використання системи.

Вартість: розгляньте вартість впровадження та використання існуючих аналогів. Порівняйте їхню вартість з бюджетом вашого навчального закладу та зважте на відношення ціна-якість.

Після проведення аналізу існуючих аналогів ви зможете визначити, який програмний продукт обліку учнів найкраще відповідає потребам вашого навчального закладу і обрати його для впровадження.

Мета проєкту: метою проєкту є розробка програмного продукту для обліку учнів в школі або іншому навчальному закладі. Програмний продукт має забезпечувати зручну та ефективну систему ведення обліку учнів, їх успішності, відвідуваності, звітності та комунікації між вчителями та батьками.

Проблематика реалізації програмного продукту: Реалізація програмного продукту пов'язана з рядом проблем, таких як:

* Недостатня кількість кваліфікованих розробників програмного забезпечення, здатних розробити програму з потрібним функціоналом.
* Потреба в розробці великої кількості функцій та можливостей, що може призвести до збільшення вартості та тривалості розробки.
* Потреба в забезпеченні високої безпеки та захисту персональних даних учнів.

Програмні засоби, з допомогою яких планується реалізація програмного продукту: для реалізації програмного продукту можуть використовуватися такі програмні засоби:

* Мова програмування Java, що забезпечить швидку та ефективну роботу програми.
* Система керування базами даних MySQL, що забезпечить зручне та ефективне збереження даних про учнів.
* Конструктор мобільних додатків Android Studio, що забезпечить зручність та простоту використання програмного продукту.
* Текстовий редактор Sublime Text для верстки інтерфейсу програмного продукту.

Опис класів та їх взаємозв'язок: клас "Головна" містить швидку інформацію про загальну статистику усіх груп та учнів, включаючи їх особисті дані, успішність, відвідуваність та інші важливі дані. Клас "Відвідуваність" містить інформацію про кожен клас, в якому навчаються учні, включаючи список учнів, що входять до класу та шкалу їх відвідуваність.

Клас "Розклад" містить інформацію про розклад занять усіх груп та кожного вчителя, що викладає у класі, включаючи його особисті дані та список предметів, які він викладає.

Клас "Групи" містить інформацію про кожену групу, кількість учнів та інформацію про них.

Клас "Студенти" містить інформацію про кожного стундента, яку оцінку отримав учень за певний предмет, включаючи дату отримання оцінки та інші деталі.

Клас "Настройки" містить налаштування програми, він взаємодіє зі всіма сторінками в програмі та міняє тему програми та її мову, також на сторінці находиться інформація про версію та розробника.

Клас "Меню" містить меню переходів між сторінками та зв’язок з ними.

Ці класи мають взаємозв'язок між собою, де кожен клас містить посилання на інші класи та їх елементи для забезпечення повної та коректної роботи програмного продукту. Наприклад, клас "Групи" має посилання на список учнів, що входять до класу, а клас "Студенти" має посилання на список предмет.

Також для програми було розроблено логотип за допомогою програми Adobe Photoshop і Adobe Illustrator.



Рисунок 1.1 Логотип

Сам інтерфейс програмного забеспечення зроблено за допомогою технологій veb view для якого спеціально було написано сайт який служитиме інтерфейсом для програми, приклад ескізу головної сторінки.

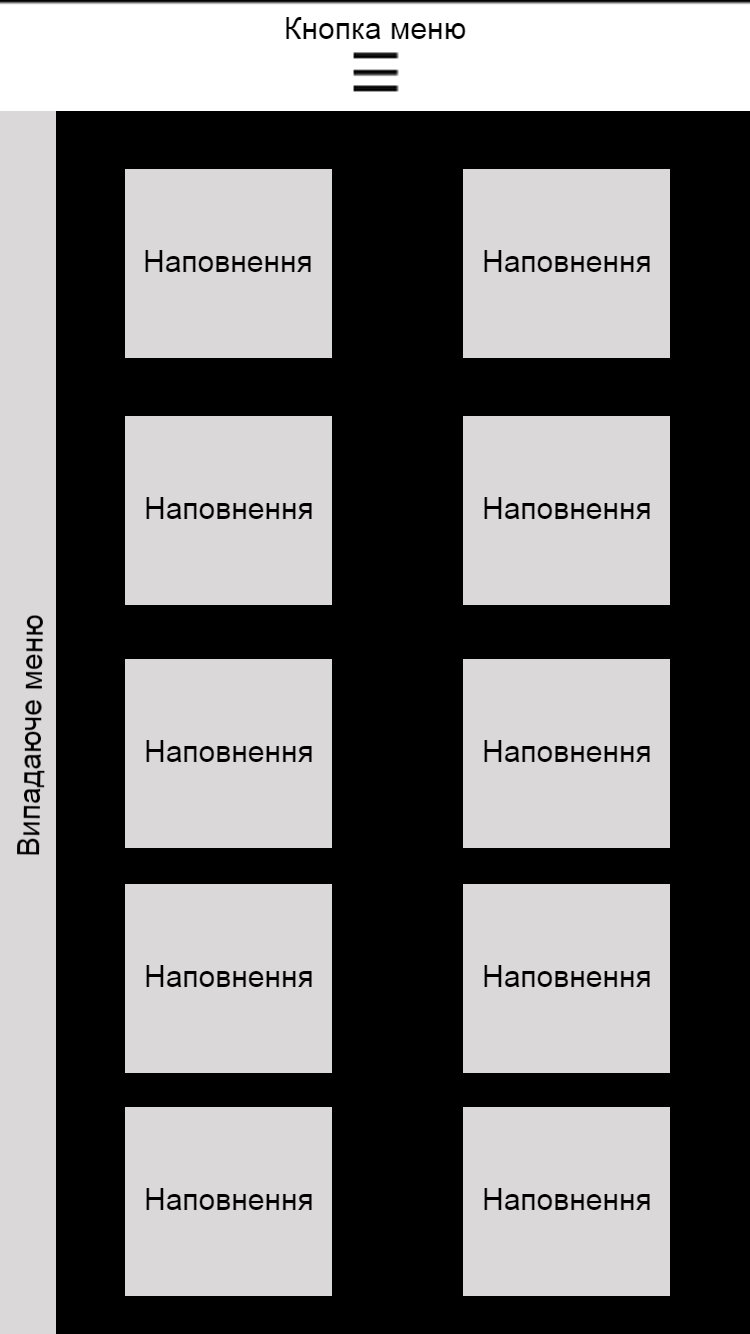


Рисунок 1.2 Макет головної сторінки

Для створення логотипу (Рис.1.1) використовувалася програма Adobe Photoshop CS6. В основу логотипа ляг мінімалістичний дизайн та фігури за допомогою яких і був створений логотип. Макет головної сторінки був також придуманий і зроблений за допомогою Adobe Photoshop CS6. (Рис.1.2)

1.3 ПОСТАНОВКА ЗАДАЧІ

Постановка задачі для створення програми обліку учнів полягає у чіткому визначенні цілей та завдань, які програма повинна виконувати. Основні складові постановки задачі включають:

* Опис функціональності: визначте основні функції, які має виконувати програма обліку учнів. Це можуть бути реєстрація учнів, ведення присутності, оцінювання, звіти про успішність, генерація статистики відвідувань, збереження та обробка особистих даних учнів тощо.
* Вимоги до інтерфейсу: визначте, яким чином користувачі будуть взаємодіяти з програмою. Розгляньте розміщення елементів інтерфейсу, зручність використання, навігацію та візуальний дизайн. Зробіть акцент на забезпечення зручного та ефективного робочого потоку для користувачів.
* Вимоги до зберігання даних: визначте, які дані про учнів потрібно зберігати, які вони є обов'язковими та які додаткові. Розгляньте питання безпеки даних, включаючи захист особистої інформації, резервне копіювання та відновлення даних.
* Вимоги до адміністрування: визначте, як будуть керувати програмою адміністратори. Розгляньте можливості налаштування системи, керування правами доступу, аудит дій користувачів та інші аспекти, пов'язані з адмініструванням програми.
* Вимоги до інтеграції: розгляньте можливості інтеграції програми обліку учнів з іншими системами, які використовуються в школі або установі освіти. Наприклад, це можуть бути системуправління розкладом занять, електронний журнал, система сповіщень для батьків, інтеграція з системою електронного документообігу тощо.
* Вимоги до забезпечення безпеки: визначте вимоги щодо захисту конфіденційності та цілісності даних учнів. Розгляньте заходи для запобігання несанкціонованому доступу до інформації, забезпечення безпечного зберігання та передачі даних, а також резервного копіювання та відновлення даних.
* Вимоги до продуктивності: визначте вимоги щодо швидкодії та ефективності програми. Розгляньте оптимізацію роботи з базою даних, швидкий доступ до інформації та мінімізацію затримок при виконанні операцій.
* Вимоги до масштабованості: визначте, як програма буде масштабуватися з плином часу та збільшенням обсягу даних. Розгляньте можливості розширення функціональності, обробки більшої кількості учнів та забезпечення підтримки ростущого навантаження.
* План тестування: розробіть план для тестування програми обліку учнів з метою перевірки її функціональності, надійності та відповідності вимогам. Визначте тести для різних сценаріїв використання, включаючи реєстрацію учнів, ведення присутності, оцінювання тощо.
* Уточнення термінів та бюджету: визначте потрібний термін розробки програми та доступні фінансові ресурси для її реалізації. Зробіть оцінку робочих годин, необхідних для розробки та впровадження програми, а також визначте потрібний бюджет для придбання необхідного обладнання або послуг.
* Наведені складові постановки задачі надають загальний огляд вимог до програми обліку учнів. Проте, важливо врахувати, що конкретні деталі та вимоги можуть відрізнятися в залежності від контексту і потреб користувачів. Рекомендується провести консультації зі зацікавленими сторонами, такими як вчителі, адміністратори школи та батьки, для отримання більш точної інформації та врахування їхніх потреб та пропозицій у постановці задачі.

2 ТЕХНІЧНЕ ЗАВДАННЯ

**Попередній аналіз** програмного продукту для обліку учнів включає деякі загальні аспекти, які можуть бути включені в такий аналіз:

Функціональність: програмний продукт для обліку учнів повинен мати широкий набір функцій, що дозволяють реєструвати, зберігати та оновлювати інформацію про учнів. Це може включати основні дані про учнів, такі як ПІБ, дата народження, контактна інформація, інформацію про батьків, статевій ідентифікації тощо. Також, можуть бути функції для ведення журналу відвідуваності, успішності, класного розкладу та інших важливих даних. Зручність використання: програмний продукт повинен бути інтуїтивно зрозумілим та простим у використанні для адміністраторів та вчителів. Інтерфейс користувача має бути зрозумілим, зручним та легко навігованим, щоб користувачі могли швидко знаходити необхідну інформацію та виконувати потрібні дії. Безпека даних: оскільки в програмному продукті міститься чутлива особиста інформація про учнів, важливо, щоб він мав потужні механізми безпеки даних. Це включає захист від несанкціонованого доступу до даних, резервне копіювання та відновлення даних, шифрування і т.д. Гнучкість та налаштовуваність: Кожна школа може мати власні потреби щодо обліку учнів, тому програмний продукт повинен бути гнучким і налаштовуваним. Це означає, що адміністратори матимуть можливість налаштовувати систему відповідно до своїх потреб, додавати спеціальні поля, настраювати права доступу, створювати звіти та інші функції, які відповідають їх вимогам.

Інтеграція з іншими системами: у школах можуть використовуватися інші програмні продукти, такі як електронний журнал, система електронного навчання тощо. Важливо, щоб програмний продукт для обліку учнів мав можливість інтегруватися з цими системами для обміну даними та покращення продуктивності.

Підтримка та навчання: Компанія-розробник програмного продукту повинна забезпечувати належну підтримку клієнтів, включаючи вирішення технічних проблем, надання оновлень та поліпшень, а також навчання користувачів. Наявність документації, онлайн-ресурсів та навчальних матеріалів є важливою для ефективного використання програмного продукту.

Масштабованість: якщо програмний продукт планується використовувати в школі або шкільному окрузі з великою кількістю учнів, важливо мати на увазі масштабованість системи. Вона повинна бути здатна обробляти великі обсяги даних та забезпечувати швидкий доступ до інформації навіть при зростанні обсягу даних.

**Основні вимоги** до програмного продукту обліку учнів можуть варіюватися залежно від потреб освітнього закладу, але основні з них включають:

Реєстрація учнів: система повинна мати можливість реєстрації учнів зі збереженням основної інформації, такої як прізвище, ім'я, дата народження, адреса, контактні дані і батьківська інформація.

Класифікація і групування: програмне забезпечення повинно мати можливість класифікації учнів за різними критеріями, такими як вік, клас, спеціалізація, рівень навчання тощо. Також може бути необхідно групувати учнів в класи або групи залежно від розкладу занять чи інших потреб.

Ведення академічного обліку: система повинна забезпечувати можливість внесення академічної інформації про учнів, такої як оцінки, відвідуваність, звіти про успішність, перевірки знань тощо. Вона також може підтримувати ведення журналів, документації та звітності.

Комунікація з батьками: програмний продукт повинен мати можливість надсилання повідомлень, сповіщень або звітів батькам учнів, щоб забезпечити зв'язок між освітнім закладом та батьками стосовно навчальних питань, розкладу, оголошень тощо.

Збереження особистих даних: враховуючи конфіденційність особистих даних учнів, програмне забезпечення повинно забезпечувати безпеку та захист особистих даних від несанкціонованого доступу або витоку інформації

Забезпечення доступу та безпеки: програмний продукт повинен мати механізми автентифікації та авторизації, щоб забезпечити, що лише авторизовані користувачі мають доступ до інформації про учнів. Важливо забезпечити безпеку даних шляхом застосування шифрування, заходів для запобігання несанкціонованому доступу та регулярного оновлення програмного забезпечення для виправлення потенційних вразливостей.

Гнучкість та розширюваність: програмний продукт повинен бути гнучким і легко адаптовуватися до змінних потреб освітнього закладу. Він може містити налаштування, які дозволяють встановлювати параметри відповідно до особливостей конкретного закладу. Крім того, важливо мати можливість розширення функціональності програмного забезпечення у майбутньому для задоволення зростаючих потреб освітнього закладу.

Інтеграція з іншими системами: у разі потреби програмний продукт обліку учнів повинен мати можливість інтеграції з іншими системами, такими як система управління навчальним процесом, система бібліотеки, фінансова система тощо. Це забезпечить обмін інформацією та уніфіковану роботу різних систем.

Ці вимоги є загальними, і їх можна доповнювати або змінювати залежно від конкретних потреб і вимог освітнього закладу. При виборі програмного продукту обліку учнів важливо враховувати ці вимоги та консультуватися зі зацікавленими сторонами, щоб забезпечити відповідність системи потребам освітнього закладу.

**Функціональні характеристики** програми обліку учнів можуть включати:

Реєстрація учнів: Можливість внесення інформації про учнів, включаючи прізвище, ім'я, дату народження, адресу, контактні дані, батьківську інформацію та інші особисті дані.

Класифікація і групування: можливість класифікувати учнів за різними критеріями, такими як вік, клас, спеціалізація, рівень навчання і статус (активний, відрахований тощо). Також можливість групувати учнів в класи, групи чи потоки відповідно до розкладу занять або інших потреб.

Ведення академічного обліку: можливість внесення академічної інформації про учнів, такої як оцінки, відвідуваність, звіти про успішність, перевірки знань, навчальні плани тощо. Забезпечення ведення журналів, планування уроків, надання домашніх завдань і створення звітів для вчителів та адміністраторів.

Комунікація з батьками та вчителями: можливість надсилання повідомлень, сповіщень, повідомлень про події та звітів батькам учнів і вчителям. Включає можливість нагадувань про збори, важливі дати, зміни в розкладі, оцінки та інші навчальні питання.

Розклад занять: можливість створення і керування розкладом занять для учнів і вчителів, враховуючи різні предмети, групи і класи. Забезпечення можливості зміни та оновлення розкладу за необхідності.

Генерація звітів і статистики: можливість створювати звіти про академічну продуктивність учнів, відвідуваність, оцінки, прогрес тощо. Забезпечення можливості аналізу та візуалізації даних для допомоги в прийнятті рішень в освітньому процесі.

Електронний журнал: можливість ведення електронного журналу, де можна записувати академічну інформацію про учнів, включаючи оцінки, відвідуваність, коментарі вчителів тощо. Забезпечення доступу для вчителів і адміністраторів до відповідних даних.

Розклад занять і планування: можливість створювати розклад занять для класів та вчителів, враховуючи різні предмети, групи, часові інтервали та ресурси. Забезпечення можливості планування уроків, розподілу вчителів і аудиторій, оновлення розкладу змін у реальному часі.

Електронний доступ для батьків і учнів: надання можливості батькам і учням отримувати доступ до відповідних даних, таких як оцінки, розклад занять, домашні завдання, повідомлення від вчителів і адміністраторів через електронну платформу або мобільний додаток.

Інтеграція з іншими системами: можливість інтеграції з іншими системами, такими як система управління навчальним процесом, система бібліотеки, система фінансового обліку, що забезпечує обмін даними та уніфіковану роботу різних систем.

**Вимоги до інтерфейсу** програмного продукту обліку учнів включають:

інтуїтивний та зрозумілий інтерфейс: Інтерфейс повинен бути легким у використанні і зрозумілим для користувачів будь-якого рівня технічної підготовки. Меню, кнопки та елементи керування повинні бути розташовані логічно і легкодоступні, щоб користувачі могли швидко зорієнтуватися в системі.

Ергономіка: інтерфейс повинен бути зручним для використання, з урахуванням принципів ергономіки. Розміщення елементів, розмір шрифту, кольорова палітра повинні забезпечувати зручність та ефективність взаємодії з програмою.

Персоналізація: можливість налаштовувати інтерфейс залежно від потреб користувача. Користувачі повинні мати можливість налаштувати вигляд, мову, тему оформлення та інші параметри відповідно до своїх вподобань і потреб.

Мультимовність: забезпечення підтримки кількох мов, щоб користувачі з різних країн та культур могли використовувати програму на своїй рідній мові.

Інтеграція з іншими системами: забезпечення зручної інтеграції з іншими системами, які використовуються в освітньому закладі, наприклад, система електронного документообігу, система електронного журналу, система електронних підручників тощо.

Респонсивний дизайн: інтерфейс повинен бути адаптивним до різних типів пристроїв, таких як комп'ютери, планшети і смартфони, забезпечуючи зручний доступ.

**Вимоги до алгоритму** роботи програми обліку учнів включають:

Реєстрація учнів: Програма повинна мати механізм для реєстрації нових учнів, який включає введення особистих даних, таких як ім'я, прізвище, дата народження, контактна інформація тощо. Дані повинні зберігатися в системі для подальшого використання.

Класифікація і групування: Програма повинна дозволяти класифікувати учнів за різними критеріями, такими як вік, клас, спеціалізація, рівень навчання тощо. Також вона повинна забезпечувати можливість групування учнів в класи, групи або потоки відповідно до розкладу занять або інших потреб.

Ведення академічного обліку: програма повинна мати можливість внесення академічної інформації про учнів, такої як оцінки, відвідуваність, звіти про успішність, перевірки знань тощо. Вона повинна також забезпечувати можливість ведення журналів, планування уроків, надання домашніх завдань і створення звітів для вчителів та адміністраторів.

Комунікація з батьками та вчителями: програма повинна мати можливість надсилання повідомлень, сповіщень, повідомлень про події та звітів батькам учнів і вчителям. Вона повинна дозволяти нагадувати про збори, важливі дати, зміни в розкладі, оцінки та інші навчальні питання шляхом електронної пошти, повідомлень в месенджерах або інших зручних способів комунікації.

Розклад занять: програма повинна забезпечувати можливість створення і керування розкладом занять для учнів і вчителів. Алгоритм роботи повинен враховувати різні параметри, такі як предмети, групи, класи, вчителі, часові інтервали та інші фактори, щоб створювати оптимальні розклади занять.

Автоматичне обчислення оцінок: програма повинна мати алгоритм для автоматичного обчислення оцінок учнів на основі введених даних про їх академічні досягнення. Алгоритм повинен враховувати вагу оцінок, методику обчислення середнього балу, наявність додаткових бонусів або покарань та інші фактори, які впливають на процес оцінювання.

Автоматичне формування звітів: програма повинна мати алгоритм для автоматичного формування звітів про академічні досягнення учнів. Алгоритм повинен використовувати зібрані дані про оцінки, відвідуваність, звіти про успішність та інші показники, щоб створювати звіти, які можуть бути використані вчителями, батьками або адміністрацією школи.

Забезпечення безпеки даних: програма повинна мати алгоритм для захисту особистих даних учнів, вчителів та батьків. Алгоритм повинен включати механізми шифрування, контролю доступу до інформації, резервного копіювання та інші заходи для забезпечення конфіденційності і цілісності даних.

**Вимоги до надійності** програми обліку учнів включають:

стійкість до помилок: програма повинна бути стійкою до помилок, таких як некоректні дані введення або непередбачувані ситуації. Вона повинна коректно обробляти помилки і надавати користувачам зрозумілі повідомлення про виниклі проблеми.

Захист від втрати даних: програма повинна мати механізми для запобігання втраті даних. Це можуть бути регулярне резервне копіювання даних, механізми відновлення даних після аварійного вимкнення або інших непередбачуваних ситуацій.

Масштабованість: програма повинна бути здатною працювати ефективно при збільшенні обсягу даних і кількості користувачів. Вона повинна бути здатною масштабуватися і обробляти великі обсяги інформації без збоїв або втрати продуктивності.

Безпека даних: програма повинна забезпечувати високий рівень захисту особистих даних учнів, вчителів та батьків. Вона повинна використовувати сучасні методи шифрування, контролю доступу та ідентифікації для запобігання несанкціонованому доступу до конфіденційної інформації.

Надійність зв'язку: якщо програма використовує мережу для комунікації з іншими системами або користувачами, вона повинна бути надійною у встановленні та підтримці зв'язку. Вона повинна вміти відновлювати з'єднання після втрати зв'язку і забезпечувати надійну передачу даних.

**Вимоги до складу і параметрів технічних засобів** для програми обліку учнів можуть включати:

апаратне забезпечення: Вимоги до комп'ютерів, серверів або хмарних ресурсів, на яких буде встановлена програма. Це може включати процесор, оперативну пам'ять, місткість жорсткого диска, мережеві можливості та інші компоненти, які забезпечують ефективну роботу програми.

Операційна система: вимоги до операційної системи, яка повинна бути підтримувана програмою. Наприклад, програма може вимагати операційну систему Windows, macOS або Linux певної версії.

Веб-браузер: якщо програма базується на веб-технологіях, можуть бути вимоги до веб-браузера. Наприклад, програма може підтримувати останні версії браузерів, таких як Google Chrome, Mozilla Firefox, Safari або Microsoft Edge.

Мережеві засоби: якщо програма використовується в мережевому середовищі, можуть бути вимоги до мережевого обладнання, такого як маршрутизатори, комутатори та інші мережеві компоненти. Вони повинні забезпечувати швидку та стабільну передачу даних між користувачами та сервером програми.

**Вимоги до інформаційної програмної сумісності** включають:

сумісність з операційною системою: Програма повинна бути розроблена таким чином, щоб працювати на різних операційних системах, таких як Windows, macOS, Linux тощо. Вона повинна підтримувати необхідні версії операційної системи та використовувати прийняті стандарти програмування для цих платформ.

Сумісність з браузерами: Якщо програма базується на веб-технологіях, вона повинна бути сумісною з різними веб-браузерами, такими як Google Chrome, Mozilla Firefox, Safari, Microsoft Edge тощо. Вона повинна коректно відображатися і працювати в цих браузерах без помилок або непередбачених проблем.

Сумісність зі стандартами обміну даними: Програма повинна дотримуватися прийнятих стандартів обміну даними, таких як XML (eXtensible Markup Language), JSON (JavaScript Object Notation) або CSV (Comma-Separated Values). Це забезпечить можливість обміну інформацією з іншими системами, які підтримують такі стандарти.

Сумісність з базами даних: якщо програма використовує базу даних, вона повинна бути сумісною з певною системою управління базами даних (СУБД), такою як MySQL, PostgreSQL, Microsoft SQL Server тощо. Вона повинна використовувати прийняті стандарти SQL (Structured Query Language) для взаємодії з базою даних.

Сумісність з іншими системами: програма повинна мати можливість інтеграції з іншими системами, які використовуються в освітньому закладі.

3 ОБҐРУНТУВАННЯ ЗАСОБІВ РЕАЛІЗАЦІЇ

Графічний інтерфейс користувача (GUI) - це спосіб взаємодії користувача з комп'ютерною програмою, що використовує графічні елементи, такі як кнопки, поля введення, вікна, меню тощо. Основні компоненти графічного інтерфейсу користувача включають:

Вікна - це графічні елементи, які представляють собою окремі вікна програми. Вони можуть містити інші елементи, такі як кнопки, поля введення, тексти тощо.

Кнопки - це графічні елементи, які користувач може натискати для виконання певних дій. Кнопки можуть мати текст або іконку, яка ілюструє їх призначення.

Поля введення - це графічні елементи, які дозволяють користувачеві вводити текст або числа. Поля введення можуть бути різних розмірів і мати різний тип введення, такий як текстовий рядок, число або дата.

Меню - це графічний елемент, який містить список опцій, з яких користувач може вибрати. Меню може бути статичним або динамічним, тобто залежно від контексту можуть з'являтися різні опції.

Текстові блоки - це графічні елементи, які містять текстову інформацію для користувача. Текстові блоки можуть містити заголовки, інструкції, повідомлення про помилки тощо.

Панелі інструментів - це графічні елементи, які містять набір інструментів, що допомагають користувачеві виконувати певні дії швидше і легше.

Побудова інтерфейсу програмного продукту для обліку учнів може включати наступні елементи: Головне меню, яке містить опції, такі як "Додати учня", "Редагувати учня", "Видалити учня", "Переглянути список учнів" тощо.

Форма додавання/редагування учня, яка містить поля для введення інформації про учня, такі як ім'я, прізвище, дата народження, клас тощо. Форма може також містити кнопки для збереження або скасування дії.

Список учнів, який містить інформацію про всіх учнів, що зберігаються в програмі. Кожен елемент списку може містити кратку інформацію про учня, таку як ім'я та прізвище, клас, дата народження тощо.

Кнопки дій, які дозволяють користувачеві виконувати різні дії з учнями, такі як додавання, редагування, видалення та перегляд.

Панель інструментів, яка містить швидкий доступ до найчастіше використовуваних опцій програми, наприклад, кнопки для додавання або перегляду учнів.

Довідка, яка містить інформацію про використання програми та її функціональність, а також може містити посилання на ресурси з додатковою інформацією про облік учнів.

Інтерфейс програмного продукту для обліку учнів може бути розроблений з такими сторінками:

* Головна сторінка: на головній сторінці можуть бути розміщені загальні дані про продукт та функціональні кнопки для швидкого доступу до інших сторінок. Також можна розмістити блок новин або повідомлень для користувачів.
* Сторінка відвідуваності: на цій сторінці можна зберігати дані про те, коли студенти були відсутні або зараховані в університет. Користувачі можуть додавати або видаляти записи зі списку.
* Сторінка розкладу: на цій сторінці можна розмістити розклад занять для кожної групи. Користувачі можуть переглядати розклад на кожен день, додавати або видаляти заняття.
* Сторінка груп: на цій сторінці можна переглядати список всіх груп, що зареєструвались в системі. Користувачі можуть створювати нові групи, додавати або видаляти учасників.
* Сторінка студентів: на цій сторінці можна переглядати інформацію про кожного студента. Користувачі можуть додавати або видаляти студентів зі списку.
* Меню: на сторінці меню можуть бути розміщені функціональні кнопки для швидкого доступу до всіх сторінок.

Ці сторінки можуть бути розміщені на веб-сайті або в додатку, який можна встановити на комп'ютер або мобільний пристрій. Крім того, можна додати функціонал для відправлення сповіщень або повідомлень студентам та викладачам.

Оформлення інтерфейсу було розроблено в мінімалістичному стилю з упором на приємну кольорову гамму і анімації, також , щоб він був інтуєтивно понятний та простий у використанні. Приклад вигляду інтерфейсу на (Рис. 2.1)

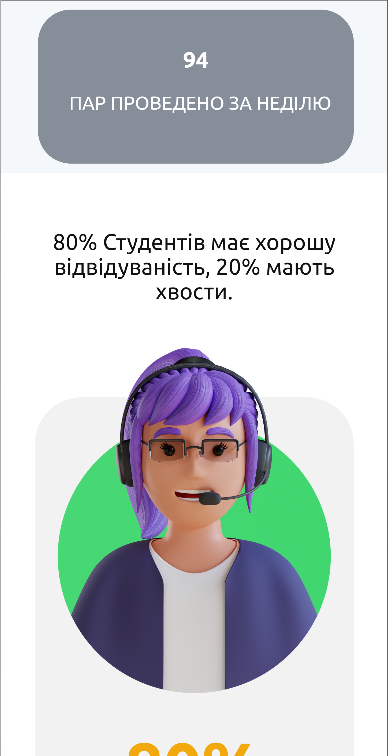


Рисунок 2.1 Приклад інтерфейсу програми

3.1 ВИБІР МОВИ ТА СЕРИДОВИЩА ПРОГРАМУВАННЯ

При виборі мови та середовища програмування для сайту обліку учнів, варто враховувати наступні фактори:

Функціональність та потреби проекту: розгляньте, які функції і можливості повинен мати сайт обліку учнів. Це можуть бути реєстрація учнів, ведення журналу, розклад занять, надання оцінок, комунікація з батьками і вчителями тощо. Переконайтеся, що мова та середовище програмування мають необхідні інструменти та бібліотеки для реалізації цих функцій.

Досвід розробника: Врахуйте досвід розробника або команди, яка буде працювати над проектом. Якщо вони володіють деякою мовою програмування або середовищем, то це може бути вагомим аргументом для вибору цієї мови або середовища. Наявність досвіду допоможе зменшити час розробки та забезпечити якість коду.

Підтримка та екосистема: переконайтеся, що мова та середовище програмування мають активну спільноту розробників, достатню кількість ресурсів, документацію та підтримку. Це дозволить вам швидко знайти відповіді на питання, розв'язати проблеми та отримати підтримку у випадку потреби.

Швидкодія та масштабованість: розгляньте вимоги до швидкодії та масштабованості сайту. Якщо передбачається велика кількість користувачів або потреба в високій продуктивності, варто звернути увагу на мови та середовища, які добре справляються з такими завданнями, наприклад, Java, Python.

Судячи по цим параметрам було вибране серидовище Android Studio і такі мови і засоби для реалізації: HTML, CSS, Java Script, Submlime text, Adobe Photoshop.

3.2 ВИБІР СИСТЕМИ УПРАВЛІННЯ БАЗАМИ ДАНИХ

При виборі системи управління базами даних (СУБД) для сайту обліку учнів, варто враховувати наступні фактори:

Функціональність: переконайтеся, що обрана СУБД має необхідні функції та можливості для збереження, організації та маніпулювання даними, які вам потрібні. Розгляньте можливості реляційних та нереляційних СУБД, таких як MySQL, PostgreSQL, MongoDB, чи інших, які відповідають вашим потребам.

Продуктивність: розгляньте вимоги до продуктивності сайту. Якщо передбачається велика кількість запитів до бази даних або потреба в швидкому доступі до даних, важливо вибрати СУБД, яка демонструє високу продуктивність та оптимізована під ваші потреби.

Масштабованість: розгляньте, як СУБД масштабується з плином часу та зростанням обсягу даних. Переконайтеся, що вибрана СУБД може ефективно працювати зі зростаючими обсягами даних та забезпечувати швидкий доступ до них.

Сумісність: врахуйте, яка СУБД буде сумісна з вибраною мовою програмування та іншими технологіями, які ви плануєте використовувати на вашому сайті. Переконайтеся, що є підтримка та зручність інтеграції між СУБД та іншими компонентами вашої інфраструктури.

Безпека: забезпечення безпеки даних є важливим аспектом для сайту обліку учнів. Виберіть СУБД, яка має механізми захисту даних, такі як автентифікація, авторизація, шифрування, контроль доступу до бази даних, а також механізми резервного копіювання та відновлення даних.

Спільнота та підтримка: перевірте, чи є активна спільнота користувачів та розробників СУБД, з якою ви можете отримувати підтримку, вирішувати проблеми та отримувати оновлення та патчі для безпеки.

Ліцензування та вартість: розгляньте ліцензійні умови СУБД, а також вартість її використання. Деякі СУБД можуть бути безкоштовними або мають відкритий вихідний код, тоді як інші можуть бути комерційними продуктами з платною ліцензією. Переконайтеся, що ви розумієте ліцензійні умови та вартість використання обраної СУБД.

При виборі СУБД для сайту обліку учнів варто ретельно проаналізувати ці фактори та врахувати ваші конкретні потреби та вимоги. Краще вибрати СУБД, яка найкраще підходить для вашого проекту та забезпечить ефективну роботу з даними вашої освітньої системи, тому для мого проекту було вибрано гугл таблиці.

4 ОПИС ЕТАПІВ РЕАЛІЗАЦІЇ

Реалізація проекту обліку учнів на сайті включає кілька етапів. Основні етапи реалізації можуть виглядати наступним чином:

Аналіз та визначення вимог: на цьому етапі проводиться детальний аналіз потреб та вимог до системи обліку учнів. Визначаються функціональні та нефункціональні вимоги, вимоги до інтерфейсу, безпеки, продуктивності, масштабованості та інших параметрів.

Проектування системи: на основі вимог створюється проект системи обліку учнів. Це включає проектування бази даних, структури даних, архітектури системи, інтерфейсу користувача та інших компонентів системи.

Розробка: на цьому етапі виконується розробка програмного коду системи обліку учнів. Залежно від вибраного середовища програмування та інструментів розробки, програмісти створюють функціональні модулі, забезпечують взаємодію з базою даних, реалізовують функціональність системи, дизайн інтерфейсу та інші компоненти.

Тестування: після завершення розробки виконується тестування системи. Це включає випробування функціональності, перевірку на відповідність вимогам, виявлення та виправлення помилок та недоліків. Тестування може включати модульне тестування, інтеграційне тестування, системне тестування та інші види тестування.

Впровадження: на цьому етапі система обліку учнів готується до впровадження в реальне виробниче середовище. Вона інтегрується з веб-сервером, базою даних та іншими необхідними компонентами інфраструктури. Виконується налаштування системи, перенесення даних із тестового середовища до продуктивного, перевірка працездатності та підготовка до використання.

Експлуатація та підтримка: після впровадження системи обліку учнів в реальне середовище відбувається її експлуатація. Забезпечується регулярна підтримка системи, включаючи виправлення помилок, оновлення та розширення функціональності за потреби. Здійснюється моніторинг та оптимізація продуктивності системи.

Навчання та документування: користувачі системи обліку учнів проводяться навчання з використання та адміністрування системи. Важливо створити документацію, яка описує функціональність системи, інструкції користувача та адміністратора, архітектуру системи та інші важливі аспекти.

Постійне вдосконалення: Після введення системи в експлуатацію важливо забезпечити постійне вдосконалення. Збираються фідбек користувачів, виявляються потреби в нових функціях та змінах. Здійснюються оновлення системи, а також проводяться аналіз та оптимізація для забезпечення найкращої продуктивності та задоволення потреб користувачів.

Варто зауважити, що реалізація проекту обліку учнів може мати свої особливості в залежності від конкретних вимог, технологій та ресурсів, що використовуються. При виконанні кожного етапу важливо забезпечити взаємодію та співпрацю між розробниками, аналітами, тестувальниками, адміністраторами та користувачами проекту. Крім того, реалізація проекту може включати додаткові етапи, такі як планування ресурсів, забезпечення безпеки даних, інтеграцію з іншими системами, впровадження змінного циклу життя (DevOps) та інші, які можуть бути необхідними в конкретному контексті проекту.

Загалом, успішна реалізація проекту обліку учнів на сайті вимагає систематичного підходу та керівництва процесами на кожному етапі. Правильне визначення вимог, грамотне проектування, якісна розробка, ретельне тестування, ефективне впровадження, надійна експлуатація, постійне вдосконалення та забезпечення підтримки - усі ці етапи співпрацюють для створення функціональної та надійної системи обліку учнів на сайті.

4 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕСПЕЧЕННЯ

Розробка програмного продукту обліку учнів в Android Studio може бути складним завданням, яке вимагає деяких знань і навичок в програмуванні для мобільних пристроїв. Однак, з правильними інструментами та підходами, це може бути досяжним завданням.

Для виконання завдання потрібно виконати наступні кроки для розробки програмного продукту обліку учнів в Android Studio:

1. Планування та дизайн: визначити основні функції продукту, які потрібні для обліку учнів, та розробити прототип дизайну інтерфейсу користувача. Цей процес можна виконати за допомогою інструментів, таких як Sketch або Figma.
2. Розробка бази даних: встановити та налаштувати базу даних SQLite для зберігання даних про учнів, їх класи, оцінки та іншу інформацію, яка потрібна для обліку учнів.
3. Розробка функцій додатка: розробити функції, які дозволяють додавати, редагувати та видаляти інформацію про учнів, їх класи, оцінки та іншу інформацію.
4. Розробка інтерфейсу користувача: створити інтерфейс користувача, який дозволяє користувачам додавати, редагувати та видаляти інформацію про учнів, їх класи, оцінки та іншу інформацію. Цей інтерфейс може бути розроблений за допомогою XML-файлів та Java-класів.
5. Тестування та налагодження: протестувати додаток та налаштувати його для оптимальної продуктивності та відповідності вимогам користувачів.
6. Опублікування додатку: опублікувати додаток на Google Play Store або іншій платформі, щоб користувачі могли завантажити та використовувати його.

Крім цього, варто використовувати хороші практики програмування, такі як розбиття програмного коду на маленькі функціональні блоки та використання патернів проектування для поліпшення структури додатку.

Також необхідно мати деякі знання Java-програмування, а також знання Android SDK та Android Studio. Можна скористатися різноманітними онлайн-ресурсами, такими як документація Android та відеоуроки, для отримання додаткової інформації та допомоги.

Для розробки програмного продукту буде використано наступні бібліотеки:

WebView - це бібліотека для Android, яка надає можливість відображати веб-сторінки в додатках. WebView вбудована в Android і дозволяє відображати HTML-сторінки, зображення, відео та інші ресурси з Інтернету без виходу з додатка.

WebView дозволяє відображати вміст веб-сторінок в межах додатка, забезпечуючи більш зручний та безпечний спосіб відображення вмісту, ніж відкривання зовнішнього браузера.

WebView надає ряд методів та властивостей для керування відображенням веб-сторінок, зокрема:

loadUrl() - метод для завантаження веб-сторінки за вказаним URL;

loadData() - метод для завантаження HTML-коду сторінки без необхідності переходу на зовнішній сервер;

goBack() та goForward() - методи для переходу до попередньої або наступної сторінки;

getSettings() - метод для отримання налаштувань WebView, зокрема можливість включення / вимикання JavaScript, кешування, зумування тощо.

WebView також дозволяє використовувати веб-сервіси з додатка, надаючи можливість взаємодії з веб-сайтами та додатками, що спрощує розробку і поліпшує користувацький досвід.

WebView є потужним інструментом для взаємодії з веб-сторінками в Android-додатках та може бути використана для реалізації різноманітних функцій, таких як відображення новин, форми зворотного зв'язку, рекламні банери та інше.

RecyclerView - це бібліотека, яка надає зручний і ефективний спосіб відображення списків даних в Android. RecyclerView дозволяє ефективно працювати з великими списками даних і забезпечує більш гнучкий і налаштовуваний підхід до відображення списків.



ButterKnife - це бібліотека для спрощення написання коду в Android Studio. Вона дозволяє швидко і легко прив'язувати компоненти інтерфейсу користувача до коду додатка, що робить розробку більш зручною і швидкою.

Меню для програми було написано по принципу схеми зображеної на (Рис. 3.1)

Рисунок 4.1 Схема меню ПЗ

Для початку потрібно написати код для ініціалізації роботи та імпорту потрібних бібліотек:

package com.example.test;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.annotation.TargetApi;  
import android.os.Build;  
import android.os.Bundle;  
import android.view.View;  
import android.webkit.WebResourceRequest;  
import android.webkit.WebView;  
import android.webkit.WebViewClient;

Даний код є класом MainActivity, який є головним екраном додатка Android. У методі onCreate() здійснюється ініціалізація і налаштування елементів екрану.

Спочатку встановлюється вміст екрану з використанням setContentView(R.layout.activity\_main), де activity\_main є макетом екрану, збереженим у файлі XML.

@Override  
protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);

Наступний рядок встановлює флаги системного інтерфейсу для екрану, що приховують панель навігації та статус-бар. Це робиться за допомогою getWindow().getDecorView().setSystemUiVisibility().

Далі створюється об'єкт WebView, що дозволяє відображати веб-сторінки в додатку, і налаштовується включення JavaScript для нього. Наступний рядок встановлює URL-адресу сторінки, яка буде завантажена в WebView.

Останнім кроком встановлюється WebViewClient, який відповідає за обробку подій, що пов'язані з відображенням веб-сторінок. В даному випадку він перенаправляє запит на завантаження сторінки в WebView.

View.SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE  
  
 | View.SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION  
  
 | View.SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN  
  
 | View.SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION  
  
 | View.SYSTEM\_UI\_FLAG\_FULLSCREEN);  
  
 WebView webView = findViewById(R.id.webView);  
 webView.getSettings().setJavaScriptEnabled(true);  
 webView.loadUrl("https://selfdestructi0n.github.io/Cursova/Головна.html");  
  
 WebViewClient webViewClient = new WebViewClient() {  
  
 @SuppressWarnings("deprecation") @Override  
  
 public boolean shouldOverrideUrlLoading(WebView view, String url) {  
  
 view.loadUrl(url);  
  
 return true;  
  
 }  
  
  
  
 @TargetApi(Build.VERSION\_CODES.N) @Override  
  
 public boolean shouldOverrideUrlLoading(WebView view, WebResourceRequest request) {  
  
 view.loadUrl(request.getUrl().toString());  
  
 return true;  
  
 }  
  
 };  
 webView.setWebViewClient(webViewClient);  
 }  
}

Для подальшої роботи та підключення програми потрібно підключити її до інтернету це робиться за допомогою даного коду:

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
 <uses-permission android:name="android.permission.INTERNET"/>  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Test"  
 tools:targetApi="31">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true"  
 android:configChanges="orientation|keyboardHidden|screenSize">>  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

Цей код є файлом маніфесту (Manifest) для додатка на платформі Android. У маніфесті описується конфігурація додатка, його компоненти, вимоги до привілеїв, дозволи та інші налаштування.

* У даному коді визначається використання дозволу INTERNET, який потрібен для доступу до Інтернету в додатку.
* У тезі application встановлюються деякі налаштування, такі як:
* allowBackup - дозволяє системі резервне копіювання даних додатка.
* dataExtractionRules - вказує на наявність правил для екстракції даних з додатка.
* fullBackupContent - вказує на наявність правил для повного резервного копіювання даних додатка.
* icon - іконка додатку.
* label - назва додатку, яку буде використовувати система.
* supportsRtl - дозволяє використання зворотнього напрямку для розміщення тексту.
* theme - тема додатку, яку буде використовувати система.

Також в коді описано активність (activity) з назвою MainActivity, яка є головною активністю додатку. В ній визначено, що активність може бути експортованою і її конфігурація залежить від орієнтації екрана, наявності клавіатури та розміру екрану. Також вказано, що активність є точкою входу (Launcher) в додаток.

Далі ми створюємо файл activity\_mein.xml для цього коду:

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <WebView  
  
 android:id="@+id/webView"  
  
 android:layout\_width="match\_parent"  
  
 android:layout\_height="match\_parent"  
  
 />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Цей код є розміткою (layout) для додатку на платформі Android і відображає структуру і розміщення елементів інтерфейсу користувача.

У даному коді використовується ConstraintLayout, який дозволяє вирішувати проблему позиціонування елементів інтерфейсу користувача на екрані за допомогою задання обмежень (constraints) між ними.

У розмітці задано єдиний елемент WebView, який відображає вміст веб-сторінки. Цей елемент знаходиться всередині ConstraintLayout і має атрибути android:layout\_width та android:layout\_height, які задають його розміри на екрані. Значення "match\_parent" вказує, що WebView повинен займати всю доступну простору на екрані.

Також в розмітці використані атрибути xmlns:android, xmlns:app та xmlns:tools, які вказують на простори імен, де визначені атрибути, що використовуються в розмітці. Крім того, у ConstraintLayout є атрибут tools:context, який вказує на контекст додатку, з яким пов'язаний цей layout. У даному випадку, це MainActivity.

5 ОПИС ПРОГРАМНОГО ПРОДУКТУ

Програмний продукт обліку учнів на сайті - це комплексна система, яка дозволяє школам або навчальним закладам ефективно збирати, організовувати та управляти даними про учнів у цифровому форматі. Опис програмного продукту включає наступні аспекти:

Функціональність: програмний продукт надає різноманітні функції для обліку учнів, включаючи реєстрацію, збереження та оновлення особистих даних, контактної інформації, академічних досягнень, відомостей про відвідування, графіку занять та інших важливих відомостей.

Інтерфейс користувача: програмний продукт має зручний та інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам легко взаємодіяти з системою. Інтерфейс може бути веб-заснованим, що дозволяє отримувати доступ до системи через веб-браузер з будь-якого пристрою.

Звіти та статистика: програмний продукт забезпечує можливість створення різноманітних звітів та отримання статистичних даних про учнів. Це допомагає адміністраторам, вчителям та іншим зацікавленим особам отримати аналітичну інформацію про успішність, прогрес у навчанні, відвідуваність та інші показники.

Безпека даних: програмний продукт забезпечує захист особистих даних учнів. Він може використовувати механізми шифрування для збереження конфіденційної інформації, а також контроль доступу та аудит дій користувачів для запобігання несанкціонованому доступу.

Інтеграція з іншими системами: програмний продукт обліку учнів може бути інтегрований з іншими системами, що використовуються в навчальному закладі. Наприклад, він може бути підключений до системи електронного навчання, де вчителі можуть прямо з програмного продукту переглядати успішність учнів, призначати та оцінювати завдання. Також він може інтегруватись з системою розкладу занять, де адміністратори можуть автоматично отримувати інформацію про графік занять учнів. Інтеграція з іншими системами допомагає уникнути дублювання даних та покращує ефективність роботи.

Модульність та розширюваність: програмний продукт може бути розроблений з урахуванням модульної архітектури, що дозволяє легко розширювати його функціональність. Наприклад, можливо додати нові модулі для ведення обліку фінансів, здоров'я учнів, електронної документації тощо. Це дозволяє пристосувати програмний продукт до потреб конкретного навчального закладу.

Технічна підтримка: команда розробників забезпечує технічну підтримку програмного продукту, включаючи виправлення помилок, оновлення та надання консультацій користувачам. Також можуть надаватися інструкції користувачам щодо використання та адміністрування системи.

Процес реалізації програмного продукту обліку учнів може варіюватись залежно від конкретних вимог та потреб навчального закладу. Важливо спільно з розробниками визначити потреби та вимоги, провести аналіз, розробити план впровадження та забезпечити належну підтримку під час впровадження програмного продукту. Крім того, важливим етапом є навчання користувачів, яке може включати тренінги та надання посібників для ефективного використання системи.

Опис програмного продукту може включати інші додаткові функції та особливості відповідно до потреб конкретного навчального закладу. Наприклад, можливість сповіщень та комунікації з батьками, інтеграцію з системою оцінювання або планування навчальних програм, підтримку багатомовності для міжнародних навчальних закладів, електронну реєстрацію на події та інші.

У кінцевому результаті програмний продукт обліку учнів на сайті спрощує та автоматизує процеси збирання, організації та управління даними про учнів. Він допомагає забезпечити централізований доступ до важливої інформації, підвищує ефективність роботи навчального закладу та полегшує спілкування з батьками та учнівською громадськістю.

5.1 РЕЗУЛЬТАТ РОБОТИ ПРОГРАМИ

Результат роботи програми обліку учнів на сайті включає різноманітні аспекти, які залежать від поставлених завдань і вимог до системи. Основні результати:

Збереження та організація даних: Програма забезпечує можливість зберігати та організовувати інформацію про учнів, таку як особисті дані, контактна інформація, академічні досягнення, прогрес у навчанні та інші відомості.

Облік учнів: Програма дозволяє проводити облік учнів, включаючи реєстрацію нових учнів, внесення змін до існуючих записів, відстеження присутності, відомостей про відвідування та інших аспектів, пов'язаних з учнівським обліком.

Генерація звітів і статистики: Програма надає можливість створювати звіти та отримувати статистичну інформацію про учнів, їх успішність, відвідуваність, активність та інші показники. Це допомагає адміністраторам, вчителям та іншим зацікавленим особам отримати уявлення про стан навчального процесу та здійснити аналіз даних.

Інтеграція з іншими системами: Якщо необхідно, програма може бути інтегрована з іншими системами, такими як системи електронного навчання, системи розкладу занять або системи електронного звітності. Це дозволяє забезпечити обмін даними та використання спільних ресурсів для оптимізації роботи.

Забезпечення безпеки даних: Програма забезпечує захист особистих та конфіденційних даних учнів. Вона може застосовувати різні заходи безпеки, такі як шифрування даних, контроль доступу, аудит дій користувачів та інші механізми для запобігання несанкціонованому доступу до інформації.

Автоматизація рутинних завдань: Програма спрощує та автоматизує багато рутинних завдань, пов'язаних з учнівським обліком, таких як реєстрація, формування звітів, розсилка повідомлень, оновлення даних тощо. Це дозволяє ефективно використовувати час та ресурси працівників.

Покращення комунікації: Програма може надавати можливості для зручної комунікації між учнями, батьками, вчителями та адміністрацією школи. Це може включати обмін повідомленнями, нагадування про події та зустрічі, спільну роботу над завданнями та інші функції, що сприяють зміцненню спільноти.

Загалом, результат роботи програми обліку учнів на сайті полягає в ефективному та систематичному збереженні, обліку та управлінні даними про учнів, полегшенні роботи працівників школи, покращенні комунікації та забезпеченні безпеки і надійності інформації.



Рисунок 5.1 Приклад роботи програми

5.2 ІНСТРУКЦІЯ КОРИСТУВАЧА

Інструкція для користувача програми забеспечення обліку учнів:

1. Встановіть програму на свій комп'ютер або мобільний пристрій з операційною системою Windows, macOS, iOS або Android. Для цього завантажте програму з офіційного сайту або магазину додатків.
2. Після запуску програми оберіть вкладку "Групи" та створіть нову групу або виберіть вже створену групу.
3. Вкладка "Розклад" дозволить вам створити розклад занять для обраної групи. Натисніть кнопку "Додати" та заповніть поля з назвою заняття, датою та часом.
4. У вкладці "Студенти" можна додавати нових студентів до групи та змінювати їхні дані (ім'я, прізвище, електронна адреса тощо). Виберіть групу, до якої потрібно додати студентів, та натисніть кнопку "Додати".
5. Вкладка "Відвідуваність" дозволяє вести облік відвідування студентів занять. Оберіть групу та дату, потім відзначайте відвідування студентів.
6. У вкладці "Меню" можна знайти інші корисні налаштування програми та змінити їх налаштування.

Наступні операційні системи підтримують програму обліку учнів: Windows, macOS, iOS та Android. Якщо ви використовуєте іншу операційну систему, переконайтеся, що вона підтримує програму, перш ніж встановлювати її.

5.3 ІНСТРУКЦІЯ АДМІНІСТРАТОРА

Інструкція для адміністратора програмного продукту обліку учнів на сайті містить наступні кроки та рекомендації:

1. Встановлення та налаштування програмного продукту:

Встановіть програмний продукт на сервер або хмарний хостинг згідно з документацією розробника.

Здійсніть необхідні налаштування, такі як з'єднання з базою даних та налаштування безпеки.

1. Створення облікових записів:

Створіть обліковий запис адміністратора для доступу до системи.

Залежно від потреб, створіть облікові записи для інших адміністраторів, вчителів та інших користувачів з відповідними ролями та правами доступу.

1. Налаштування основних параметрів:

Налаштуйте загальні параметри системи, такі як назва навчального закладу, логотип, адреса, контактна інформація тощо.

Встановіть правила щодо обробки та збереження особистих даних учнів згідно з вимогами конфіденційності.

1. Імпорт даних:

Якщо необхідно, здійсніть імпорт початкових даних про учнів, таких як імена, прізвища, контактна інформація, класи тощо, за допомогою відповідного інструменту імпорту.

Перевірте, чи відповідають дані формату та структурі системи після імпорту.

1. Налаштування ролей та прав доступу:

Визначте рівні доступу для користувачів з різними ролями, такими як адміністратори, вчителі, батьки.

Встановіть обмеження доступу до окремих функцій та модулів системи залежно від потреб та обов'язків користувачів.

1. Налаштування звітів та статистики:

Встановіть параметри для створення звітів та отримання статистичних даних про учнів, таких як успішність, прогрес у навчанні, відвідуваність тощо. Визначте критерії фільтрації та сортування даних для звітів. Налаштуйте регулярне автоматичне створення звітів та їхню розсилку відповідним користувачам.

1. Технічна підтримка та оновлення:

Забезпечте технічну підтримку користувачам програмного продукту, включаючи розв'язання технічних проблем та відповіді на запитання. Встановіть процедуру регулярного оновлення програмного продукту та його компонентів. Повідомляйте користувачів про оновлення, нові функції та поліпшення програмного продукту.

1. Забезпечення безпеки:

Застосовуйте необхідні заходи для забезпечення безпеки даних учнів. Моніторингуйте систему на наявність потенційних загроз та ризиків безпеки. Забезпечуйте резервне копіювання та відновлення даних для запобігання втраті інформації.

Ця інструкція є загальним описом кроків та рекомендацій для адміністратора програмного продукту обліку учнів на сайті. Залежно від конкретних потреб та налаштувань системи, можуть бути додаткові кроки та деталі, які необхідно врахувати. Рекомендується ознайомитися з документацією та підтримкою розробника програмного продукту для отримання більш докладної інформації.

5.4 РОБОТА ПРОГРАМНОГО ПРОДУКТУ В МЕРЕЖІ

Програмний продукт обліку учнів має працювати в мережевому середовищі за допомогою хосту на якому він розміщений, що дозволяє користувачам з різних пристроїв і місцезнаходжень отримувати доступ до системи і взаємодіяти з нею. Нижче наведено деякі аспекти роботи програмного продукту в мережі:

Веб-заснований доступ: програмний продукт може бути доступним через веб-браузер, що дозволяє користувачам відкривати його на будь-якому пристрої з доступом до Інтернету. Користувачі можуть входити в систему, використовуючи свої облікові записи, і отримувати доступ до функціональності продукту.

Аутентифікація та авторизація: програмний продукт зазвичай має механізми аутентифікації, які дозволяють користувачам підтвердити свою ідентичність перед отриманням доступу до системи. Після аутентифікації система проводить авторизацію, що визначає рівень доступу та права користувача.

Розподілений доступ до даних: програмний продукт зберігає дані про учнів в базі даних, до якої можуть мати доступ декілька користувачів одночасно. Це дозволяє адміністраторам, вчителям та іншим користувачам здійснювати операції з даними одночасно, забезпечуючи актуальну інформацію.

Синхронізація даних: у випадку, коли програмний продукт інтегрований з іншими системами, такими як система електронного навчання чи система розкладу занять, може бути необхідна синхронізація даних між цими системами. Наприклад, оновлення успішності учня в системі електронного навчання можуть автоматично відображатися в програмному продукті обліку учнів

6 ЕКОНОМІЧНА ЧАСТИНА

**6.1 Опис програмного продукту**

**Назва ПП: "Student registration"**

**6.1Характеристика виготовленого ПП:**

1. Назва програмного продукту: "Student Registration – веб програма".
2. Основні характеристики програмного продукту:

* Веб-вью програма для обліку та регистрації учнів в навчальному закладі.
* Можливість перегляду інформації про студентів та викладачів навчального закладу.
* Зручна система запису на обслуговування або консультації з менеджером нз.
* Розділ новин, який дозволяє користувачам отримувати актуальну інформацію про загальну статистику.
* Контактна інформація та соціальні медіа.

1. Програмне середовище та платформи:

* Веб-сайт розроблений на базі платформи Sublime Text, використовуючи HTML, CSS, JavaScript.
* Сумісність з популярними веб-браузерами (Chrome, Firefox, Safari, Edge) на різних операційних системах (Windows, macOS, Linux).

1. Ринки збуту програмного продукту:

* Україна:навчальні заклади усіх типів.
* Міжнародний: користувачі з інших країн, які мають інтерес до обліку у навчальних закладах в Україні або отримувати інформацію та консультації через веб-сайт.

**6.2 Визначення собівартості розробки ПІП**

Для керування ходом робіт і ведення всього проекту в цілому необхідна

посада керівника. Таким керівником при розробці даного ПП виступає керівник дипломного проекту . На дипломне проектуванні керівнику надається 5 годин по тарифікації – 120 грн. за годину . Для розробки програмного продукту та його подальшого налагодження необхідна участь програміста, в даному випадку програмістом є учень. Середня зарплата ІТ-спеціаліста в Івано-франківську становить 600 доларів. Учень – є початківцем на ІТ-ринку, тому його заробіток вимірюється в розмірі – 400 доларів за місяць, який потрібно перерахувати у гривневому еквіваленті. Для цього потрібно 400 доларів помножити на поточний курс долара відносно української гривні, станом на момент виконання ІТ-розробки. Для проведення тестування необхідна посада тестувальника. Вважається, що тестувальником є певна особа з середньою зарплатою 300 гривень за годину. Над роботою працює 2 робочі дні. Робочий день становить – 8 годин. Економічний супровід виконує викладач економічних дисциплін, який консультує виконання економічної частини дипломного проєкту. Консультування проводиться протягом 5 годин по тарифікації 110 грн. за годину.

Термін виконання завдання в кожного розробника ПП – різний, в залежності від складності розробки ПП. Враховуються тільки робочі дні, виключаються з терміну всі вихідні та святкові дні. Перелік святкових днів, які можуть припадати на час виконання розробки ПП – 1 січня, 7 січня, 8 березня, перший день Великодня, 1 травня, 9 травня, перший день Святої Трійці. Якщо святковий день припадає на вихідні дні, то перший робочий день, який слідує за ним є вихідним.

Склад виконавців роботи та розрахунок загального фонду заробітної плати Таблиця:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Склад виконавців | Місячний оклад | Погодинна ставка | Кількість годин, днів на виконання завдання |
| 1 | Керівник ДР |  | 120 грн/год | 5 год |
| 2 | Програміст | 400\*36.57=14628грн |  | 16 год |
| 3 | Тестувальник |  | 300 грн/год | 16 годин |
| 4 | Економіст |  | 110грн/год | 5 годин |

Заробітна плата – винагорода за працю залежно від кваліфікації працівника, складності, кількості, якості, терміну та умов виконуваної роботи, а також компенсаційні виплати і стимулюючі виплати.

До витрат на заробітну плату праці відносяться основна і додаткова заробітна плата персоналу, зайнятого безпосередньо при виконанні даної розробки ПП.

Основна заробітна плата (ЗПосн) складається з суми зарплати:

1. Керівника ДП

2. Програміста

3. Тестувальника

4. Економіста

1. Основна заробітна плата керівника (ЗПо.к.) розраховується за формулою:

**ЗПо.к =погодинна ставка \*кількість відпрацьованих годин ( грн., коп.)**

**ЗПо.к=120\*5=600 грн.**

2. Основна заробітна плата програміста (ЗПо.п.) розраховується за

формулою:

**ЗПо.п =середньоденна зарплата \*кількість відпрацьваних днів ( грн., коп.)**

**ЗПо.п=91,43\*16=1462,80грн**

**Середньоденна зарплата = Оклад / кількість робочих днів ( грн., коп.)**

**Середньоденна зарплата =14 628/20=731,4грн**

**Середньогодинна зарплата=731,4/8=91,43 грн**

3. Основна заробітна плата тестувальника (ЗПо.т.) розраховується за

формулою:

**ЗПо.т=зарплата за день \*кількість відпрацьваних днів ( грн., коп.)**

**ЗПо.т=300\*16=4800грн**

4. Основна заробітна плата економіста (ЗПо.е.) розраховується за

формулою:

**ЗПо.е = погодинна ставка \*кількість відпрацьваних годин ( грн., коп.)**

**ЗПо.к=110\*5=550грн**

Підсумовуємо всю основну зарплату виконавців ПП:

**ЗПосн. = ЗПо.к + ЗПо.п + ЗПо.т + ЗПо.е. ( грн., коп.)**

**ЗПосн.=600+1462,8+4800+550=7412,80грн**

Далі проводиться розрахунок додаткової заробітної плати (ЗПдод). Дадаткова зарплата – це всі виплати, які дораховуються до основного заробітку, у вигляді компенсаційних та заохочувальних виплат ( відпускні, премії, лікарняні, надбавки…). Припустимо, що додаткова зарплата становить 20% (Н дод ) від основної заробітної плати і розраховується за формулою:

**ЗПдод = ЗПосн ∗ Ндод. ( грн., коп.)**

**ЗПдод=7412,80\*20%=1482,56грн**

де Ндод – коефіцієнт додаткової зарплати, рівний 20%. Разом, загальний фонд заробітної плати становить:

**ЗП = ЗПосн + ЗПдод ( грн., коп.)**

**ЗП=7412,80+1482,56=8895,36грн**

**6.3 Нарахування ЄСВ**

Єдиний соціальний внесок (ЄСВ) – це той обов’язковий платіж, який здійснюється роботодавцями за своїх працівників при нарахуванні їм заробітної плати, винагород за цивільно-правовими договорами (ЦПД), а також деяких інших виплат. ЄСВ становить в Україні 22% від загального фонду нарахованої зарплати. ЄСВ розраховується за формулою:

**ЄСВ = ЗП ∗ % ЄСВ, ( грн., коп.)**

**ЄСВ=8895,36\*22%=1956,98грн**

де % ЄСВ = 22%;

**6.4 Розрахунок витрат на матеріали**

До витрат на матеріали включаються матеріали, необхідні для розробки даного ПП, враховуючи їх потребу, кількість та ціни на момент складання розрахунку. Зазначимо, що для розробки даного ПП необхідно: 1 шт. олівець, 1 шт. блокнот та кольорові олівці за потребою. Якщо існують інші матеріальні витрати, будь ласка, вкажіть їх. Ціни на матеріали вказуються згідно з даними про ціни на місцевому ринку. Розрахунок матеріальних витрат проводиться за такою формулою:

**М = ∑ (Q × Ц), (грн., коп.)**

**М=1\*5+50\*2+7\*10=240грн**

де М - сумарні витрати на матеріали, Q - кількість використаних одиниць матеріалів, Ц - ціна одиниці матеріалу.

Розрахунок матеріальних витрат можна представити у вигляді таблиці:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Назва матеріалів | Одиниці виміру | Ціна за одиницю, (грн. коп) | Кількість | Вартість,  (грн. коп) |
| 1 | Блокнот | шт. | 40 грн | 2 | 80 грн. |
| 2 | Ручка | шт. | 10грн | 2 | 20 грн. |
| 3 | Кава 3 в 1 | шт. | 7грн | 20 | 140 грн. |
|  | Всього витрат на матеріали |  |  |  | 240грн. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**6.5 Розрахунок витрат на використану обладнанням електроенергію**

Витрати на використану обладнанням електроенергію (Е) розраховуються за наступною формулою:

**Е = M × t × TкВт, (грн., коп.)**

Е = (M × t × TкВт) = ( (0,03 кВт/год × 10 год) × 2,64 грн./кВт год

Е = 0,3 кВт × 2,64 грн./кВт год

Е = 0,79 грн.

де M - потужність устаткування, виражена кількістю енергії, яку споживає обладнання за одиницю часу (кВт/година); t - кількість годин використання устаткування за період проведення розробки ПП; ТкВт - тариф, вартість використання 1 кВт електроенергії.

Для розрахунку витрат на використану обладнанням електроенергію необхідно знати потужність кожного пристрою та кількість годин його використання.

За наведеними в даних, потужність приладів для розробки ПП складає:

* комп'ютер: 1 кВт/година;
* ноутбук: 0,03 кВт/година;
* планшет: 0,03 кВт/година;
* смартфон: 0,03 кВт/година;

Тариф на електроенергію становить 2,64 грн./кВт годин.

Для розрахунку кількості годин використання устаткування за період проведення розробки ПП можна виходити з припущення, що устаткування працювало в середньому 4 години за 1 день виконання розробки.

**6.6. Розрахунок витрат на амортизацію устаткування.**

Амортизація - це систематичний розподіл вартості необоротних активів, що амортизується протягом їх строку корисного використання. У даному випадку застосовується метод прямолінійного списання.

Розрахунок річної амортизації устаткування (Ар) проводиться за формулою:

**Ар = ПВ / Т (грн., коп.)**

1. Ноутбук: Ар\_ = 15,000 / 4 = 3,750 грн.
2. Смартфон: Ар\_Смартфон = 7,000 / 4 = 1,750 грн

де ПВ - вартість персонального комп'ютера (ноутбука, смартфона, планшета, айфона); Т - термін корисного використання устаткування, що становить 4 роки.

Витрати на амортизацію вказуються за місячною амортизацією, яка розраховується за формулою:

**Ам = Ар / 12 місяців (грн., коп.)**

1. Ноутбук: Ам\_Ноутбук =3750/12=312.5грн
2. Смартфон: Ам\_Смартфон = 1750/12=145,83грн

Отриману місячну амортизацію ділимо на 30 діб і отримаємо денну амортизацію..

Ноутбук:312,5/30 = 10,42 грн.

Смартфон:145,83/30 = 4,86 грн.

Програміст працює 16 годин, тобто 2 робочих дні, відповідно :

Ноутбук 10,42\*2 = 20,84 грн.

Смартфон 4,86\*2=9,72 грн.

Отже, загальна амортизація становить :

20,84 + 9,72 = 30,56 грн.

Проведені розрахунок амортизації для кожного устаткування, враховуючи вартість та термін корисного використання:

**6.7 Розрахунок інших витрат**

До інших витрат на розробку ПП слід віднести:

1. Вартість оплати послуг зв'язку:
   1. Інтернет: 250 грн. на місяць.
   2. Телефон: 100 грн. на місяць.
2. Загальновиробничі витрати:
   1. Загальновиробничі витрати становлять 20% від суми основної зарплати.
   2. Загальновиробничі витрати = 0.2 \* **7412,88** грн. = 1482,58грн.

**6.8 Складання калькуляції розробки ПП.**

|  |  |  |
| --- | --- | --- |
| № | Назва калькуляційної статті | Сума, грн. коп. |
| 1 | Матеріальні витрати | 240 |
|  | *основна зарплата* | 7412,80 |
|  | *додаткова зарплата* | 1482,36 |
| 2 | Витрати на оплату праці | 8895,36 |
| 3 | Єдиний соціальний внесок | 1956,98 |
| 4 | Витрати на спожиту енергію | 1,27 |
| 5 | Амортизація основних засобів | 30,56 |
|  | *загальновиробничі витрати* | 1482,58 |
|  | *витрати на інтернет* | 250,00 |
|  | *витрати на мобільний зв’язок* | 100,00 |
| 6 | Інші витрати | 1757,58 |
| Повна собівартість розробки | | 12881,75 |

**6.9 Прогнозування комерційних ефектів від реалізації результатів ПП.**

Методологія розрахунку ціни реалізації розробки ПП:

1. Повна собівартість розробки \* % очікуваного прибутку: 12881,75 \* 0.50 = 6440,87 грн. (прибуток)
2. Повна собівартість розробки + прибуток: 12881,75 + 6440,87 =

19 322,6 грн. (додаткова вартість на яку нараховується ПДВ)

1. Додаткова вартість \* 20%: 19 322,6 \* 0,20 = 3864,52 грн. (сума ПДВ)
2. Додаткова вартість + ПДВ: 19 322,6 + 3864,52 = 23603,49 грн. (ціна розробки ПП)

Отже, прогнозована ціна реалізації розробки програмного продукту складає 23070,12 грн.

**6.10 Висновок**

Розрахунок собівартості та вартості є завершальним етапом створення програмних продуктів чи послуг. У даному розділі була проведена оцінка всіх кількісних витрат на розробку представленого в дипломній роботі програмного продукту, а також були здійснені розрахунки економічних показників щодо реалізації ПП на ринку аналогічних товарів.

На розробку даного програмного продукту були затрачені такі види витрат: матеріальні витрати, витрати на оплату праці, єдиний соціальний внесок, витрати на спожиту електроенергію, амортизація основних засобів та інші витрати. За результатами розрахунків була встановлена сума собівартості розробки програмного продукту яка дорівнює 12 816,87 грн.

Для реалізації програмного продукту була розрахована ціна розробки, яка включає додаткову вартість та суму ПДВ. Ця ціна визначається з урахуванням очікуваного прибутку, припускаючи, що очікується отримати 50% прибутку. Таким чином, встановлена ціна розробки програмного продукту становить 23187,12грн.

7 ОХОРОНА ПРАЦІ

**7.1 ЗНАЧЕННЯ ОХОРОНИ ПРАЦІ НА ВИРОБНИЦТВІ**

Охорона життя та здоров’я громадян у процесі їх трудової діяльності, створення безпечних та нешкідливих умов праці є одним з найважливіших державних за­вдань.

Головною метою охорони праці є створення на кожному робочому місці безпечних та нешкідливих умов праці, безпечної експлуатації обладнання, зменшення або повна нейтралізація дії шкідливих і небезпечних виробничих факторів на організм людини і, як наслідок, зниження рівня виробничого травматизму та професійних захворювань.

Проблеми створення безпечних і нешкідливих умов праці існували завжди. Однак, у період науково-технічного прогресу вони набули особливого значення, адже істотно зросла ціна кожного нещасного випадку та аварії. За приблизними оцінками щороку в світі внаслідок виробничого травматизму гине близько 100 тис. чоловік. Нещасні випадки на виробництві та професійні захворювання були й залишаються величезною людською трагедією, є причиною значних економічних втрат і призводять до тяжких соціальних наслідків.

Законодавчими актами, що визначають основні положення про охорону праці, є загальні закони України, а також спеціальні законодавчі акти. В основному законі України — Конституції питанням охорони праці присвяче­ні статті 43, 45 та 46.

В статті 43 Конституції України записано: «Кожен має право на працю, що включає можливість заробляти собі на життя працею, яку він вільно обирає, або на яку вільно погоджується», «Кожен має право на належні, безпечні і здорові умови праці, на заробітну плату, не нижчу від визначеної законом», «Використання праці жінок і Неповнолітніх на небезпечних для їхнього здоров’я роботах забороняється».

Кожен, хто працює, має право на відпочинок (ст. 45 Конституції України). Це право забезпечується наданням днів щотижневого відпочинку, а також оплачуваної щорічної відпустки, встановленням скороченого робочого дня щодо окремих професій і виробництв, скороченої тривалості роботи у нічний час.

У тексті статті 46 Конституції України вказано на те, що громадяни мають право на соціальний захист, що включає право на забезпечення їх у разі повної, часткової або тимчасової втрати працездатності, втрати годувальника, безробіття з незалежних від них обставин, а також у старості та в інших випадках, передбачених законом.

Основним законодавчим документом у галузі охорони праці є Закон України «Про охорону праці», дія якого поширюється на всі підприємства і організації незалежно від форм власності та видів їх діяльності, на усіх громадян, які працюють, а також залучені до праці на цих підприємствах.

В першій статті даного закону дається визначення охорони праці.

Охорона праці — це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних та лікувально-профілактичних заходів і засобів, спря­мованих на збереження здоров’я та працездатності людини в процесі праці.

Для користувачів ПК найбільш характерними нормативними документами є Вимоги щодо безпеки здоров’я працівників під час роботи з екранними пристроями, затверджені наказом Міністерства соціальної політики 14.02.2018 року №207 та Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. ДСанПіН 3.3.2.007-98.

**7.2 АНАЛІЗ ПОТЕНЦІЙНИХ ШКІДЛИВИХ ТА НЕБЕЗПЕЧНИХ ФАКТОРІВ**

В процесі роботи працівник вступає у взаємодію з обладнанням, інструментами, іншими працівниками. Крім того, на нього впливають різні параметри виробничого середовища, в якому відбувається праця – виробничі фактори. За своїм впливом та часом дії на працівника виробничі фактори поділяють на небезпечні та шкідливі.

До небезпечних виробничих факторів, які діють на працівника раптово та призводить до травми під час користуванням ПК, відносяться:

- наявність електричного струму;

- пошкодження або відсутність захисного заземлення або занулення;

- відсутність захисних огороджень (бокових стінок) на блоці живлення;

- рухомі елементи обладнання (вентилятор).

До шкідливих виробничих факторів, які діють на працівника поступово (тривалий час) і спричиняють виникнення професійних захворювань, відносяться:

- підвищений рівень постійного шуму;

- метеорологічні умови, особливо в осінньо-зимовий період;

- недостатнє освітлення робочого місця та робочої зони;

- підвищений рівень іонізуючого та електромагнітного випромінювань;

- підвищений рівень шуму;

- розумові перевантаження, монотонність праці та малорухомість впродовж робочого дня.

При влаштуванні на роботу інженер з охорони праці проводить з працівником вступний інструктаж, а безпосередній керівник проводить на робочому місці первинний інструктаж. Інструктаж проводиться на основі інструкцій з охорони праці, які розроблені на всі види робіт та обладнання на підприємстві. Для користувачів ПК розроблена інструкція з охорони праці під час роботи за персональним комп’ютером.

**7.3 ЗАБЕЗПЕЧЕННЯ ОПТИМАЛЬНИХ УМОВ ПРАЦІ**

Під час роботи на працівника впливає сукупність виробничого середовища та виробничого процесу, які складають умови праці. Щоб працівник міг плідно працювати, йому необхідно створити оптимальні умови праці для виконання нею своїх виробничих обов’язків. Оптимальні умови праці характеризуються відсутністю або доведеними до мінімуму впливу небезпечних та шкідливих виробничих факторів.

Мікроклімат виробничих приміщень характеризуються такими параметрами: температура повітря в межах 21-230 С (в холодні пору року) та 21-230 С (в теплу пору року); відносна вологість повітря 40-60% та швидкість руху повітря – 0,1 м/с. Такі параметри є найбільш сприятливі для виконання професійних обов’язків, а при їх невідповідності використовують додаткові заходи та засоби для їх покращення (вентиляція приміщення, кондеціонування).

Приміщення, де використовуються персональні комп’ютери, обов'язково встановлюється вуглекислотний вогнегасник.

**7.4 РОЗРАХУНОК ОСВІТЛЕННЯ**

Для освітлення приміщення, де встановлено робоче місце працівника, використовуємо люмінесцентні лампи, які енергетично є більш економічні та максимально наближаються до природного світла.

Метою розрахунку штучного освітлення є визначення потужності джерел світла за заданою освітленістю та кількості необхідних світильників і ламп.

Для розрахунку освітлення використовуємо метод світлового потоку.

Вихідні дані:

Е – нормована освітленість для робіт середньої точності при загальному освітленні приймаємо 500 лк;

Кз – коефіцієнт запасу, що враховує зниження освітлення в результаті забруднення та старіння ламп, приймаємо 1,5;

Z – коефіцієнт нерівномірності освітлення для люмінесцентних ламп 1,1;

Н – висота приміщення – 3,5 м;

А – довжина приміщення – 9м

В – ширина приміщення – 9м

п – кількість ламп в світильнику – 4;

тип світильника з люмінесцентними лампами ЛПО01;

*стелі –* коефіцієнт відбиття світла стелі - 70% (0,7)

*стін*  *–* коефіцієнт відбиття світла стін - 50% (0,5)

Основні формули для розрахунку:

світлового потоку ,

індексу приміщення - 

кількості світильників - 

У формулах використовуємо вихідні та табличні дані.

Розрахунок починаємо з визначення висоти підвісу світильника над робочою поверхнею за формулою:

*h= Н – hс - hр*

*hс* – відстань від світильника до стелі 0,4 м.

*hр* – висота робочих поверхонь 0,8 м.

h= 3,5 – 0,4 – 0,8 = 2,3 м

Відстань між світильниками *L* визначається за формулою ,

L= 0,7 х 2,3 = 1,61 м

Кількості світильників (N), яка необхідна для освітлення дільниці, визначається за формулою: ,

N= 9 х 9 : 1,61 2= 81 : 2,59 = 31,27 шт.

Приймаю кількість світильників 31 шт. і розташовую їх у чотири ряди.

Після цього визначаю показник приміщення *і* за формулою:

 = 9 х 9 : 2,3 (9 + 9) = 81 : 41,4 = 2.

Відповідно до табл. 3.26 [ 24 ] *і* приймаю 1 та визначаю η – коефіцієнт використання світлового потоку – 0,61.

Визначаємо світловий потік за формулою:

, де

S – площа – А х В, 81м2;

N – кількість світильників, 32 шт.

*Фл* =500 х 81 х 1,5 х 1,1: 4 х 32 х 0,61 = 66 825 : 75,64 = 883,4 лм

З табл. 3.27 [ 24 ]вибираємо лампуЛДпотужністю20Вт**,** світловий потік якої становить1000 лм.

Різниця між розрахунковим та табличним значенням світлового потоку становить – 1000 – 883,4 = 116.6 лм, в процентному відношенні це складає 116,6:1000х100 = 11,6%, яка є в межах -10% < Фл < +20%

Сумарна потужність усіх ламп становить:

*Рзаг = Рсв N n =* 20 х 32 х 4 **=** 2 480 Вт = 2,4кВт

Висновок***:*** Кількість встановлених світильників у приміщенні достатня для його освітлення.

**7.4 ОХОРОНА ПРАЦІ КОРИСТУВАЧІВ ПК**

При розміщенні робочих столів з ПК слід дотримувати такі відстані між бічними поверхнями ПК - 1,2 м, відстань від тильної поверхні одного ПК до екрана іншого - 2,5 м.

Висота робочої поверхні робочого столу з ПК має регулюватися в межах 680...800 мм, а ширина і глибина - забезпечувати можливість виконання операцій у зоні досяжності моторного поля (рекомендовані розміри: 600...1400 мм, глибина - 800...1000 мм).

Робочий стіл повинен мати простір для ніг заввишки не менше ніж 600 мм, завширшки не менше ніж 500 мм, завглибшки (на рівні колін) не менше ніж 450 мм, на рівні простягнутої ноги - ніж 650 мм.

Робочий стілець має бути підйомно-поворотним, регульованим за висотою, з кутом і нахилу сидіння та спинки і за відстанню від спинки до переднього краю сидіння поверхня сидіння має бути плоскою, передній край - заокругленим. Регулювання за кожним із параметрів має здійснюватися незалежно, легко і надійно фіксуватися. Для зниження статичного напруження м'язів верхніх кінцівок слід використовувати стаціонарні або змінні підлокітники завдовжки не менше ніж 250 мм.

Робоче місце має бути обладнане підставкою для ніг завширшки не менше ніж 300 мм, завглибшки не менше ніж 400 мм, що регулюється за висотою в межах до 150 мм і за кутом нахилу опорної поверхні підставки до 20 град. Підставка повинна мати рифлену поверхню і бортик по передньому краю заввишки 10 мм.

Екран ПК має розташовуватися на оптимальній відстані від очей користувача, що становить 600...700 мм, але не ближче ніж за 600 мм з урахуванням розміру літерно-цифрових знаків і символів.

Клавіатуру слід розташовувати на поверхні столу на відстані 100...300 мм від краю, звернутого до працюючого. У конструкції клавіатури має передбачатися опорний пристрій (виготовлений із матеріалу з високим коефіцієнтом тертя, що перешкоджає мимовольному її зсуву), який дає змогу змінювати кут нахилу поверхні клавіатури у межах 5...15 град.

При організації праці, що пов'язана з використанням ПК, для збереження здоров'я працюючих, запобігання професійним захворюванням і підтримки працездатності слід передбачити внутрішньозмінні регламентовані перерви для відпочинку.

Внутрішньозмінні режими праці і відпочинку мають передбачати додаткові нетривалі перерви в періоди, що передують появі об'єктивних і суб'єктивних ознак втомлення і зниження працездатності. При виконанні протягом дня робіт, що належать до різних видів трудової діяльності, за основну роботу з ПК слід вважати таку, що займає не менше 50 % часу впродовж робочої зміни, мають передбачатися:

- перерви для відпочинку і вживання їжі (обідні перерви);

- перерви для відпочинку і особистих потреб (згідно з трудовими нормами);

- додаткові перерви, що вводяться для окремих професій з урахуванням особливостей трудової діяльності.

Тривалість обідньої перерви визначається чинним законодавством про працю і Правилами внутрішнього трудового розпорядку організації.

Внутрішньозмінні режими праці і відпочинку при роботі з ПК розроблено з урахуванням характеру трудової діяльності, напруженості і важкості праці диференційовано для кожної професії (10-15 хвилин після кожної години роботи).

При 12-годинній робочій зміні регламентовані перерви повинні встановлюватися в перші 8 годин роботи аналогічно перервам при 8-годинній робочій зміні, а протягом останніх 4-х годин роботи, незалежно від характеру трудової діяльності, через кожну годину тривалістю 15 хвилин.

Працюючі з ПК підлягають обов'язковим медичним оглядам: попереднім - при влаштуванні на роботу і періодичним - протягом трудової діяльності. Періодичні медичні огляди мають проводитися раз на два роки комісією в складі терапевта, невропатолога та офтальмолога. До складу комісії, що проводить попередні та періодичні медичні огляди, при необхідності (за наявністю медичних показань) можуть залучатись до оглядів лікарів інших спеціальностей. Жінки, що працюють з ПК, обов'язково оглядаються лікарем акушером-гінекологом один раз на два роки.

ВИСНОВКИ

На основі виконаної роботи можна зробити декілька висновків:

Розроблений програмний продукт для обліку учнів є ефективним та зручним для використання. Він дозволяє зберігати всю необхідну інформацію про учнів, здійснювати її швидкий пошук та оновлення.

У процесі розробки було виявлено деякі недоліки в попередніх версіях програмного продукту, які будуть усунені в нових версіях. Наприклад, була додана можливість зберігати більше інформації про учнів, а також були покращені функції пошуку та сортування.

Для подальшого удосконалення програмного продукту можна розглянути додавання додаткових функцій, таких як можливість відслідковувати академічний прогрес учнів, створення різних звітів для аналізу даних тощо.

Також можна розглянути можливість інтеграції програмного продукту з іншими системами, що використовуються у навчальних закладах, для забезпечення більшої автоматизації процесів.

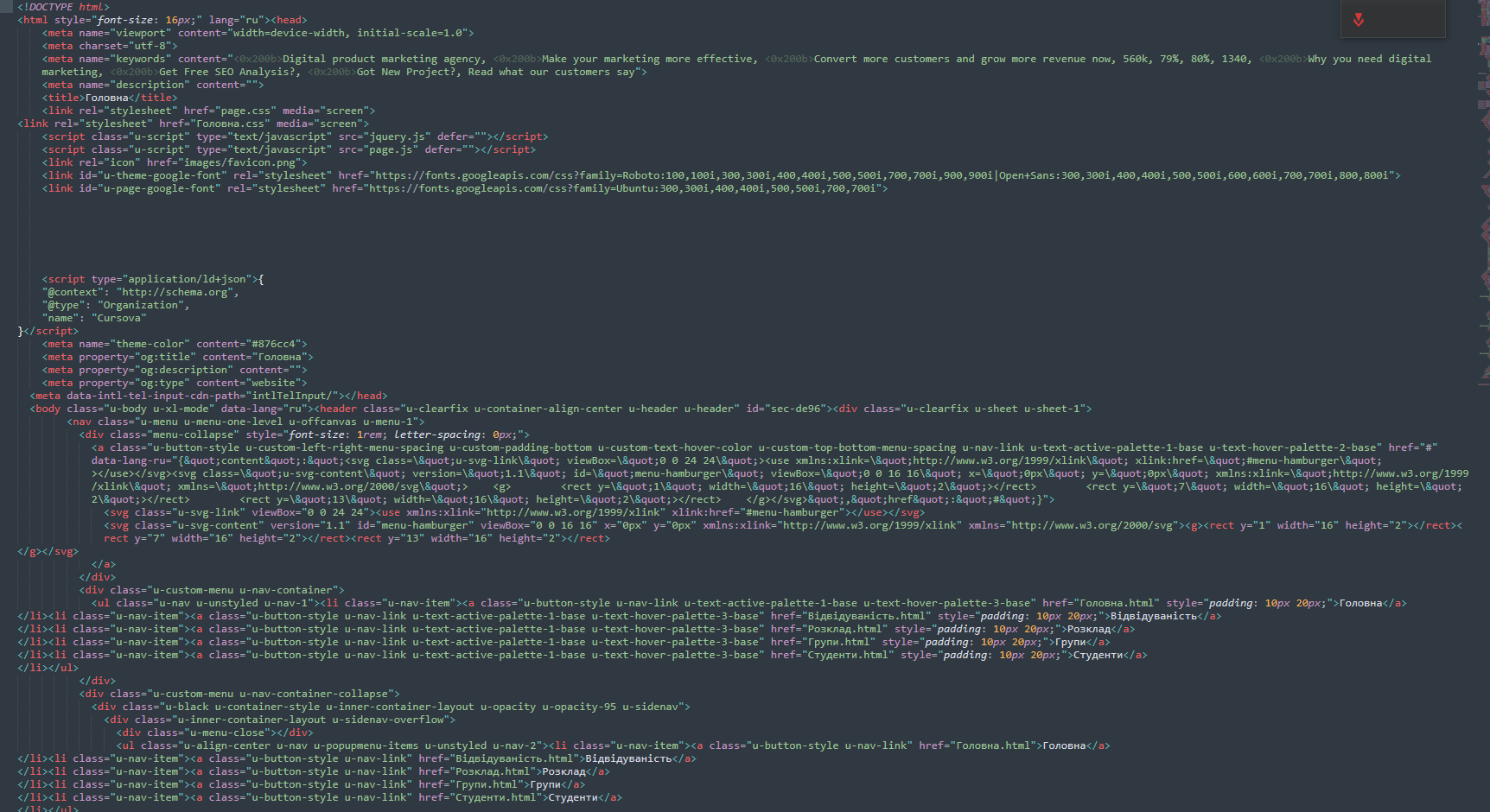
Отже, можна стверджувати, що розроблений програмний продукт є ефективним та зручним для використання, а додавання додаткових функцій може покращити його функціональність та забезпечити більшу автоматизацію процесів у навчальних закладах.

# **ПЕРЕЛІК ПОСИЛАНЬ НА ДЖЕРЕЛА**

1. https://www.youtube.com/watch?v=-skvJrF23Kc
2. https://telegra.ph/WebView-AdStudy-05-22
3. https://intelligencloud.com/ua/student-accounting-program
4. Окулов З. М. Програмування алгоритмах. - М:БІНОМ. 2002
5. The Web Application Security Consortium [Електронний ресурс]: Режим
6. Code Project. Спільнота розробки програмного забезпечення. [Електронний ресурс]. – Режим доступу: <http://codeproject.com>
7. Професійна розробка сайтів на Drupal 7; Дніпро, 2013. – 688 c
8. WEB-дизайн як навчальна дисципліна, 2021. - KURSOVIKS - URL: <https://ua.kursoviks.com.ua/kompyuterni/web-dizayn>
9. Практикум з охорони праці 2007 О.К.Шукаібов

**ДОДАТОК А**

Додаток А Приклад коду інтерфейсу головної сторінки



**ДОДАТОК Б**

Додаток Б Приклад коду стилів головної сторінки

