КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра прикладних інформаційних систем

122 Комп'ютерні науки
«Прикладне програмування»

Кваліфікаційна робота бакалавра

на тему: «Інформаційна система електронного документообігу загальноосвітньої школи»

Виконав студент 4 курсу групи ПП−41

_____
(Підпис)
Павелко Т. М.
(прізвище, ім'я, по батькові)

Керівник к.т.н. Міронова В.Л.
(прізвище, ім'я, по батькові)

_____
(Резолюція «До захисту»)

Попередній захист:

_____
(Висновок: "До захисту в екзаменаційній комісії ")

Завідувач кафедри Плескач В.Л. _____ _____
(Прізвище, ініціали) (Дата) (Підпис)

Київ – 2021

Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій

Кафедра прикладних інформаційних систем

Назва теми: «Інформаційна система електронного документообігу загальноосвітньої школи»

Освітня програма: Прикладне програмування

Спеціальність: Комп'ютерні науки

ПІБ	Підпис
Павелко Тарас Миколайович	

Назва роботи українською та англійською мовами

Інформаційна система електронного документообігу загальноосвітньої школи School electronic workflow information system

Мета бакалаврської кваліфікаційної роботи, завдання

Мета роботи: Організація процесу ефективного збереження та поширення інформації з розмежуванням доступу в загальноосвітній школі за допомогою розробки інформаційної системи з можливістю спряження її з пристроями Інтернету речей.

План роботи:

1. Вивчити підходи до розроблення сучасних систем електронного документообігу.
2. Розробити структуру інформаційної системи та підібрати стек технологій, необхідний для її реалізації.
3. Розробити спеціалізовані пристрої Інтернету речей, що, будучи спряженими з інформаційною системою, зможуть автоматизувати технологічні процеси школи.
4. Реалізувати та впровадити інформаційну систему електронного документообігу в загальноосвітній школі.

Міронова Вікторія Леонідівна, доцент, кандидат технічних наук: _____

ВІДОМІСТЬ ДИПЛОМНОЇ РОБОТИ

Складові частини дипломної роботи	Обсяг, арк.
Титульний аркуш	1 ст.
Завдання до дипломної роботи	1 ст.
Відомість дипломної роботи	1 ст.
Календарний план	1 ст.
Реферат (Анотація)	1 ст.
Анотація (іноземною мовою-англійською)	1 ст.
Зміст	2 ст.
Вступ	2 ст.
1. Теоретичні основи побудови інформаційної системи електронного документообігу загальноосвітньої школи	5 ст.
2. Розробка архітектури та алгоритмів інформаційної системи електронного документообігу	24 ст.
3. Розробка архітектури та алгоритмів роботи, спряжених з інформаційною системою, пристроїв Інтернету речей	21 ст.
4. Опис роботи системи та заходи щодо забезпечення безпеки використання	7 ст.
Висновки	1 ст.
Перелік посилань	3 ст.

				ДП XXXX 00.000.00		
	ПІБ	Підп.	Дата			
Розробн.				Відомість дипломної роботи	Лист	Листів
Керівн.						
Н/контр.	Макаренко С.А.					
Зав.каф.	Плескач В.Л.					

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Номер	Назва етапів кваліфікаційної роботи	Термін виконання етапів кваліфікаційної роботи	Відмітка про виконання
1.	Вибір теми та наукового керівника кваліфікаційної роботи бакалавра	26.10.2020	Виконано
2.	Видача завдання кваліфікаційної роботи бакалавра	23.11.2020	Виконано
3.	Настановча групова співбесіда з питань кваліфікаційної роботи бакалавра	01.12.2020	Виконано
4.	Затвердження плану кваліфікаційної роботи бакалавра	18.02.2021	Виконано
5.	Підбір та вивчення літературних та інших джерел з теми дослідження	25.02.2021	Виконано
6.	Підготовка і подання науковому керівнику першого варіанту I розділу роботи	05.03.2021	Виконано
7.	Підготовка і подання науковому керівнику першого варіанту II розділу роботи	09.04.2021	Виконано
8.	Підготовка і подання науковому керівнику першого варіанту III розділу роботи	07.05.2021	Виконано
9.	Подання роботи у першому варіанті	11.05.2021	Виконано
10.	Оформлення пояснювальної записки кваліфікаційної роботи бакалавра	12.05.2021	Виконано
11.	Подання кваліфікаційної роботи бакалавра на попередній захист	24.05.2021	Виконано
12.	Врахування зауважень керівника і подання роботи в остаточному варіанті (з відповідним висновком про допуск) на кафедру	28.05.2021	Виконано
13.	Затвердження роботи в цілому (підготовка письмового відгуку керівника, письмова рецензія на бакалаврської роботу)	11.06.2021	
14.	Захист кваліфікаційної роботи бакалавра	22.06.2021	

Здобувач вищої освіти _____
(підпис)

Керівник _____
(підпис)

РЕФЕРАТ

Проєкт кваліфікаційної роботи містить 71 сторінок, 26 рисунків, 1 додаток та 40 літературних джерел.

Метою кваліфікаційної роботи є організація процесу ефективного збереження та поширення інформації з розмежуванням доступу в загальноосвітній школі за допомогою розробки інформаційної системи з можливістю спряження її з пристроями Інтернету речей.

Кваліфікаційна робота виконана студентом – Павелко Т. М.

Назва роботи – Інформаційна система електронного документообігу загальноосвітньої школи.

Шифр та назва спеціальності – 122 «Комп’ютерні науки».

Освітня програма – Прикладне програмування.

Актуальність обраної теми полягає в потребі закладів загальної середньої освіти в програмній системі, що надасть інструментарій для ведення необхідних документів у режимі онлайн, та автоматизації рутинних процесів.

Об’єктом дослідження кваліфікаційної роботи є інформаційна система електронного документообігу для загальноосвітньої школи.

Предметом дослідження кваліфікаційної роботи є програмні, технічні та організаційні підходи до реалізації інформаційної системи електронного документообігу.

Дана кваліфікаційна робота включає повноцінну реалізацію інформаційної системи електронного документообігу та спряжених з нею пристроїв Інтернету речей з послідуючим її впровадженням в одну із загальноосвітніх шкіл.

Ключові слова: інформаційна система, електронний документообіг, база даних, Інтернет речей.

ANNOTATION

Qualification bachelor’s project work contains 71 pages, 26 figures, 1 appendix and 40 literary sources.

The purpose of the qualification work is to organize the process of effective storage and dissemination of information with the delimitation of access in secondary school through the development of an information system with the ability to connect it to the Internet of Things.

Qualification work performed by a student – Taras Pavelko.

Title of work - School electronic workflow information system.

Code and name of the specialty - 122 "Computer Science".

Educational program - Applied programming.

The relevance of the chosen topic lies in the necessity of general secondary education institutions in the software system, which will provide tools for maintaining the necessary documents online, and automation of routine processes.

The object of study of the qualification work is the information system of electronic document management for secondary school.

The subjects of research of qualification work are software, technical and organizational approaches to the implementation of the information system of electronic document management.

This qualification work includes the full implementation of the information system of electronic document management and related Internet of Things devices with its subsequent implementation in one of the secondary schools.

Keywords: information system, electronic document management, database, Internet of Things.

ЗМІСТ

ВСТУП

Ведення звичайної паперової документації в закладах отримання загальної середньої освіти в епоху глобальної діджиталізації та повсюдної автоматизації виглядає архаїзмом з попереднього століття. Саме тому є досить актуальною задача створити інформаційну систему електронного документообігу для загальноосвітніх шкіл.

Актуальність теми цієї кваліфікаційної роботи зумовлено потребою закладів загальної середньої освіти в програмній системі, що надасть інструментарій для ведення більшості необхідних документів у режимі реального часу, для моніторингу інформації різними типами користувачів з розмежуванням доступу, та для автоматизації рутинних процесів.

Метою кваліфікаційної роботи є організація процесу ефективного збереження та поширення інформації з розмежуванням доступу в загальноосвітній школі за допомогою розробки інформаційної системи з можливістю спряження її з пристроями Інтернету речей, для цього необхідно вирішити такі **завдання**:

- Дослідити теоретичні основи побудови інформаційної системи електронного документообігу;
- Проаналізувати програмні та технічні рішення, а також різні види забезпечення інформаційної системи;
- Спроектувати та реалізувати інформаційну систему електронного документообігу;
- Спроектувати, розробити та налагодити спряжені з інформаційною системою пристрої Інтернету речей;
- Налаштувати та впровадити систему для конкретного користувача.

Об’єктом дослідження кваліфікаційної роботи є інформаційна система електронного документообігу. Її функціональні рішення, принципи та підходи до вирішення проблем, пов’язаних зі збиранням, обробленням та поширенням інформації.

Предметом дослідження кваліфікаційної роботи є програмні, технічні та організаційні підходи до реалізації інформаційної системи електронного документообігу.

Практичне значення полягає в отриманні в результаті дослідження й розробки повноцінну, не маючу аналогів, інформаційну систему електронного документообігу для загальноосвітньої школи, яка має потужну сучасну архітектуру, що дозволяє користувачу проводити взаємодію не лише з використанням веб-інтерфейсу. Важливим практичним значенням є широкі можливості системи в галузі автоматизації рутинних процесів, як на програмному рівні, так і на апаратно-програмному за допомогою також спеціально розроблених та спряжених пристроїв Інтернету речей.

У процесі виконання кваліфікаційної роботи було використано: документація класів та методів .NET Framework, JavaScript, jQuery,програмне забезпеченняCorelDraw, Microsoft Visual Studio, Visual Studio Code, Microsoft SQL Server, SQL Server Management Studio, Arduino IDE, Google Chrome, Saleae Logic, Rigol Ultra Scope Software, міжнародні та національні стандарти з побудови програмних систем, а також цифрові мультиметр та осцилограф, лабораторний блок живлення та логічний аналізатор.

Апробація результатів дослідження та розробки проводилася на базі загальноосвітньої школи I-III ступенів Рівненської міської ради №18, де інформаційна система була впроваджена в тестовому режимі.

Публікація тезисів, пов’язаних з дослідженням галузі та розробкою інформаційної системи електронного документообігу відбулася на XXV міжнародній науково-практичній конференції «Інформаційні технології в економіці, менеджменті та бізнесі. Проблеми науки практики та освіти».

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ПОБУДОВИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ ДЛЯ ЗАГАЛЬНООСВІТНЬОЇ ШКОЛИ

1.1 Огляд і аналіз інформаційних систем, принципів і засобів створення систем електронного документообігу

Створення власної інформаційної системи електронного документообігу є актуальною проблемою багатьох сучасних закладів загальної середньої освіти. Адже подібні системи вже давно перейшли із розряду перспективних нововведень елітних шкіл у розряд крайнє необхідних інструментів для всіх закладів освіти. Така інформаційна система дозволяє з легкістю акумулювати інформацію, з можливістю її подальшого модифікування, поширювати цю інформацію в колі зацікавлених в ній осіб, зберігати інформацію нескінченно довго, а також автоматично виконувати деякі рутинні операції над інформацією, що компенсує час, затрачений на ввід її в систему.

Зараз існує досить багато сервісів, що надають подібний функціонал, але всі вони або коштують дуже дорого, або реалізовані не найкращим чином: мають обмежений набір функцій або перевантажений інтерфейс, що не сприяє полегшенню роботи людей, які їх використовують. Для проведення аналізу існуючих систем були розглянуті:

- e-schools.info
- nz.ua
- atoms.com.ua

Функціонал цих додатків було вивчено та проаналізовано. Аналіз надав розуміння принципів побудови таких систем.

1.2 Аналіз архітектури існуючих систем електронного документообігу

Переважна більшість розглянутих при дослідженні систем були виключно веб-сервісами, що не дозволяло їм вийти за межі взаємодії з

користувачем тільки через веб-сайт. Також, іноді, зустрічаються шкільні системи електронного документообігу, що являють собою мобільний за стосунок під одну конкретну платформу. Спроба нарощення нестандартного функціоналу чи поширення на нову платформу завжди призводить до майже повного пере створення системи з нуля.

Вирішенням проблеми одноплатформеності може стати створення єдиного програмного інтерфейсу застосунку (API), що виконуватиме роль спеціалізованого програмного забезпечення, яке б в своєму складі мало реалізацію всіх необхідних класів, функцій, структур, констант та процедур з якими могли взаємодіяти застосунки, побудовані на різних платформах для проведення складних обрахунків та операцій з даними.

Так як інформаційна система електронного документообігу загальноосвітньої школи перш за все передбачає віддалений доступ з мережі Інтернет, то і єдиний програмний інтерфейс повинен бути розміщений на віддаленому сервері й спроектований по технології Web API для обробки HTTP запитів. Варто чітко зрозуміти різницю між технологіями Web API та MVC, адже на перший погляд може здатися, що вони повністю повторюють одна одну. Web API призначений для реалізації REST сервісів, що охоплює повний спектр методів запитів (GET, POST, PUT, та DELETE). Результатом, що буде повернено сервером API в переважній більшості буде структурований JSON, а також є можливість пакування даних у XML. MVC в свою чергу повертає готове представлення у вигляді HTML документу [7].

Отже, для реалізації системи електронного документообігу загальноосвітньої школи варто обрати архітектуру, що базується на технології Web API. Це може призвести до витрати більшого часу на етапі проектування та розробки перших прототипів, але Web API – це єдиний правильний вибір при системи, що може взаємодіяти не тільки за допомогою веб-інтерфейсу, але, наприклад, мобільних застосунків різноманітних платформ чи спеціалізованих пристроїв Інтернету речей.

1.2 Обґрунтування мети побудови інформаційної системи електронного документообігу для загальноосвітніх шкіл

Із проведеного аналізу існуючих систем електронного документообігу можна зробити наступні висновки:

- більшість безкоштовних систем надають досить обмежений функціонал;
- системи, що надають гідний інструментарій, є або важкими в налаштуванні й незручними у використанні, або доступними тільки після оформлення платної підписки;
- на базі жодної з них не можна побудувати екосистему, що б включала функції, що виходять за рамки поняття інформаційної системи (взаємодія з різними графічними оболонками та пристроями Інтернету речей).

Реалізація та впровадження інформаційної системи для загальноосвітньої школи з усім необхідним функціоналом та можливістю легкого його нарощення в майбутньому – є найголовнішою метою цієї кваліфікаційної роботи. Варто відзначити, що одною із найбільш вагомих переваг системи є те, що систему можливо адаптувати під кожну конкретну школу окремо, враховуючи всі вимоги та побажання адміністрації. Саме це робить дану систему такою доступною для більшості шкіл, які дуже хочуть модернізувати свою архаїчну паперову документацію в цифровий рівень, але водночас не бажають підлаштовуватися під незручності та проблеми, які виникають, якщо впроваджувати абсолютно неадаптовану інформаційну систему.

1.3 Постановка задачі на розробку інформаційної системи електронного документообігу для загальноосвітніх шкіл

Завдання кваліфікаційної роботи – створити інформаційну систему для загальноосвітньої школи, яка матиме повний набір функцій та методів, характерних для подібних систем (ввід, зберігання, поширення інформації), дозволить автоматизувати рутинні процеси (підрахування та виставлення тематичних, семестрових та річних оцінок) а також зможе працювати з необхідними IoT пристроями. Для виконання поставленої мети необхідно послідовно виконати наступні задачі:

- провести аналіз існуючих інформаційних систем з подібним призначенням;
- визначити сильні та слабкі сторони уже існуючих систем;
- створити макет зручного та зрозумілого графічного інтерфейсу для всіх типів користувачів;
- зробити адаптивну верстку сайту по макету, для нормального його відображення на пристроях різних типів;
- спроектувати базу даних для системи;
- розробити та реалізувати усі необхідні алгоритми для всіх функцій системи;
- забезпечити коректну роботу зв’язки frontend-backend;
- організувати достойний рівень захищеності системи;
- розробити та реалізувати спряжені з системою пристрої IoT;
- виконати повноцінне всеосяжне тестування системи вцілому та кожного її модуля окремо;
- зробити систему глобально доступною, розмістивши її на хостинг;
- написати документацію на систему;
- адаптувати систему під конкретну школу;
- ввести систему в користування.

Висновки до розділу

У цьому розділі було проведено аналіз ситуації, що пов’язана з документообігом, у загальноосвітніх школах. Виявлено велику кількість недоліків звичайного методу ведення паперової документації, що дозволило обґрунтувати мету створення інформаційної системи електронного документообігу загальноосвітньої школи. У процесі огляду й порівняння уже існуючих систем було отримано розуміння набору основних функцій, якими подібні системи повинні володіти, а також було сформульовано ідеї, що будуть унікальними.

Було проаналізовано базові архітектурні рішення для побудови інформаційної системи, що дозволять реалізувати всі задумані ідеї.

Також у цьому розділі було сформульовано постановку задачі на розробку інформаційної системи з чітким планом-послідовністю дій.

РОЗДІЛ 2. РОЗРОБКА АРХІТЕКТУРИ ТА АЛГОРИТМІВ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ

2.1 Інформаційне забезпечення системи електронного документообігу

Для виконання вимог технологічного завдання при реалізації безпосередньо інформаційної системи було використано наступні технології:

- HTML;
- HTTP;
- CSS;
- JavaScript;
- jQuery;
- AJAX;
- ASP.NET Core Web API;
- C#;
- ADO.NET;
- Microsoft SQL Server;
- SQL.

HTML (Hyper Text Markup Language) – це мова розмітки, яка повідомляє веб-переглядачу, як відображати веб-сторінку [8]. Цю мову можна назвати кросплатформенною, адже більшість популярних операційних систем для всіляких пристроїв (ноутбуків, планшетів, смартфонів, годинників) мають у своєму арсеналі веб-браузер, який працює з HTML документами. Для інтерпретації вмісту сторінки браузером використовуються теги, які на екран не виводяться, але можуть містити корисну інформацію або бути складовою частиною складної ієрархії елементів всієї сторінки. Кожен із тегів може мати свої атрибути. Вони використовуються для визначення характеристик HTML-елементів і вписуються безпосередньо у відкритий тег у вигляді ключ = значення.

Для передачі документа з веб-сервера до браузера використовується **HTTP** (Hyper Text Transfer Protocol)–протокол прикладного рівня передачі даних [9]. Основою протоколу є технологія «клієнт-сервер». Основними та найчастіше використовуваними методами HTTP запитів вважаються GETі POST. Перший найчастіше використовується для отримання даних з вказаного ресурсу, коли другий – для передачі користувацьких даних на вказаний ресурс. Найбільш вагомими відмінностями між методами можна вважати те, що GET, на відміну від POST, є ідемпотентним, іншими словами повторна дія над будь-яким об’єктом не буде змінювати результату, а також параметри, що передаються в запиті від клієнта є частиною URLцільового ресурсу.

Як було сказано раніше, HTML являє собою лише «каркас» сторінки, саме тому для адекватного відображення його ще необхідно стилізувати. Для виконання цієї задачі використовується **CSS** (Cascading Style Sheets) – формальна мова опису зовнішнього вигляду документа, написаного за допомогою мови розмітки [10]. Завдяки CSSми можемо змінювати різноманітні параметри елементів: розмір, колір, форму, тип тощо. А також можна створювати анімації для елементів, шляхом зміни їх стилів.

У сьогоднішніх реаліях каскадна (блочна) верстка веб-сторінок прийшла на заміну табличній. Одною із найголовніших переваг такої верстки є можливість розділення змісту (даних) і його візуального відображення. Це знайшло своє застосування при забезпеченні адаптивності сайту. Адже різні типи пристроїв мають екрани різних розмірів та відношення сторін, і під кожен із них потрібно перебудувати сторінку так, щоб інтерфейс залишався зручним та зрозумілим, не говорячи про принтери, на друк яким варто відправляти лише корисну інформацію, прибираючи елементи управління.

CSS код може розміщуватися в самому HTML документі в тегу <style> або в однойменному атрибуті, якщо мова йде про конкретний окремо взятий елемент, а також цей код можна повноцінно відокремити в файл з

розширенням .css, посилання на який необхідно буде розмістити в тег <head> за допомогою <linkrel=”stylesheet”>.

Для обробки усіх необхідних подій та запитів на стороні клієнта використовується **JavaScript** – інтерпретована прототипно-орієнтована мова сценаріїв з динамічною типізацією, яка найбільш часто використовується браузерами для надання інтерактивності веб-сторінкам [13]. Не дивлячись на подібність синтаксису з мовою C, JavaScriptсильно від неї відрізняється наявністю анонімних функцій, об’єктів з можливістю інтроспекції, системи автоматичного приведення типів, системи автоматичного збору сміття тощо.

Якщо розглядати JavaScript тільки з точки зору виконання браузером, то його структура складається із трьох елементів: ядро, об’єктна модель браузера (Browser Object Model) та об’єктна модель документа(Document Object Model). BOM є зв’язуючим елементом між ядром і DOM. Основним завданням об’єктної моделі браузера є управління вікнами браузера та забезпечення їх коректної взаємодії, а також управління окремими фреймами, системними діалогами, інформацією про браузер тощо. DOM – інтерфейс програмування застосунків на HTML, згідно з яким веб-сторінка має бути представлена у вигляді дерева об’єктів, зі своїми унікальними властивостями, якими можна маніпулювати, а саме: отримання, додавання, модифікування, видалення вузлів, а також зв’язків між ними.

JavaScript можна під’єднати до головного документа декількома способами: в тегу <head>використати тег <script>з атрибутом src, в якому вказати посилання на файл з розширенням .js, який в свою чергу може розміщуватися локально або на віддаленому сервері, наступним способом є написання JavaScript коду безпосередньо після відкриття тегу <script> у будь-якому із місць всередині тегу <body>.

Для спрощення роботи з DOM використовується **jQuery**– набір функцій JavaScript, основними завданнями яких є взаємодія JavaScript і HTML [14]. За допомогою цієї бібліотеки стає легшим процес доступу до елементів DOM та маніпуляції над ними.

Одною із найголовніших переваг jQuery є більш зручний інструментарій для роботи з **AJAX** (Asynchronous JavaScript and XML) – один із підходів до побудови інтерактивних користувацьких інтерфейсів веб-застосунків, основним принципом якого є обмін даними між клієнтом і сервером у фоновому режимі [15]. Завдяки цьому у веб-сторінки стають відсутніми перезавантаження при оновленні даних, що в свою чергу сприяє підвищенню швидкості й зручності роботи застосунків вцілому.

AJAX відправляє запити на сервер, який реалізований на **ASP.NET Core 5.0** – найновішій технології від Microsoft, що призначена для створення кросплатформених веб-застосунків будь-яких розмірів з найрізноманітнішим функціоналом. ASP .NET Core, на відміну від застарілого ASP .NET Framework, є проєктом із повністю відкритим кодом, який розміщений на офіційній сторінці у GitHub. Як зазначалося вище, однією із найбільших переваг нової технології, є підтримка кросплатформеності, яка дає можливість розгортання свого проекту на всіх сучасних популярних операційних системах, таких як Windows, MacOS, Linux. Ця можливість відкриває нові горизонти при виборі хостингу для створеного проєкту.

Сервер інформаційної системи, що розроблюється, реалізує шаблон **Web API**, що, на відміну від Model-View-Controller, складається всього з двох найголовніших компонентів: моделей і контролерів. Цей шаблон є, на мою думку, найбільш вдалим для розробки системи, що виходить за межі звичайного визначення інформаційної системи, що підтримує не одну тільки графічну оболонку доступу у вигляді веб-сайту, але й надає можливість викликати серверні функції з мобільних застосунків і розумних пристроїв Інтернету речей на базі мікроконтролерів. Контролер приймає запити, обробляє дані, що надійшли від користувача або пристрою IoT та взаємодіє з моделлю шляхом використання останньої як специфічної структури даних для отримання користувацької інформації, отримання даних з бази даних та для формування пакету результуючих даних у відповідь користувачу.

Іншими словами, модель являє собою клас, що описує логіку організації даних в застосунку (рис. 2.1).

Рисунок 2.1 – Схема взаємодії компонентів у шаблоні Web API

Шаблон Web API був спеціально створений компанією Microsoft для надання розробникам можливості створювати свої проєкти на REST (Representation State Transfer) архітектурі.

Найбільш широко використовуваними методами звернення до сервера у стилі REST є:

- GET – метод для запиту даних із серверу, зазвичай, у вигляді масиву об’єктів JSON. Як правило, використовується без параметрів у блоці body;
- GET{id}– підвид GET методу з передачею ідентифікатора (…Controller/id);
- POST–метод для створення нових даних на сервері. Майже завжди отримує дані в JSON об’єкті у блоці body;

- PUT–метод для оновлення уже існуючих даних на сервері. Майже завжди отримує дані в JSON об’єкті у блоці body;
- DELETE–метод для видалення даних на сервері. Майже завжди отримує дані в JSON об’єкті у блоці body;

При обробці запитів Web API опирається на систему маршрутизації, яка співставляє всі вхідні запити з визначеними в системі маршрутами, які вказують, який контролер й метод повинні обробляти даний запит. Вбудований маршрут по замовчуванню передбачає наступну структуру: контролер-дія-параметр.

У рамках ASP .NET Core Web API для програмування контролерів і не тільки буде використовуватися найбільш сучасна та прогресивна із доступних у .NET об’єктно-орієнтована строго типізована мова програмування **C#**. Мова має Сподібний синтаксис, але все ж таки більше схожа на C++ iJava. Також C# завдяки .NET володіє величезним стеком технологій, які можна використовувати при розробці різного роду застосунків, а саме: для роботи з базами даних ADO.NET та Entity Framework, для побудови графічних інтерфейсів WPF і UWP, а для більш простих – Windows Forms, для створення мобільних застосунків Xamarin, а для створення веб-сайтів – ASP.NET [19].

У даній системі для організації зв’язку з базою даних буде використовуватися технологія **ADO.NET**, яка надає набір класів, за допомогою яких можна буде відправляти запити до БД, встановлювати підключення, отримувати результати запитів та проводити ряд інших операцій [20].

ADO.NET використовує один і той же набір об’єктів для роботи з різними джерелами даних. Для забезпечення такої кросплатформної роботи в ADO.NET застосовуються різноманітні провайдери даних. У даному випадку використовується провайдер для **MS SQL Server** – одної із найбільш популярних, надійних, безпечних, продуктивних і в той же час простих систем керування базами даних (СКБД).

Центральним аспектом будь-якої СКБД є база даних, яка являє собою сховище даних, організоване визначеним способом. Для організації баз даних MS SQL Server використовує реляційну модель, яка передбачає зберігання даних у вигляді таблиць, кожна з яких складається із рядків і стовпців. Кожен рядок зберігає окремий об’єкт, а кожен стовпець один із атрибутів цього об’єкту [21].

Для взаємодії з базою даних використовується мова **SQL** (Structured Query Language). Варто уточнити, що в даному випадку, коли вживається абревіатура SQL, необхідно розуміти одну із їх різновидів, а саме T-SQL. Клієнтська сторона у визначений момент формує запит на мові SQL, використовуючи ADO.NET відправляє його на сервер, де СКБД належним чином його інтерпретує, виконує та відправляє клієнту результат виконання.

2.1.1 Структура сторінок проектованої системи

Для виконання всіх завдань кваліфікаційної роботи та для забезпечення зручності й ефективності користування системою було створено ряд наступних сторінок сайту:

- Головна сторінка авторизації;
- Сторінка щоденника учнів/батьків;
- Сторінка журналу успішності учнів/батьків;
- Сторінка розкладу занять учителя/адміністратора;
- Сторінка журналу класу вчителя/адміністратора.

Після успішного введення імені користувача та пароля на головній сторінці, оператора буде авторизовано, автоматично визначивши його ранг у системі, та йому буде наданий доступ до сторінок, що передбачені його статусом.

Якщо в системі був успішно авторизований користувач зі статусом учня чи батька, то йому відкривається доступ до сторінки електронного щоденника та сторінки його особистого журналу успішності.

Але в іншому випадку, якщо в системі був успішно авторизований користувач зі статусом вчителя чи адміністратора, то йому відкривається доступ до сторінки розкладу занять та сторінки журналу, того чи іншого класу, в якому конкретний вчитель викладає.

Найкраще структуру сторінок та можливості переходів між ними ілюструє діаграма активностей, що зображена на рисунку 2.2.

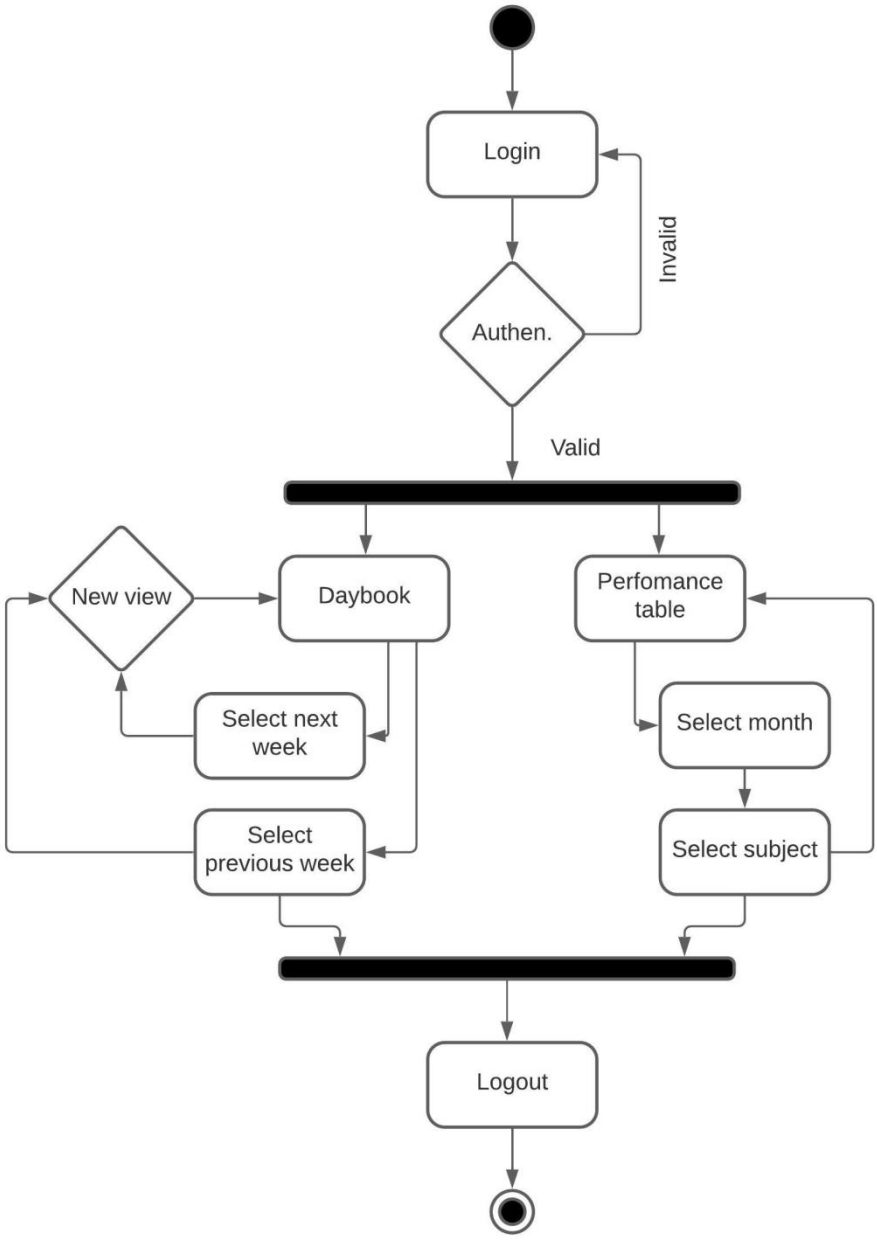


Рисунок 2.2 – Діаграма активностей інформаційної системи електронного документообігу

2.1.2 Функціональні можливості інформаційної системи електронного документообігу

Задля забезпечення виконання усіх поставлених завдань та вимог до інформаційної системи електронного документообігу, а також організації простого, зручного та зрозумілого інтерфейсу взаємодії користувача й системи було вирішено реалізувати наступний функціонал:

- Авторизація користувача – після введення імені користувача та пароля проводяться процеси ідентифікації та автентифікації, успішне завершення яких гарантує надання користувачу доступу до сторінок, що дозволені йому за статусом в системі. Якщо ж дані авторизації введено неправильно, то система надасть відповідне повідомлення;
- Перегляд електронного щоденника – функція перегляд учнем/батьком таблиць вигляду номер уроку – предмет – домашнє завдання – оцінка, згрупованих по днях тижня;
- Навігація по тижнях електронного щоденника – функція швидкої навігації «вправо», «вліво» дозволяє переглядати сторінки щоденника найближчих попередніх та наступних тижнів. Часткова чи повна відсутність кнопок навігації означає, що в системі не існує інформації про відповідні тижні;
- Перегляд журналу успішності – функція перегляду учнем/батьком оцінок по кожному із уроків конкретного предмету та місяця;
- Навігація по місяцях журналу успішності – функція швидкої навігації «вправо», «вліво» дозволяє переглядати сторінки щоденника найближчих попередніх та наступних місяців;
- Вибір місяця й предмету для журналу успішності – функція забезпечує обрання з випадаючих списків необхідного місяця та предмету (доступного саме для цього місяця) для виводу необхідної сторінки журналу успішності;

- Перегляд розкладу уроків – функція перегляд вчителем чи адміністратором таблиць вигляду номер уроку – предмет – клас проведення – задане домашнє завдання, згрупованих по днях тижня;
- Перегляд журналу класу – функція перегляду вчителем чи адміністрацією всіх оцінок усіх учнів конкретного класу, предмету та місяця;
- Вибір місяця, предмету та класу для журналу класу/підгрупи – функція забезпечує обрання з випадаючих списків необхідного місяця, предмету (доступного саме для цього місяця) та класу (доступного тільки для цієї комбінації місяць-предмет) для виводу необхідної сторінки журналу класу;
- Виставлення поточної оцінки – функція дозволяє поставити конкретному учню в обрану дату поточну оцінку від 1 до 12 або «н» у разі відсутності учня на уроці.
- Створення стовпця особливих оцінок – функція організовує додатковий стовпець для виставлення оцінок, які передбачені програмою але не є поточними, контрольними, тематичними, семестровими чи річними.
- Встановлення статусу контрольної роботи для стовпця поточних оцінок – функція забезпечує можливість вчителю встановити уроку, конкретної дати, статус контрольної роботи.
- Автоматизоване виставлення тематичних (семестрових, річних) оцінок – функція доступна до виклику вчителем чи адміністрацією, яка в автоматичному режимі визначає та виставляє тематичні (семестрові, річні) оцінки всім учням на поточній сторінці журналу класу.

Усі вище описані функціональні можливості, а також додаткові можливості адміністратора, що здійснюються за допомогою безпосередньої взаємодії з базою даних ілюструє діаграма варіантів використання (рис. 2.3).

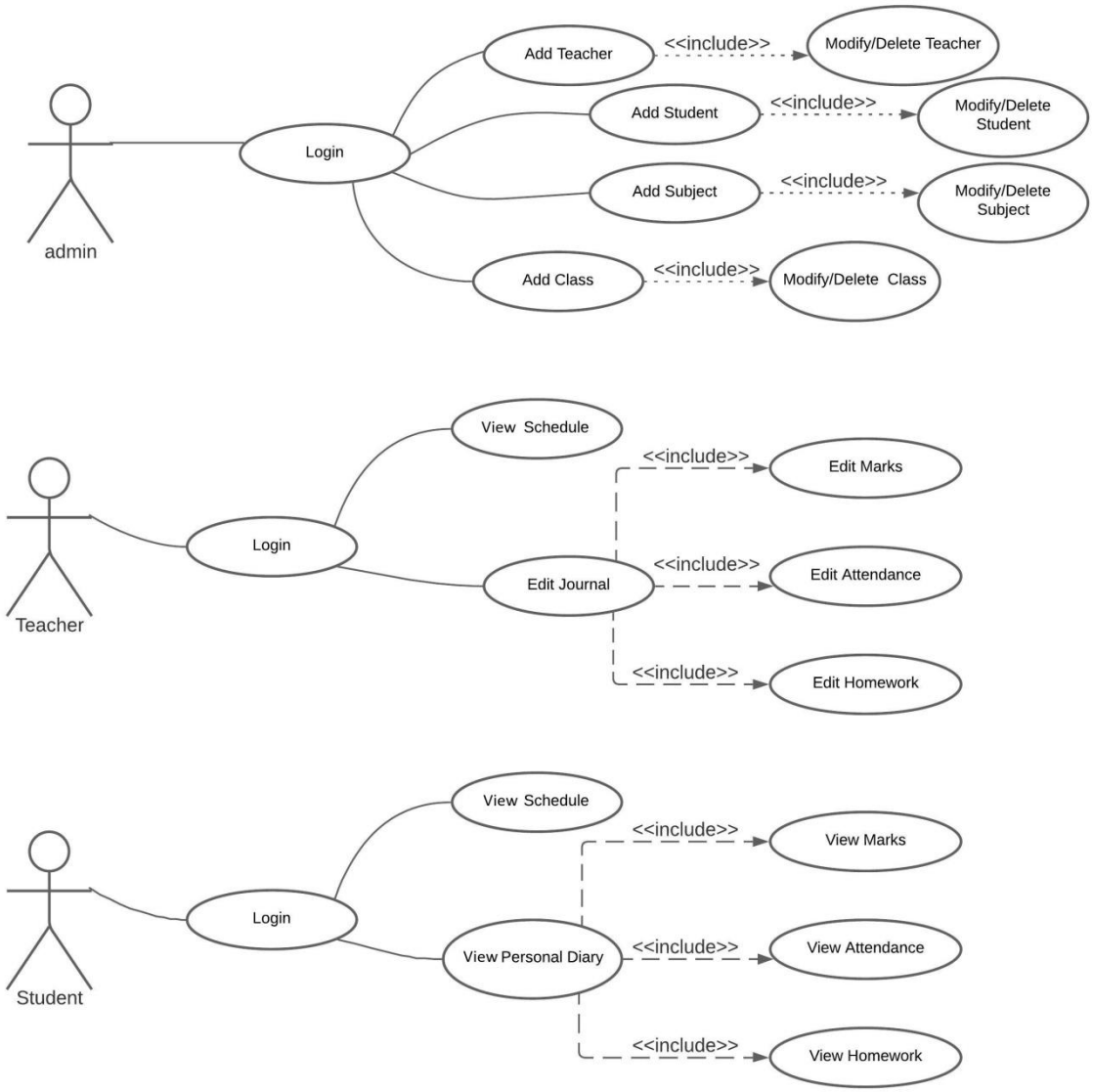


Рисунок 2.3 – Діаграма варіантів використання інформаційної системи електронного документообігу

2.1.3 Структура бази даних

База даних – це сукупність даних, що зберігаються відповідно до схеми даних, операції над якими (зберігання, модифікування та обробка) виконують за правилами засобів моделювання даних. Бази даних є одним із найголовніших компонентів архітектури динамічних сайтів зі великими об’ємами даних.

Задля організації повного функціоналу інформаційної системи, що б відповідав усім поставленим вимогам, було створено наступні таблиці в БД:

- Користувачі;
- Авторизації;
- Люди;
- Адреси;
- Працівники;
- Учні;
- Батьки;
- Батьки Учнів;
- Класи;
- Учні Класів;
- Предмети;
- Предмети Класів;
- Вчителі Предметів;
- Уроки Дня;
- Оцінки;
- Підгрупи Учнів;
- Предмети Підгруп Учнів;
- Відвідувачі;
- Дзвонарі;
- Дзвінки.

Структура бази даних спроектована таким чином, щоб вона відповідала третій нормальній формі баз даних. Схему БД наведено на рисунку 2.4.

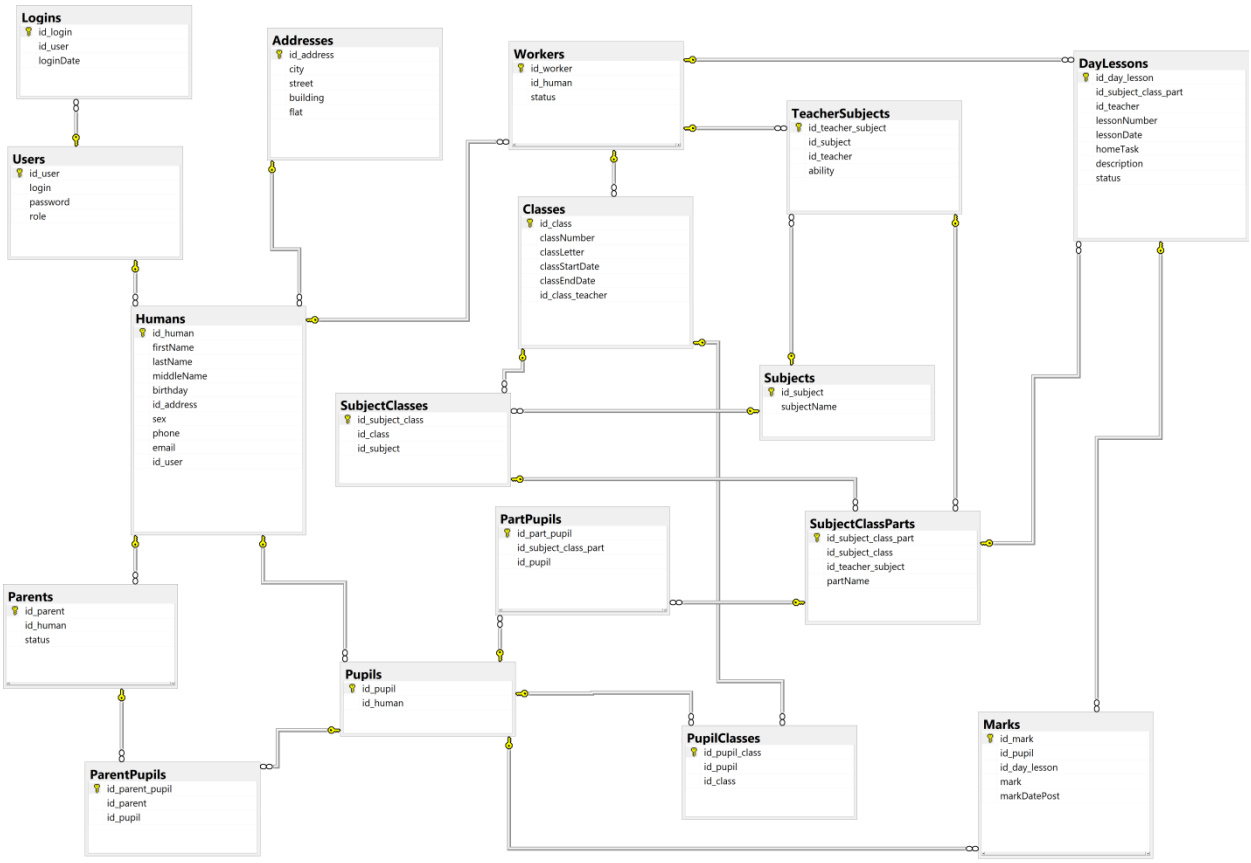


Рисунок 2.4 – Схема бази даних інформаційної системи електронного документообігу загальноосвітньої школи

2.1.4 Опис бази даних

Як було сказано раніше, БД організована на базі Microsoft SQL Server. Підключення й відправка запитів до якої зображено на рисунку 2.5.

```
string connectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;
using (SqlConnection connection = new SqlConnection(connectionString))
{
    connection.Open();
    SqlCommand command = new SqlCommand();
    command.CommandText = $@"select COUNT(*) from DayLessons, SubjectClassParts, PartPupils
                            where lessonDate = '{date.ToString("d")}' and id_pupil = 1 and
                            DayLessons.id_subject_class_part = SubjectClassParts.id_subject_class_part and
                            SubjectClassParts.id_subject_class_part = PartPupils.id_subject_class_part";
    command.Connection = connection;
    SqlDataReader reader = command.ExecuteReader();

    reader.Read();
    count = Int32.Parse(reader[0].ToString());
    reader.Close();
}
```

Рисунок 2.5 – Підключення та відправка запиту до бази даних

Для створення підключення необхідно мати рядок конфігурації підключення. Його, звичайно, можна вказувати безпосередньо в коді

програми, але це викличе ряд труднощів при зміні сервера баз даних, адже тоді прийдеться змінити цей рядок у всіх місцях, де він зустрічається, а також обов'язково перекомпілювати програму. Для того, щоб цього уникнути, рядок конфігурації було переміщено в спеціальний конфігураційний файл, де він зберігається в окремому блоці (рис. 2.6).

```
<connectionStrings>
  <add name="DefaultConnection" connectionString="Data Source=localhost;
       Initial Catalog=NewSchoolView; Integrated Security=true;"
       providerName="System.Data.SqlClient" />
</connectionStrings>
```

Рисунок 2.6 – Рядок підключення до БД в конфігураційному файлі

Після отримання рядка підключення, створюється об'єкт класу SqlConnection й викликається метод Open, в якому й безпосередньо проходить спроба підключення. Запит рядкового типу присвоюється властивості CommandText об'єкту SqlCommand, точно так, як і успішно підключений об'єкт SqlConnection властивості Connection. А далі в залежності від типу запиту виконується зчитування даних (у запитах типу SELECT)або отримується кількість оброблених рядків (у запитах типу INSERT, UPDATE, DELETE).

У даному випадку, після отримання з бази даних усієї необхідної інформації, підключення закривати не потрібно, адже об'єкт SqlConnection був створений у конструкції using, і це буде зроблено автоматично.

2.2 Розробка алгоритму автоматизованого виставлення тематичної оцінки

Одною із найголовніших задач інформаційної системи на ряду із зберіганням даних є автоматизація виконання рутинних процесів. Одним із таких є виставлення тематичних оцінок.

Метод, що виконує задачу виставлення тематичних оцінок підгрупі або цілому класу, приймає один аргумент в якості параметра, а саме, ідентифікатор даної підгрупи, завдяки якому та зв'язкам реляційної бази

даних можна буде з легкістю визначити клас, предмет, перелік учнів, викладача тощо.

Далі виконується запит, який отримує останню дату тематичної оцінки та контрольної роботи. Відповідність даних досягається завдяки зв’язкам між таблицями а також перевірки ідентифікатора підгрупи (рис. 2.7).

```
$@"SELECT MAX(case status when 3 then lessonDate else NULL end) as Tematichna,
  MAX(case status when 2 then lessonDate else NULL end) as Kontrolna
  FROM SubjectClassParts, TeacherSubjects, DayLessons
  WHERE DayLessons.id_subject_class_part = SubjectClassParts.id_subject_class_part
      and SubjectClassParts.id_teacher_subject = TeacherSubjects.id_teacher_subject
      and SubjectClassParts.id_subject_class_part = {classPart}
      and (TeacherSubjects.id_teacher = {Session["id_teacher"]}
          or DayLessons.id_teacher = {Session["id_teacher"]})";
```

Рисунок 2.7 – Запит для отримання дати останньої тематичної оцінки та контрольної роботи

Після цього, проводиться перевірка чи контрольна,взагалі,була проведена та, чи дата останньої тематичної оцінки не перевищує дату останньої контрольної роботи. Це все робиться для того, щоб не виставляти некоректну оцінку й не дублювати уже, можливо, існуючі дані.

Після успішного проходження усіх перевірок виконується SQL процедура InsertTematychnaDayLesson для створення нового стовпця для оцінок зі статусом «Тематична оцінка» (рис. 2.8). Важливим фактором, що призвів до використання процедури замість звичайного запиту типу «INSERT», є можливість повернення ідентифікатору стовпця відразу після його створення.

```
USE [NewSchoolViewN]
GO
/****** Object:  StoredProcedure [dbo].[InsertTematychnaDayLesson]
Script Date: 17/05/2020 16:25:48 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      Taria
-- Create date: 21-03-2020
-- Description: Insert into DayLesson tematychna lesson and get it ID
-- =============================================

ALTER PROCEDURE [dbo].[InsertTematychnaDayLesson]
    @SubjectClassPart varchar(10),
    @LessonDate varchar(10),
    @TematychnaID int OUTPUT
AS

INSERT INTO DayLessons (id_subject_class_part, id_teacher,
    lessonNumber, lessonDate, homeTask, description, status)
    VALUES (@SubjectClassPart, NULL, 15, @LessonDate, NULL, NULL, 3);

SET @TematychnaID = @@IDENTITY
```

Рисунок 2.8 –SQL процедура InsertTematychnaDayLesson

Наступним кроком є формування запиту для отримання масиву даних для кожного учня класу, що містить середнє арифметичне усіх оцінок та середнє арифметичне контрольних робіт у цій темі(рис. 2.9).

```
$@"SELECT id_pupil, AVG(Cast(m as smallmoney)) as avgMark, AVG(Cast(kontrolmark as smallmoney)) as kr
  FROM (
  SELECT id_pupil, case q1.pf when q2.ps then mark else null end as m,
      case status when 2 then (case q1.pf when q2.ps then mark else null end) else null end as kontrolmark
  FROM
      (SELECT Marks.id_pupil as pf, (case mark when 0 then null else mark end) as mark, DayLessons.id_day_lesson as ff
      FROM Marks right join PartPupils on PartPupils.id_pupil = Marks.id_pupil,
          SubjectClassParts, DayLessons
      WHERE DayLessons.id_subject_class_part = SubjectClassParts.id_subject_class_part
          and SubjectClassParts.id_subject_class_part = PartPupils.id_subject_class_part
          and DayLessons.id_day_lesson = Marks.id_day_lesson
          and lessonDate > '{tematichnaDate.ToString("yyyy.MM.dd")}' and lessonDate <= '{kontrolnaDate.ToString("yyyy.MM.dd")}'
          and SubjectClassParts.id_subject_class_part = {classPart}) as q1
      right join (SELECT id_pupil as ps, id_pupil, status, DayLessons.id_day_lesson as ss
      FROM PartPupils, SubjectClassParts, TeacherSubjects, DayLessons
      WHERE DayLessons.id_subject_class_part = SubjectClassParts.id_subject_class_part
          and SubjectClassParts.id_subject_class_part = PartPupils.id_subject_class_part
          and SubjectClassParts.id_teacher_subject = TeacherSubjects.id_teacher_subject
          and lessonDate > '{tematichnaDate.ToString("yyyy.MM.dd")}' and lessonDate <= '{kontrolnaDate.ToString("yyyy.MM.dd")}'
          and SubjectClassParts.id_subject_class_part = {classPart}
          and (TeacherSubjects.id_teacher = {Session["id_teacher"]} or DayLessons.id_teacher = {Session["id_teacher"]})) as q2
      on q1.ff = q2.ss
  ) as fin
  group by id_pupil";
```

Рисунок 2.9 – Запит для отримання середнього арифметичного усіх оцінок та контрольних робіт останньої теми

Головний запит має декілька вкладених запитів різного рівня вкладеності. Перший підзапит другого рівня вкладеності отримує масив усіх

оцінок учнів з цього предмету в проміжку між останньою тематичною оцінкою й останньою контрольною роботою. Другий підзапит другого рівня вкладеності отримує масив усіх уроків, проведених у обраній темі цього класу. Від першого до другого підзапиту застосовується об’єднання справа по полю-ідентифікатору уроку, щоб в результаті отримати перелік усіх уроків для кожного учня з оцінкою, якщо вона є, або зі значенням NULL, якщо оцінки за цей урок у поточного учня немає. У головному запиті нульової вкладеності викликаються агрегатні отримання середнього арифметичного AVG.

Останніми кроками вважається формування безпосередньо тематичних оцінок та створення запиту для завантаження їх в базу даних (рис.2.10). Якщо учень отримував оцінки й був присутнім на контрольних роботах в даній темі, то його тематична оцінка має дорівнювати середньому арифметичному усіх оцінок теми, якщо ця середня оцінка не перевищує більше ніж на 2 бали середнє арифметичне контрольних робіт, а також не є більш ніж на 1 бал за нього меншим. У інакших випадках обирається крайня можлива оцінка.

```
string strTematychnaMarks = String.Empty;
while (reader.Read())
{
    int finalMark = -1;
    if(reader[1].ToString() != String.Empty && reader[2].ToString() != String.Empty)
    {
        double krMark, avgMark;
        krMark = Double.Parse(reader[2].ToString());
        avgMark = Double.Parse(reader[1].ToString());

        if (avgMark - 2 > krMark)
            finalMark = (int)Math.Round(krMark, MidpointRounding.AwayFromZero) + 2;
        else if (avgMark + 1 < krMark)
            finalMark = (int)Math.Round(krMark, MidpointRounding.AwayFromZero) - 1;
        else
            finalMark = (int)Math.Round(avgMark, MidpointRounding.AwayFromZero);
    }
    else if(reader[1].ToString() != String.Empty)
        finalMark = (int)Math.Round(Double.Parse(reader[1].ToString()), MidpointRounding.AwayFromZero);
    else if(reader[2].ToString() != String.Empty)
        finalMark = (int)Math.Round(Double.Parse(reader[2].ToString()), MidpointRounding.AwayFromZero);

    strTematychnaMarks += "(" + reader[0] + ", " + tematychnaID + ", " + finalMark + ", '" +
        nowDate.ToString("yyyy.MM.dd") + "'),";
}
reader.Close();

command.CommandText = "INSERT INTO Marks (id_pupil, id_day_lesson, mark, markDatePost) VALUES "
    + strTematychnaMarks.Substring(0, strTematychnaMarks.Length - 1);
```

Рисунок 2.10 – Формування запиту для завантаження тематичних оцінок в БД

2.3 Програмне забезпечення

Усі етапи розробки інформаційної системи електронного документообігу були проведені з використанням звичайного персонального комп’ютера, що працює під управлінням операційної системи Windows 10.

На перших етапах розробки безпосередньо веб-сторінок було використано графічний редактор векторної графіки CorelDraw, в якому були спроектовані всі елементи векторної графіки на сторінках (фоновий малюнок сторінки авторизації, фоновий малюнок сторінок авторизованих користувачів та деякі емблеми для органів навігації). А також в графічному редакторі було створено макети сторінок вцілому.

Далі було використано середовище розробки Microsoft Visual Studio Code, в якому було виконано верстку та стилізацію усіх веб-сторінок інформаційної системи. А також закладено базис для обробки подій викликаних діяльністю користувачів системи електронного документообігу.

Наступним кроком було використання середовища розробки Microsoft Visual Studio, в якому завдяки технологій ASP .NET Core Web API було створено головний проєкт системи. Сюди було імплементовано та адаптовано сторінки, розроблені раніше, а також була здійснена розробка та налагодження всієї серверної частини інформаційної системи.

На різних етапах реалізованості використовувався веб-переглядач з інструментами розробника Google Chrome, у якому дуже зручно відслідковувати різноманітні параметри елементів верстки та стилізацій, а також переглядати вміст запитів з клієнтської частини до серверної.

Для створення та всеосяжного налагодження бази даних використовувався застосунок Microsoft SQL Server Management Studio, який надає інструментарій не тільки для створення та найпростішого заповнення таблиць у БД, а ще й для повного спектру операцій адміністрування та обслуговування.

Щоб створити програму для пристрою IoT та завантажити її безпосередньо в мікроконтролер було використано середовище розробки Arduino IDE. За допомогою цього середовища розробки можна створювати прошивки не тільки для Arduino-сумісних плат, але й для плат інших виробників завдяки завантажуваним модулям розширення.

Для налагодження апаратної складової, спряжених із системою інформаційного документообігу, приладів Інтернету речей використовувався логічний аналізатор від компанії Saleae. Він має пропрієтарне програмне забезпечення для взаємодії з комп’ютером Saleae Logic. Ця, на перший погляд, примітивна програма має широкий функціонал: вибір частот роботи логічного аналізатору та загального часу проведення вимірювань, встановлення тригерів по різних подіях окремо на кожному каналі, налаштування автоматичного розпізнавання даних на лініях цифрових протоколів тощо. Також варто зазначити, що Saleae Logic надає користувачеві можливість зручної обробки та аналізу сигналу після отримання повної вибірки, сюди можна віднести: розгалужену система

вимірювань, систему пошуку за параметрами та, звичайно, можливість збереження отриманої вибірки у файл задля доопрацювання її в майбутньому.

Також на етапі налагодження апаратної складової пристроїв IoT було використано Rigol Ultra Scope Software. Цей за стосунок можна віднести в категорію не найважливіших, але тих, що полегшують життя розробника. За допомогою даної програми можна управляти фізичним осцилографом віддалено з персонального комп'ютера, а також надзвичайно легко створювати знімки екрана самого осцилографа у високій роздільній здатності. Ця програмна система дозволяє не відволікатися на зміну параметрів вимірювального пристрою при роботі за комп'ютером, що іноді економить досить багато часу.

2.3.1 Опис структури системи електронного документообігу

Уся система електронного документообігу на етапі перегляду проекту в середовищі розробки у вікні оглядача рішень представлена у вигляді файлів і папок, описаних в таблиці 2.1. Більшість файлів і папок було згенеровано середовищем розробки ще на етапі створення проекту автоматично відповідно до норм та правил MVC, але їх наповнення формувалося протягом усього процесу розробки проекту.

Після компіляції проєкту та підготовки до публікації структура виглядає наступним чином:

- Тека «bin»;
- Тека «Content»;
- Тека «Views»;
- Файл «favicon.ico»;
- Файл «Global.asax»;
- Файл «packages.config»;

  Файл «Web.config».

Таблиця 2.1 – Опис головних файлів та папок інформаційної системи електронного документообігу загальноосвітньої школи

Назва елемента	Опис
Посилання	Тут зберігаються усі посилання на сторонні бібліотеки, які під’єднуються до інформаційної системи.
AppStart	Зберігає ряд статичних файлів, які зберігають логіку ініціалізації застосунку при запуску.
Content	Зберігає допоміжні файли, які не містять код на C# чиJavaScript, і розгортаються разом із за стосунком, наприклад стилі css.
Controllers	Зберігає файли класів контролерів.
Models	Зберігає файли моделей.
Views	Зберігає вигляд кожної із сторінок, що міститься в додатковій однойменній папці.
favicon.ico	Емблема веб-сайту.
Global.asax	Файл, що запускається при старті системи та виконує початкову ініціалізацію.
packages.config	Файл, що зберігає встановлені в проекті пакети Nuget.
Web.config	Файл конфігурації застосунку.

2.3.2 Перелік функцій та особливості взаємодії компонентів інформаційної системи електронного документообігу

Після натиснення будь-якої кнопки на клієнтській стороні, яка передбачає операцію зв’язку із сервером запускається низка подій:

1) Дія вловлюється обробником подій;
2) Виконуються перевірка всіх необхідних параметрів, валідація полів (при необхідності) та інші підготовчі операції;
3) Формується AJAX запит до сервера, в параметрах якого вказується метод запиту, тип даних, назва контролера та методу, до якого буде виконано підключення, а також безпосередньо аргументи, що передаються;
4) Спрацьовує викликаний метод на серверній стороні, в якому виконуються всі необхідні розрахунки, зв’язки з базою даних та іншими методами;
5) Метод повертає результуючі дані. Ці дані можуть бути різноманітних типів, найчастіше це JSON або String;
6) Якщо клієнтська сторона переконалася в успішному виконанні запиту, виконується процес парсингу та застосування отриманих даних.

На серверній стороні найголовнішими функціональними елементами вважаються контролери. У даній інформаційній системі їх було реалізовано три. Усі вони, а також списки їх складових функцій представлені у додатку А.

2.3.3 Синтетичне тестування сайту інформаційної системи

Тестування веб-сайту системи виконувалося в веб-переглядачі Google Chrome із застосуванням інструментів розробника.

Спочатку було проведено тест швидкості завантаження сторінки авторизації системи (рис.2.11). Це можна зробити в панелі Network, що дозволяє досліджувати процес завантаження сторінки та всіх файлів, що

підвантажуються. Було визначено, що час завантаження менше 0,3 секунди, що вважається прекрасним показником.

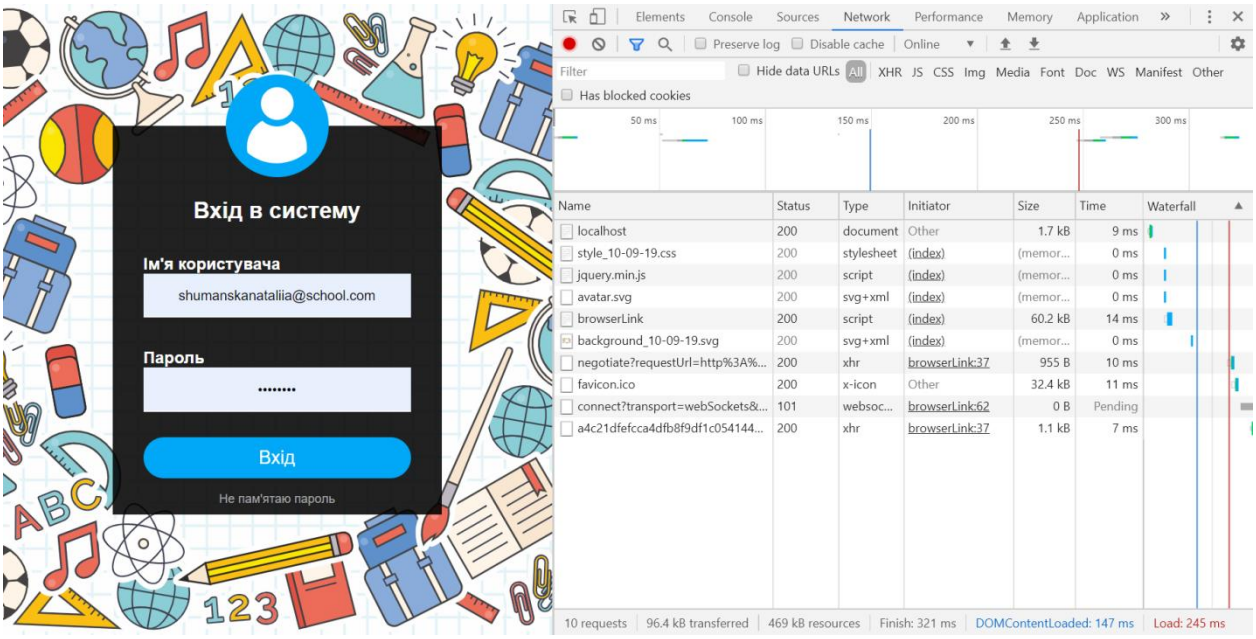


Рисунок 2.11 – Тестування швидкості завантаження сторінки авторизації

Наступним кроком було тестування таких параметрів, як продуктивність, доступність та оптимальність рішень (рис. 2.12). Це було зроблено в панелі Audits.

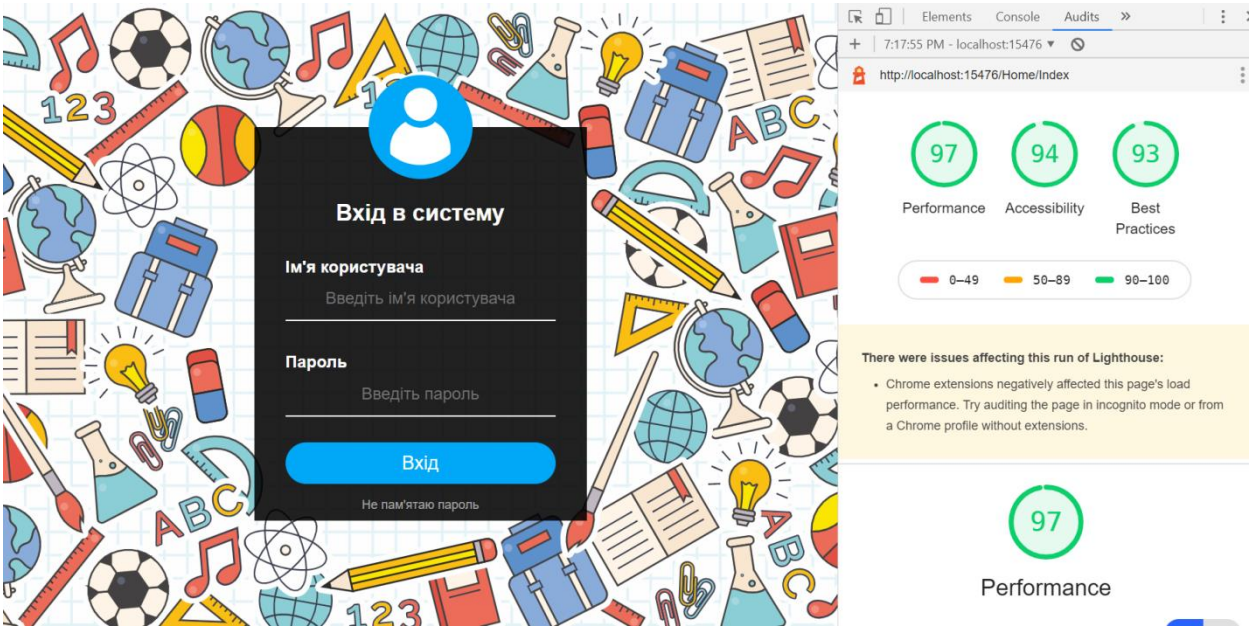


Рисунок 2.12 – Тестування продуктивності, доступності та оптимальності рішень сторінки авторизації

При тестуванні однієї із найважчих операцій, автоматичного виставлення тематичних оцінок, були отримані узагальнені максимально

приближені до реальності дані, що лягли в основу побудови часової діаграми (рис. 2.13).

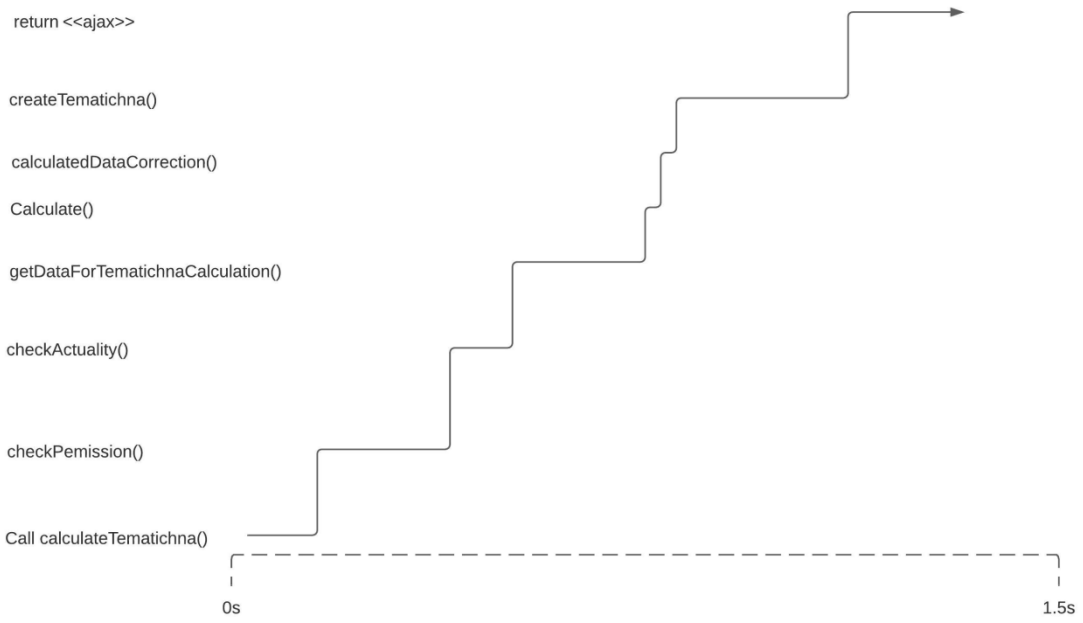


Рисунок 2.13 – Часова діаграма, подрібненої на окремі операції, функції автоматичного виставлення тематичної оцінки

Абсолютно всі показники при тестуванні показують достойні результати. Це буде сприяти зручності користування системою, а також дозволить пошуковим системам розміщувати посилання на сторінку системи ближче до вершини списку.

Висновки до розділу

У цьому розділі було сформований повний список технологій та повноцінну архітектуру бази даних, що використані для розробки ядра системи з графічною оболонкою у вигляді веб-сайту, описано структуру сторінок системи для різний типів користувачів, можливості та повний спектр функцій, якими забезпечує система кінцевого користувача.

Розділ містить опис програмного забезпечення для створення програмної складової системи. І, найголовніше, тут описано хід реалізації основних функціональних алгоритмів системи та результати її синтетичного тестування.

РОЗДІЛ 3. РОЗРОБКА АРХІТЕКТУРИ ТА АЛГОРИТМІВ РОБОТИ СПРЯЖЕНИХ З ІНФОРМАЦІЙНОЮ СИСТЕМОЮ ПРИСТРОЇВ ІНТЕРНЕТУ РЕЧЕЙ

Звичайно, система, що володіє функціями накопичення, розповсюдження, а іноді навіть генерації інформації надає загальноосвітній школі велику кількість переваг. Але, як показує практика, цих функцій уже недостатньо. У сучасних реаліях подібна система повинна мати змогу взаємодіяти з різноманітними пристроями Інтернету речей для збільшення обсягу автоматизації.

Для виконання вимог технологічного завдання, що стосуються розробки та налагодження пристроїв Інтернету речей, було використано наступні програмні та апаратні технології:

- Мова програмування C++;
- Технологія налагодження за допомогою мультиметра;
- Технологія налагодження за допомогою осцилографа;
- Технологія налагодження за допомогою логічного аналізатора;
- Цифровий протокол UART;
- Цифровий протокол SPI.

Найбільш широко використовуваною мовою програмування мікроконтролерів вважається мова C. Саме вона є тією золотою серединою між максимальною оптимізацією і простотою написання коду. Більше оптимізованою мовою може бути хіба що Assembler. Але у наші часи бізнес диктує свої умови по швидкості розробки, тому все більше й більше нових проектів пишуться на мові **C++**. Це високорівнева компіляційна мова програмування з C-подібним синтаксисом, статичною типізацією, яка уже багато років вважається однією із найпопулярніших мов програмування у самих різноманітних галузях: від розробки ігор до програмування пристроїв Інтернету речей. C++ є надзвичайно потужною й водночас швидкою мовою програмування, яка унаслідувала від C майже безмежні можливості по роботі з пам’яттю, водночас набувши статусу об’єктно орієнтованої, що уже

дозволяє представляти програму, як сукупність взаємодіючих між собою класів й об’єктів.

На рівні коду C++ може вважатися мовою з хорошою переносимістю, звичайно якщо не використовуються специфічні функції чи регістри конкретного процесора. Але після компіляції в виконуючий файл з машинними інструкціями його вже не можна буде запустити на будь-якій іншій платформі навіть з однаковою архітектурою [28].

Написання програмного коду це, звичайно, важливий етап розробки пристрою Інтернету речей, але далеко не єдиний. Для того щоб з’явилася можливість щось запрограмувати спочатку необхідно спроектувати, розробити й налагодити апаратне забезпечення. Всі ці процеси не є можливими без спеціалізованого обладнання й правильних технологій роботи з ним.

Базовим і найбільш часто використовуваним пристроєм при розробці є **мультиметр**. Це універсальний комбінований пристрій для вимірювання різних величин [31].

У більшості мультиметрів присутні наступні функції:

- Вимірювання напруги – базова функція. Два щупи встановлюютьсяв схемі паралельно елементу, на якому необхідно дізнатися падіння напруги;
- Вимірювання струму – базова функція. Для проведення вимірювань електричне коло необхідно розірвати й щупами підключитися в цей розрив, щоб виникло послідовне з’єднання. Зазвичай для вимірювання струму виділяється декілька терміналів для підключення анодного щупа в залежності від величини максимально допустимого струму;
- Вимірювання опору – базова функція. Для проведення вимірювань опору якогось конкретного елемента, його необхідно вилучити з схеми, адже інші компоненти схеми можуть вносити похибку у вимірювання. Щупи підключаються паралельно елементу, що вимірюється;

- Вимірювання ємності – опціональна функція. Алгоритм дій такий же, як і в опору;
- Вимірювання індуктивності – опціональна функція. Алгоритм дій такий же, як і в опору;
- Продзвонка – базова функція. Для проведення вимірювань опору зі звуковою індикацією. Дуже корисна при пошуку коротких замикань або навпаки обривів у схемі. Алгоритм дій такий же, як і в опору;
- Тестування біполярних транзисторів – базова функція. Для перевірки працездатності а також коефіцієнта підсилення транзисторів різної провідності: NPN таPNP. Зазвичай на корпусі присутній спеціальний термінал для під'єднання транзисторів;
- Логування даних - опціональна функція. У мультиметрах, що призначені для розробки, а не для домашнього використання, зазвичай присутній спеціалізований оптично розв'язаний роз'єм, що дозволяє підключити пристрій до персонального комп'ютера по USBінтерфейсу й проводити легування даних, що зчитуються.

Наступним надзвичайно корисним пристроєм є **осцилограф**. Він дозволяє вимірювати амплітудні, частотні характеристики та форму електричного сигналу [32].

Незалежно від виробника, класу точності, типу (цифровий, аналоговий), кількості вхідних каналів – всі осцилографи володіють наступним функціоналом:

- Вимірювання напруги сигналу в різні моменти часу;
- Вимірювання частоти (періоду) сигналу;
- Визначення зсуву фаз;
- Визначення спотворення еталонного сигналу при зміні конфігурації компонентів схеми;
- Визначення співвідношення амплітуди шуму до амплітуди сигналу.

У сучасних осцилографах також присутні й інші надзвичайно зручні функції, такі як тригер, що дозволяє фіксувати сигнал для більш детального його дослідження, чи вбудовані математичні функції, які, наприклад, можуть розкладати сигнал в ряд Фур'є в режимі реального часу.

Схожим по функціоналу до осцилографа, але призначеним для дослідження цифрових сигналів є **логічний аналізатор**. Він може протягом певного часу збирати цифрові дані (0 або 1) одночасно з декількох каналів, а потім відобразити їх послідовність для детального дослідження. Часто використовується для налагодження цифрових пристроїв та їх компонентів. Більшість моделей можуть розпізнавати дані на пінах цифрових протоколів, таких як UART, SPI, I2C та інших [33].

UART (Universal asynchronous receiver/transmitter) – універсальний послідовний асинхронний цифровий протокол прийому й передачі даних [34]. Зазвичай використовується для спряження тільки двох пристроїв (хоча можлива конфігурація з одним masterпристроєм і декількома slave). Для підключення, як правило, використовується три виводи: земля живлення, пін прийому та пін передачі. Пін прийому master пристрою з'єднується з піном передачі slave, а пін передачі master з піном прийому slave.

Швидкість передачі по UART повинна співпадати на всіх підключених пристроях. Вона конфігурується в ручному режимі для кожного під'єднаного пристрою. Зазвичай використовують одну із стандартних швидкостей передачі в межах від 300 біт/с до 1 Мбіт/с.

Дані по UART передаються пакетами, найчастіше, 8 біт, але крім цих корисних даних до пакета додаються ще два технічних біта: один на початку пакета (сигналізує приймачу про початок передачі) й один в кінці пакета (сигналізує приймачу про кінець передачі). Довжина кожного біту чітко контролюється, адже час – це єдиний параметр, який дозволяє відрізнити кінець одного біту від початку наступного. Саме для чіткого контролю часу біля більшості мікросхем USB–TTL конверторів встановлюють тактовий генератор на основі кварцу.

Стандартом прийнятий стан піна передатчика в пасивному режимі - логічна 1, тому стоп-біт також логічна 1, старт-біт логічний 0. Приймач знаходиться в стані покою до того моменту, поки в нього не викличеться переривання по переходу в логічний 0, відразу розпочинається підрахунок часу по тактах генератора. Якщо проходить більше половини часу тривалості одного біта й сигнал продовжує триматися в логічному 0, то приймач фіксує початок прийому пакету. Остання операція внесена в стандарт для того, щоб відкинути помилкові спрацювання.

UART став найрозповсюдженішим протоколом передачі даних, що використовується для налагодження пристроїв Інтернету речей завдяки своїй простоті, невибагливості та широкій розповсюдженості у більшості існуючих моделей мікроконтролерів.

Другим по розповсюдженості можна впевнено назвати **SPI** (Serial Peripheral Interface) – це синхронний послідовний периферійний протокол передачі даних, який використовується для стабільного й високошвидкісного спряження двох або більше пристроїв. Один пристрій завжди головний (master), а інші підрядні (slave) [35].

Протокол для роботи вимагає щонайменше чотири сигнальних лінії:

- MISO(master input slave output) – лінія прийому головного пристрою, що з’єднується з лінією передачі всіх підрядних пристроїв;
- MOSI(master output slave input) - лінія передачі головного пристрою, що з’єднується з лінією прийому всіх підрядних пристроїв;
- SCK (serial clock) – лінія тактування, що здійснюється виключно головним пристроєм;
- CS (chip select) – щонайменше одна лінія вибору конкретного підрядного пристрою для взаємодії з головним. Часто (крім випадків реалізації кільцевої структури зв’язку) кількість ліній вибору підрядних пристроїв дорівнює загальній кількості підрядних пристроїв.

Як уже було сказано вище, джерело тактування на всі пристрої одне – головний пристрій. Звідси випливає велика кількість переваг, порівняно з раніше розглянутим протоколом UART: частоту тактувань необхідно конфігурувати лише на одному пристрої (головному), а всі інші підлаштуються автоматично; частота тактувань може відрізнятися при взаємодії головного пристрою з різними підрядними; досягається порівняно висока стабільність синхронізації, адже не може бути розсинхронізації пристроїв через нечітке співпадіння роботи різних тактових генераторів, адже він всього один. Варто зазначити, що частота тактів не повинна перевищувати меншу максимально допустиму частоту одного із взаємодіючих пристроїв.

Подібно до UART, дані в протоколі SPI передаються пакетами. Довжина пакетів, зазвичай, 8 біт. Першим кроком перед передачею необхідно перевести пін вибору необхідного підрядного пристрою в стан логічний 0 (адже стандартом визначено, що в пасивному стані всі піни CS повинні знаходитися в стані логічної 1). Далі починається генерація тактових імпульсів на лінії SCK, паралельно з якою іде прийом і передача даних. Підрядний пристрій відслідковує рівень сигналу на лінії MOSI при кожному такту. Якщо рівень високий, то вважається, що отримано біт 1, інакше – 0. Аналогічна ситуація й з прослуховуванням лінії MISO головним пристроєм.

Стандарт передбачає повернення сигналу вибору підрядного пристрою у стан логічної 1 після закінчення передачі кожного пакету для скидання лічильника тактів, але, як показує практика, у взаємодії з деякими пристроями це робити необов’язково.

3.1 Розробка пристрою автоматичної ідентифікації учнів

Для реалізації ідеї приладу контролю пропуску учнів у школу необхідно:

1) Датчик, що міг би зчитувати якісь унікальні параметри учня (відбиток пальця, райдужну оболонку ока чи просто персональну ID-картку);
2) Мікроконтролер, що б опрацьовував дані, отримані зі зчитуючого пристрою та зв’язувався з сервером по HTTPпротоколу;
3) Спеціалізований контролер на веб-сервері для опрацювання запитів цього пристрою.

Після вивчення та аналізу існуючих датчиків та мікроконтролерів, що володіють необхідними характеристиками було зроблено оптимальний вибір: RFID модуль RC522 та мікроконтролер ESP8266.

Радіочастотна ідентифікація (RFID) – це безконтактна технологія ідентифікації об’єктів за допомогою радіочастотного каналу зв’язку. Кожна електронна мітка, будь-то ID карта, брелок чи кільце з вбудованим чіпом, має свій власний унікальний ідентифікатор, за допомогою якого й проводиться ідентифікація об’єктів [37]. Навколо зчитувача за допомогою радіально розміщених доріжок створюється електромагнітне поле, яке вловлюється також подібно розміщеними доріжками всередині електронної мітки. Цю технологію можна грубо вважати електричним трансформатором без сердечника, саме завдяки якому мітка, в безпосередній близькості до зчитувача, забезпечується енергією для передачі корисної інформації (ідентифікаційного номеру, даних пам’яті тощо).

До переваг технології RFID варто віднести: безконтактний спосіб взаємодії, можливість прихованого встановлення міток, високу швидкість зчитування даних, можливість встановлення зчитувачів у місцях де в ефірі присутня велика кількість шумів різних частот, неможливість підробки.

Обраний модуль RFID зчитувача володіє наступними характеристиками:

- Напруга живлення: 3,3 В;
- Сила струму, що поглинається: 13-26 мА;
- Частота взаємодії з мітками: 13,56 МГц;

- Дальність спрацювання: 0-50 мм;
- Максимальна швидкість передачі даних: 10МБіт/с (залежить від протоколу зв’язку);

Модуль підтримує можливість взаємодії по одному із трьох протоколів: UART, SPI та I2C [38]. Вибір бажаного інтерфейсу визначається подачею логічної одиниці на визначений вивід мікросхеми. За замовчуванням обрано SPI, його й будемо використовувати в проекті, адже перехід на UART призведе до зниження швидкості обміну даними, та UART буде використовуватися для налагодження роботи пристрою, а апаратний I2C не передбачений конструкцією обраного мікроконтролера. На рисунку 3.1 зображено сам модуль RC522 та призначення кожного з його виводів:

1) SS – вивід вибору пристрою для взаємодії з мікроконтролером. При подачі логічного нуля модуль виходить із стану очікування і стає готовим для взаємодії;
2) SCK – вивід зчитування тактування шини SPI від мікроконтролера;
3) MOSI – вивід зчитування корисних даних від мікроконтролера;
4) MISO – вивід передачі корисних даних до мікроконтролера;
5) IRQ – вивід генерації апаратних переривань;
6) GND – вивід, що підключається до катоду живлення;
7) RST – вивід для перезавантаження модуля;
8) VCC - вивід, що підключається до аноду живлення.

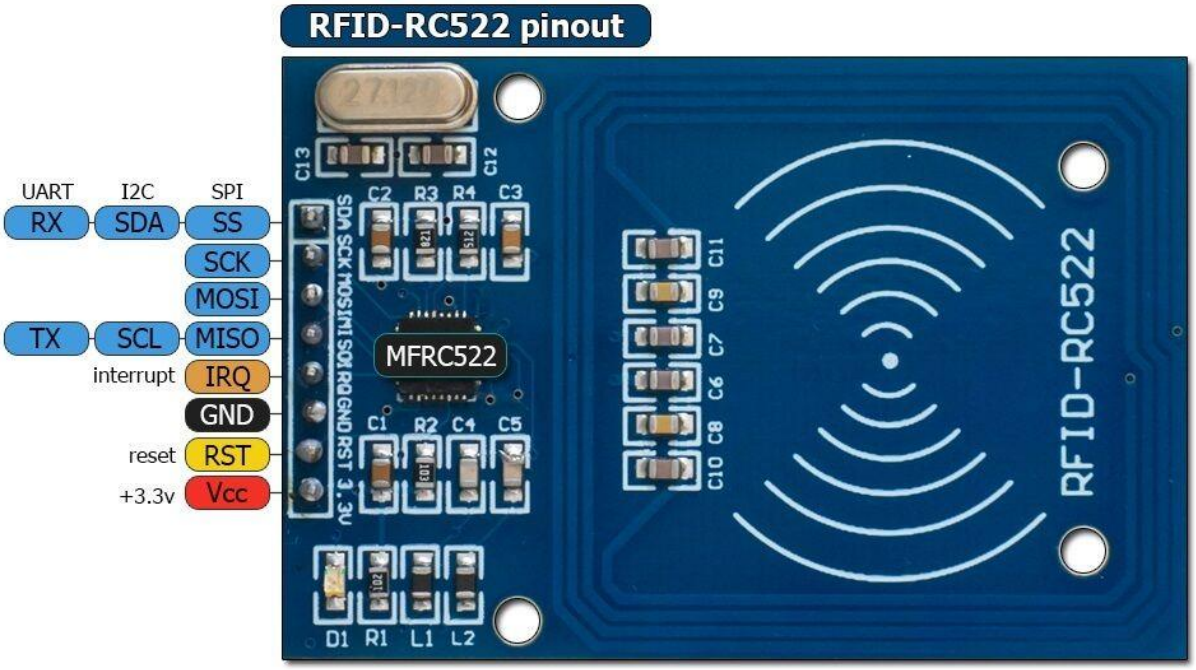


Рисунок 3.1 – Розміщення та призначення виводів модулю RC522

У якості мікроконтролера ESP8266 обрано не випадково, адже, перш за все, у нього вбудована підтримка роботи з безпровідним Wi-Fi протоколом, наявність якої є обов'язковою при побудові бездротових пристроїв Інтернету речей. Також до вирішальних плюсів у виборі мікроконтролера можна віднести частоту його роботи, що починається від 80 МГц (з можливістю її збільшення за допомогою зміни запобіжних бітів), велику кількість портів вводу-виводу, апаратну підтримку UART та SPI, програмну підтримку I2C та низьке енергоспоживання, що в максимальній завантаженості ядра з увімкненим одночасно процесом передачі даних не перевищує 215мА. На жаль, мікроконтролер не має вбудованої flash-пам'яті, але виробник це з легкістю компенсує виробництвом закритих модулів з розміщеними на них мікросхемах flash-пам'яті (у нашому випадку 4 МБ) [36].

Звичайно, можна просто використати тільки мікроконтролер, створивши його електронну компонентну обв'язку власноруч, але це неминуче призведе до збільшення часу розробки на перших етапах, що в свою чергу матиме негативні наслідки на всі інші етапи. Тому було обрано стратегію проводити розробку пристрою на платі для налагодження на базі мікроконтролера ESP8266.

В якості плати для налагодження було обрано NodeMCU V3. Крім самого мікроконтролера плата містить:

- Лінійний стабілізатор живлення на 3,3 В;
- Фільтруючі танталові конденсатори по живленню;
- USB-TTL перетворювач на базі мікросхеми CH340 організації зв’язку мікроконтролера і USB порту комп’ютера чи смартфону по UART протоколу;
- Micro-USB роз’єм для підключення шнура даних;
- Кнопка Reset для перезавантаження мікроконтролера у випадку його зависання чи необхідності почати виконання програми спочатку;
- Діод по входу живлення для передбачення неправильної полярності підключення блоку живлення;
- Інші конденсатори, резистори та кварц, що передбачені стандартною схемотехнікою ESP8266 та CH340.

Схема зв’язку модулів між собою та з веб-сервером зображено на рисунку 3.2.

Рисунок 3.2 – Схема зв’язку модулів

Після того, як мікроконтролер зчитав дані з ID-картки учня, він відправляє запит до веб-сервера з цим ідентифікатором. Сервер, в свою чергу, зв’язується з базою даних для перевірки такого ідентифікатора в системі. У разі успішного підтвердження ідентифікатора, створюється новий запис у базі даних, у іншому випадку з сервера повертається код помилки на мікроконтролер.

3.1.1 Розробка та налагодження апаратної складової пристрою автоматичної ідентифікації учнів

Перш за все варто правильно під’єднати модуль радіочастотної ідентифікація до плати налагодження:

1) Вивід SS можна з’єднати з будь-яким портом вводу-виводу мікроконтролеру, адже він конфігурується програмно. Було обрано пін, що на платі відзначений як D4;
2) SCK під’єднується до генератора тактових імпульсів шини SPI, що об’єднаний з цифровим піном D5;
3) MOSI під’єднується до MISO мікроконтролера, що об’єднаний з цифровим піном D7;
4) MISO під’єднується до MOSI мікроконтролера, що об’єднаний з цифровим піном D6;
5) GND під’єднується до катоду живлення, що відзначений на платі NodeMCU теж як GND;
6) RST можна з’єднати з будь-яким портом вводу-виводу мікроконтролеру, адже він теж конфігурується програмно. Було обрано пін, що на платі відзначений як D3;
7) 3.3v під’єднується до катоду живлення, що відзначений на платі як 3v;

На етапі раннього налагодження все припаюється за допомогою дротів МГТФ навісним монтажем (рис. 3.3), адже це дозволяє швидко та легко змінювати конфігурацію пристрою. Пайка, на відміну від роз’ємного з’єднання, забезпечить надійний контакт, а фторопластова ізоляція провідників, на відміну від ПВХ, не руйнується від декількох навіть довготривалих нагрівань при пайці.

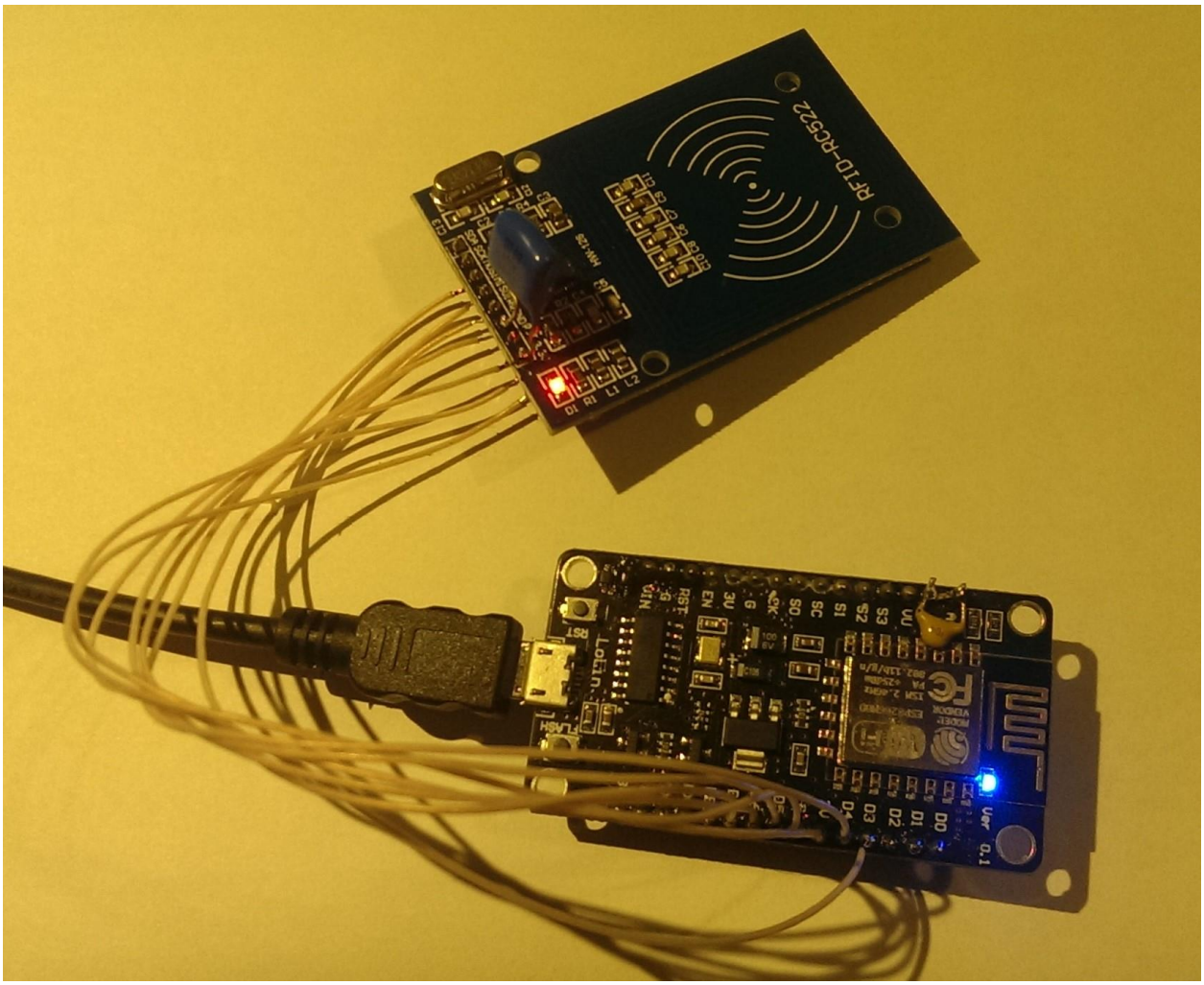


Рисунок 3.3 – З’єднання модуля RC522 та NodeMCU навісним монтажем

Написавши найпростішу програму, скомпілювавши й завантаживши її в мікроконтролер для тестового запуску взаємодії з модулем, виявилося, що пристрій не працює. Після перевірки апаратного підключення й програмного коду стало ясно зрозуміло, що проблема десь глибше.

Дослідження несправності було почате з вимірювання напруги на всіх контрольних точках схеми за допомогою мультиметра. Виявилося, що всі показники в нормі:

- Вхід живлення з micro-USB роз’єму – 5,1 В;
- Вихід напруги з лінійного стабілізатору напруги – 3,35 В;
- Вхід напруги на модуль RC522 – 3,31 В.

Наступним кроком було підключення логічного аналізатора до шини SPI та цифровізація сигналів на виводах MISO, MOSI, SCK, SS. Аніліз

показав, що мікроконтролер відправляє дані для ініціалізації модуля радіочастотної ідентифікації, а у відповідь постійно приходять хаотичні дані, що свідчить про часткову несправність роботи саме модуля RC522.

Для більш глибокого аналізу проблеми вирішено було провести ряд вимірювань за допомогою осцилографа. Було виявлено надзвичайно сильні пульсації на вході живлення модуля радіочастотної ідентифікації (рис. 3.4). Пульсації досягають 1В та мають частоту 55 кГц.

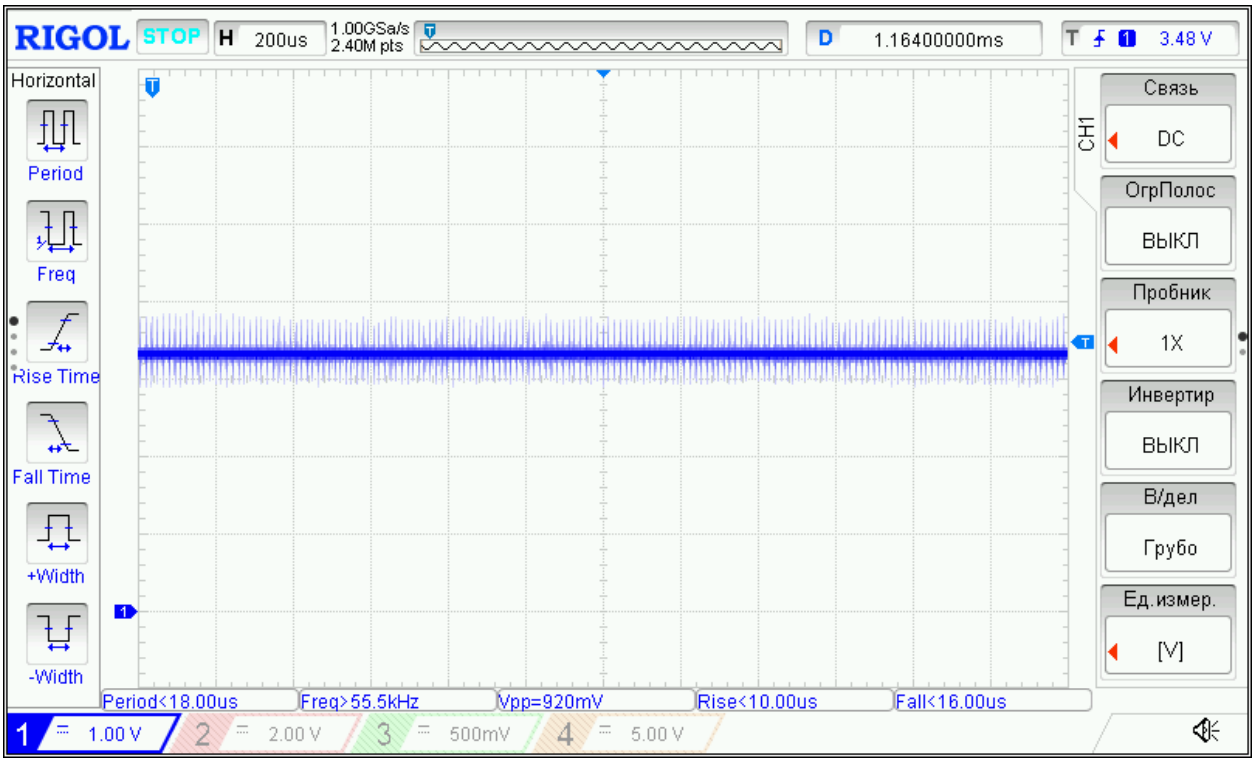


Рисунок 3.4 – Осцилограма пульсацій на шині живлення модуля RC522

Частковим вирішенням проблеми стало підключення паралельно шині живлення невеликого конденсатора ємністю 100нФ. Амплітуда пульсацій значно зменшилася до 500мВ (рис. 3.5).

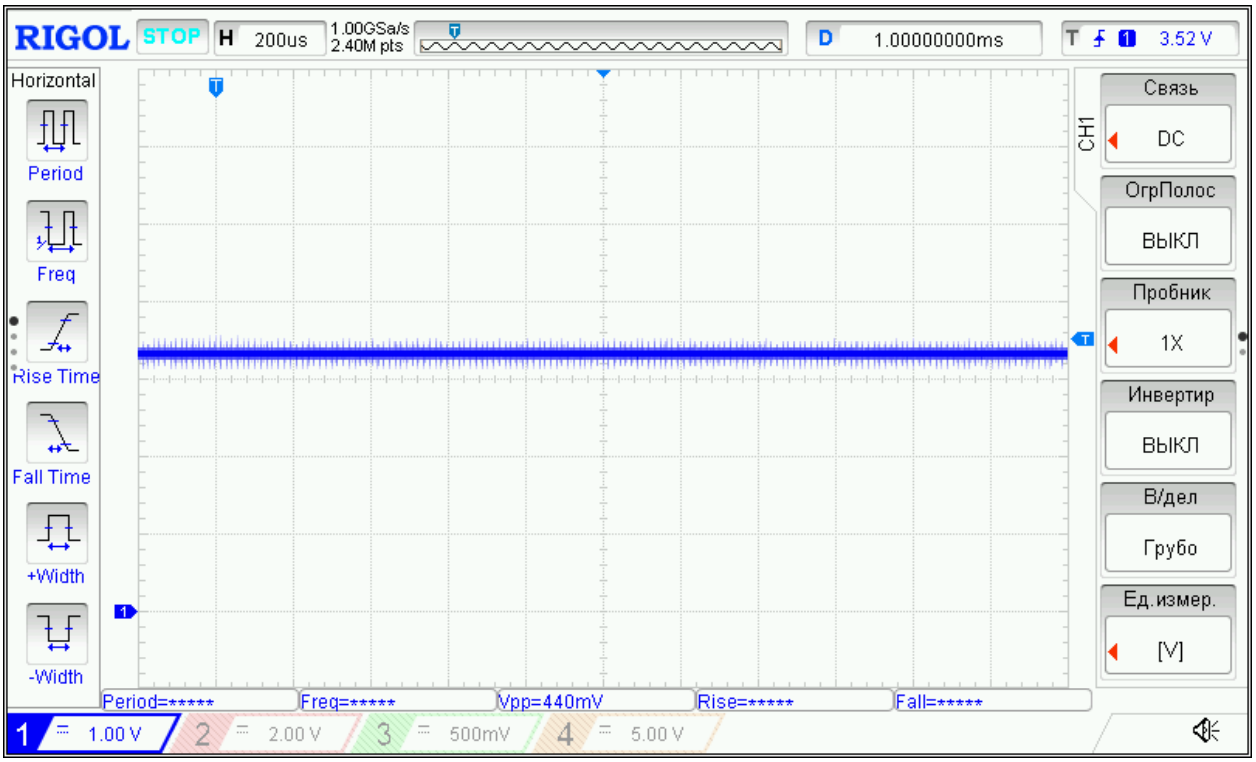


Рисунок 3.5 - Осцилограма пульсацій на шині живлення модуля RC522 з конденсатором

Всі наступні експериментами з різними конденсаторами бажаного результату не дали. Саме тому було вирішено створити окрему від мікроконтролера лінію живлення для модуля радіочастотної ідентифікації з хорошим фільтром. Так як модуль живиться від напруги 3.3В, а з micro-USB приходить 5В – необхідно встановити понижуючий стабілізатор напруги. Вибір впав на AMS-1117-3.3, адже він одночасно в собі комбінує низьку ціну й повністю підходящі для проєкта характеристики (вихідна напруга, максимальний струм тощо). Для фільтрації перед стабілізатором було встановлено LC-фільтр, що складається з індуктивності 1мГн та керамічного конденсатора 10мкФ. Після стабілізатора встановлено два конденсатори (10мкФ та 100нФ), кожен із яких повинен гасити шуми різних частот. Кінець кінцем, вдалося отримати ідеально чисте живлення на модулі.

Далі було знову підключено логічний аналізатор й проведено точно такий дослід, що й раніше. Ситуація змінилася. Тепер пристрій чітко проходив ініціалізацію й на всі однакові запити відповідав сталими

значеннями. Можна з легкістю відслідкувати закон взаємодії цього модуля по шині SPI (рис. 3.6):

1) SPI-ENABLE (пін SS) встановлюється в стан логічного нуля;
2) Запускається тактування на лінії SPI-CLOCK (пін SCK);
3) Передаються 8 біт корисних даних від мікроконтролера по шині MOSI (модуль радіочастотної ідентифікації в цей момент нічого не передає);
4) Зупиняється тактування;
5) Затримка;
6) Знову запускається тактування на лінії SPI-CLOCK;
7) Передаються 8 біт корисних даних від модуля по шині MISO (мікроконтролер в цей момент нічого не передає);
8) Зупиняється тактування;
9) SPI-ENABLE встановлюється в стан логічної одиниці.

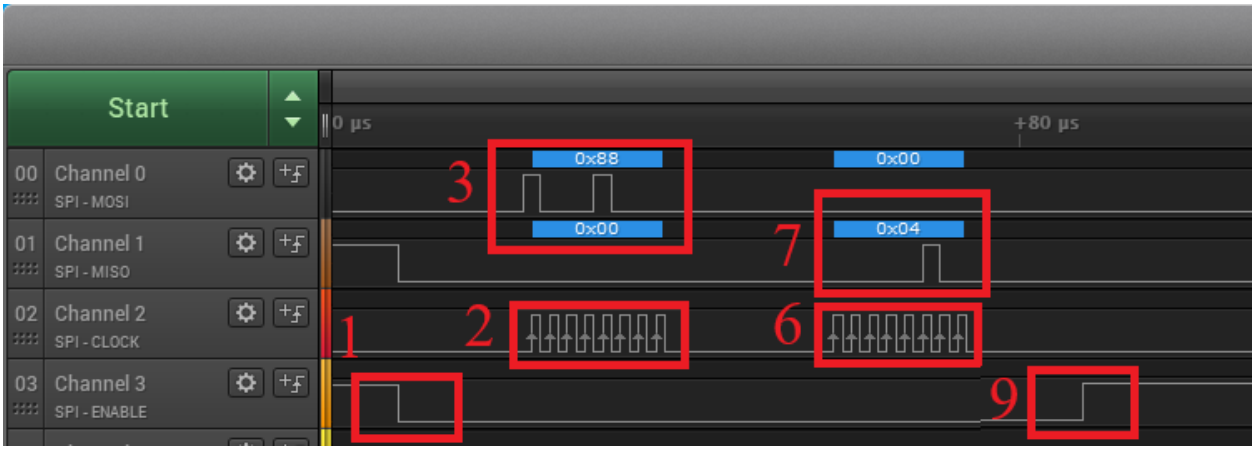


Рисунок 3.6 – Цифровізовані сигнали взаємодії з модулем радіочастотної ідентифікації

Із останніх модифікацій до схеми було додано двокольоровий світлодіод зі спільним анодом, включеним через резистор 200Ом, для індикації зчитування RFID мітки. Червоний колір свідчить про успішне зчитування, але помилку ідентифікації, а зелений про повністю успішно завершену операцію.

Тепер все працює й можна переходити до повноцінної розробки та налаштування програмної частини пристрою.

3.1.2 Розробка та налагодження програмної складової пристрою автоматичної ідентифікації учнів

Першим кроком в написанні повноцінно діючої програми для пристрою автоматичної ідентифікації учнів є написання мінімалістичного коду для запуску і налаштування всіх апаратних складових на програмному рівні. Для цього необхідно включити до складу програми дві бібліотеки через файли-заголовки: SPI.h – для правильної роботи мікроконтролера з шиною SPI та MFRC522.h – для взаємодії з модулем радіочастотної ідентифікації. В конструктор для створення об’єкту класу MFRC522 передаються в якості двох параметрів виводи мікроконтролера, що призначаються для управління пінами Reset та SS модуля-зчитувача RFID. У блоці setup відбуваються всі необхідні ініціалізації (портів вводу виводу, протоколу UART, SPI та безпосередньо RC522). Ну а в блоці loop знаходиться код, що виконується мікроконтролером циклічно на максимально можливій частоті. Саме у функції loop відбувається перевірка піднесення нової RFID мітки (для уникнення повторного читання мітки, що була уже зчитана) та перевітка успішного закінчення читання. Якщо всі перевірки пройшли успішно, то виконується найпростіший код виводу зчитаного унікального ID через інтерфейс UART. Вивід супроводжується спалахом світлодіода, для чого на його катодний вивід подається логічний нуль, адже анод за замовчуванням через резистор підтягнутий до позитивного полюсу джерела живлення. UART використовується виключно для налагодження і в закінченому готовому пристрої функціонувати не буде.

Провівши декілька тестувань й перевіривши повну працездатність конструкції на тестовому програмному коді, прийшов час для написання спеціалізованого контролера на стороні сервера, який буде обробляти запити від мікроконтролера. До контролера можна звернутися за допомогою GET методу, передавши в якості параметру унікальний ідентифікатор, зчитаний з RFID мітки. Контролер перевіряє існування такого ідентифікатору в базі даних системи електронного документообігу загальноосвітньої школи і, якщо

він існує, робить запис до таблиці Відвідувачі з відміткою часу проходження ідентифікації цією міткою, інакше мікроконтролер отримує інформацію про помилку.

Тепер настав час дописати програмний код пристрою автоматичної ідентифікації учнів. Необхідно додати файли-заголовки наступних бібліотек: ESP8266WiFi.h – для можливості керування вбудованим у ESP8266 Wi-Fi модулем та ESP8266HTTPClient.h – для можливості створення клієнта та відправки запитів по HTTP протоколу. До ініціалізацій блоку setup додамо спробу та очікування підключення до Wi-Fi мережі. Ну а в функції loop після успішного проходження уже існуючих перевірок додамо перевірку чи не було втрачене підключення до Wi-Fi мережі, формування рядка підключення до контролера по зчитаним даним з RFID мітки, вибір та виконання методу запиту та обробник отриманих даних.

3.2 Розробка пристрою автоматизованого бездротового управління шкільним дзвінком

Для реалізації ідеї приладу автоматизованого безпровідного управління шкільним дзвінком, що отримує дані безпосередньо від інформаційної системи електронного документообігу загальноосвітньої школи необхідно:

1) Мікроконтролер, який циклічно опитуватиме сервер на необхідність спрацювання дзвінка. Опитування повинно здійснюватися за допомогою безпровідної технології Wi-Fi по HTTP протоколу;
2) Силове реле з драйвером його управління для комутації мережевого навантаження у вигляді дзвінка. Максимальний теоретично можливий струм протікання через котушку дзвінка – 1А;
3) Спеціалізований контролер на веб-сервері для опрацювання запитів цього пристрою.

У якості головного ядра пристрою було вирішено використовувати теж мікроконтролер ESP8266 на базі плати для налагодження NodeMCU V3, адже цей пристрій також вимагає підключення до мережі Інтернет й базується на

подібному алгоритмі спряження з інформаційною системою електронного документообігу, як і пристрій автоматичної ідентифікації учнів.

Для комутації навантаження було обрано електромагнітне реле HF7520/005-HSTP, що володіє наступними характеристиками:

- Максимально допустима напруга комутації постійного струму – 30 В;
- Максимальний стабільний постійний струм – 1 А;
- Максимально допустима напруга комутації змінного струму – 250 В;
- Максимальний стабільний змінний струм при найгіршому значенні коефіцієнта потужності $\cos\varphi = 0{,}4$ – 8 А;
- Напруга живлення обмотки – 5 В;
- Струм утримання реле не перевищує – 0,05 А;
- Час спрацювання не перевищує – 15 мс.

Як альтернатива реле, розглядалися схеми комутації, що базуються на напівпровідникових симісторах. Вони мають одну перевагу перед релейними комутаторами – відсутність механічної складової, що, за дотримання усіх технологічних умов експлуатації, надає їм майже нескінченний ресурс роботи. Але симістори також мають і недоліки: виділення значної кількості тепла при роботі (при потужних навантаженнях цей пункт є критичним), внесення значних шумів у мережу та більш складна схемо-технічна складова, що вимагає специфічної оптичної розв’язки на базі оптосимісторної збірки.

3.2.1 Розробка та налагодження апаратної складової пристрою автоматизованого бездротового управління шкільним дзвінком

Електромагнітне реле HF7520/005-HSTP має всього чотири виводи: два для живлення котушки та два силові, що вмикаються в розрив електричного кола. На перший погляд все достатньо просто – необхідно просто підключити один вивід котушки до катоду живлення, а інший до одного із портів вводу-виводу мікроконтролера й управляти програмно. Але це не зовсім так. По-перше, обране реле дивиться від напруги 5В, які насправді приходять на

плату налаштування, але сам мікроконтролер живиться від 3,3В, що формуються стабілізатором напруги, який розміщений на платі для налаштування, тому напруга логічної одиниці на будь-якому із портів не перевищує 3,3В. По-друге, струм, що необхідний для замкнення й утримання контактів реле може перевищувати 50мА, а не перевищує можливості портів вводу-виводу щонайменше у п’ять разів. Саме тому для правильного управління котушкою реле необхідний драйвер, в ролі якого можна використати транзистор малої або середньої потужності. В цьому випадку підійдуть як польові транзистори з каналом N-типу, так і біполярні транзистори NPN типу. Головна різниця між ними заключається у способі управління: польові управляються за допомогою прикладання напруги до його затвора, а біполярні за допомогою пропускання струму через його базу. Мною було обрано транзистор малої потужності КТ605АМ з максимально допустимим током колектора 100мА і статичним коефіцієнтом передачі току 10…40. Тепер уже струм з порту вводу-виводу мікроконтролера на управляючу базу транзистора необхідно обмежити – це зроблено за допомогою резистора 100 Ом. Також варто зважати, що котушка реле – це індуктивне навантаження, яке здатне накопичувати певну кількість енергії, коли через нього протікає струм і випромінювати цю енергію у вигляді електричної напруги з оберненою полярністю, коли протікання струму зупиняється. Цей викид при відмиканні реле здатний пошкодити управляючий транзистор, тому для його погашення в схему варто увімкнути діод паралельно контактам котушки реле з полярністю, оберненою живленню. Тимчасове тестування блоку реле з транзисторним драйвером, зібране навісним монтажем, що живиться й управляється струмом з лабораторного блоку живлення, зображено на рисунку 3.7.

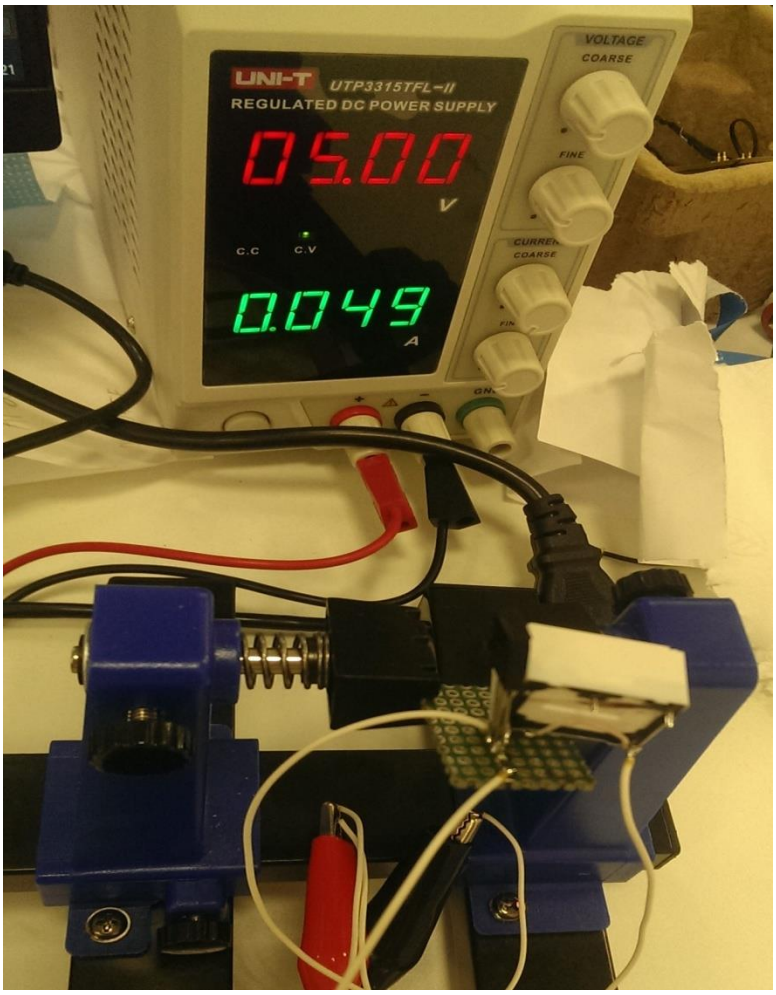


Рисунок 3.7 – Тестування блоку реле на лабораторному БЖ

Також в схему додано світлодіод з обмежуючим резистором для індикації замкнутості контактів реле.

3.2.2 Розробка та налагодження програмної складової пристрою автоматизованого бездротового управління шкільним дзвінком

Знову ж таки, спочатку напишемо програмний код, що дозволить запустити й перевірити правильність взаємодії усіх апаратних модулів пристрою. У блоці setup проводимо конфігурацію виводу управління реле, визначаючи його як вихід. За замовчуванням він стає типу Push-Pull, що означає, що при встановленні логічної одиниці вивід підтягується до аноду живлення, а при встановленні логічного нуля – до катоду. Ну а в функції loop циклічно виконуватимемо переключення станів виводу управління реле з затримкою 500мс. Запрограмувавши мікроконтролер було зафіксовано

систематичні клацання контактів реле без пропусків і «залипань», отже, все працює правильно.

Тепер уже можна перейти до написання повноцінного програмного коду з підключенням до інформаційної системи електронного документообігу. Програма частково перекликається з тою, що писалася для пристрою автоматичної ідентифікації учнів: також підключаються бібліотеки ESP8266WiFi й ESP8266HTTPClient, та відбувається підключення до Wi-Fi мережі у блоці setup. Головна відмінність знаходиться у функції loop, адже тепер пристрій не є ініціатором події на сервері, а навпаки – він циклічно опитує сервер, з частотою 4 рази в секунду, на предмет настання події, що змусить його замкнути контакти реле на 5 секунд для відтворення звукового сигналу дзвінка.

Висновки до розділу

У цьому розділі описано повний список програмних та апаратних технологій, що знадобилися для повноцінного дослідження, розробки та налагодження спряжених з інформаційною системою пристроїв Інтернету речей.

Розділ містить послідовність операцій, що здійснювалися на усіх етапах розробки пристроїв ідентифікації учнів та бездротового управління шкільним дзвінком з описом та розгорнутим обґрунтуванням необхідності їх виконання.

Результатом проведених робіт стало створення повністю функціонуючих пристроїв Інтернету речей, що вже можуть бути використані для автоматизації ще більшої кількості процесів загальноосвітньої школи.

РОЗДІЛ 4. ОПИС РОБОТИ СИСТЕМИ ТА ЗАХОДИ ЩОДО ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ВИКОРИСТАННЯ ПРОГРАМИ

Для того, щоб система не залишалася просто черговою реалізацією хорошої ідеї без майбутнього, необхідно знайти зацікавлену в ній аудиторію. І, щоб забезпечити віддалений доступ користувачів до системи, її необхідно вивантажити, налаштувати та запустити на виконання на підходящому веб-сервері.

Реалізований застосунок є повнофункціональною віддалено доступною інформаційною системою для загальноосвітньої школи, що охоплює функціонал не тільки збору та збереження даних в однонаправленому режимі, але й дозволяє поширювати та модифікувати інформацію з дотриманням усіх норм конфіденційності та правил розподілення доступу.

Перш за все користувачу необхідно потрапити на сторінку авторизації (рис. 4.1). Це можна зробити, перейшовши за посиланням www.newschoolview.somee.com.

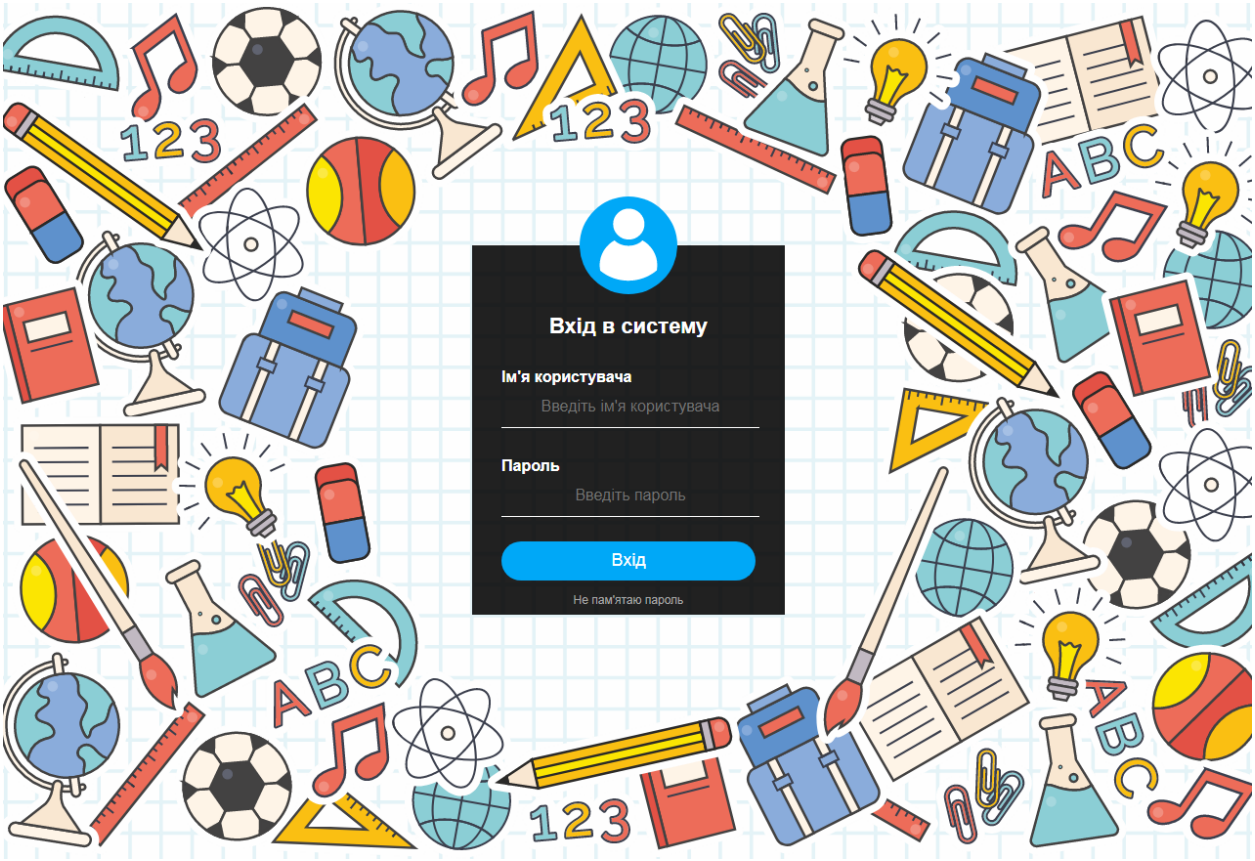


Рисунок 4.1 – Веб-сторінка авторизації системи

Далі необхідно ввести ім’я користувача та пароль, після чого натиснути кнопку «Вхід». У разі успішної ідентифікації та автентифікації, система автоматично перемістить користувача на сторінку, що відповідає його статусу.

4.1 Інструкція користувача статусу Учень/Батько

З точки зору учнів та батьків, головними функціями даної системи є поширення інформації про домашні завдання та розклад уроків, а також інформування користувачів про поточний стан їхньої успішності у різних форматах.

Для перегляду щоденника, учню необхідно у висувному меню зліва обрати пункт «Щоденник» (рис. 4.2).

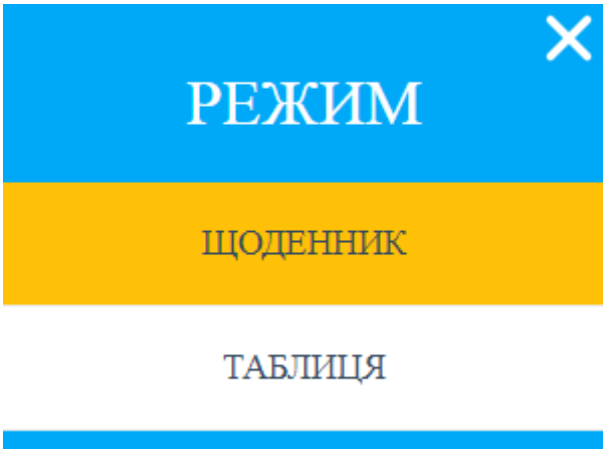


Рисунок 4.2 – Вибір режиму перегляду учнем чи батьком

У навігаційній частині кожної сторінки присутні: кнопка виклику меню та кнопки навігації по суміжних тижнях (місяцях).

Сторінка щоденника (рис. 4.3) розділена на блоки, кожен із яких вміщує таблицю з даними про перелік уроків з нумерацією, назвою предметів, домашніми завданнями та оцінками (вміст останніх двох може бути порожнім). Над кожним таким блоком позначається день тижня та дата.

Завдяки такій структурі, сторінка легко була адаптована під пристрої різних типів та форматів екрану.

Рисунок 4.3 – Веб-сторінка щоденника учня

Наступним режимом є перегляд таблиць успішності учнів (рис. 4.4). Тут учень чи батько може обрати місяць та предмет, з яких він хоче побачити оцінки. Цей вибір робиться за допомогою випадаючих списків, вміст яких формується динамічно. При оновленні вибору зі списку місяців оновлюється й список предметів, адже він може відрізнятися від попереднього.

Таблиця успішності має вигляд горизонтальної таблиці з датами усіх уроків у головному рядку й оцінками (при їх наявності) у другорядному. Колір стовпців вказує на статус дати чи оцінки: звичайна, контрольна робота, тематична оцінка, семестрова, річна чи особлива (оцінка за зошит, вірш тощо).

Нагадування про статус кожного кольору присутнє відразу під таблицею.

Рисунок 4.4 – Веб-сторінка таблиці успішності учня

Таким чином, завдяки інформаційній системі електронного документообігу, кожен учень завжди залишається проінформованим про усі процеси, що стосуються його і тільки його. Адже за новітніми правилами кожен учень класу може отримувати інформацію тільки про свої оцінки, щоб не травмувати нестійку дитячу психіку [40].

4.2 Інструкція користувача статусу Вчитель/Адміністрація

Користувацький інтерфейс вчителя поєднує в собі найважливіші параметри: візуальну простоту, лаконічність та зрозумілість з багатим вбудованим функціоналом.

Завдяки уніфікації багатьох модулів та стилів, система є впізнаваною на кожній сторінці, а головне простою в освоєнні. Таким чином меню викладача виглядає точно так, як і в учня. Змінився лише набір режимів: «Розклад уроків» та «Журнал класу».

«Розклад уроків» також має блочну структуру, але таблиці в середині кожного блоку відрізняються набором стовпців: номер уроку, назва предмету, клас, задане домашнє завдання.

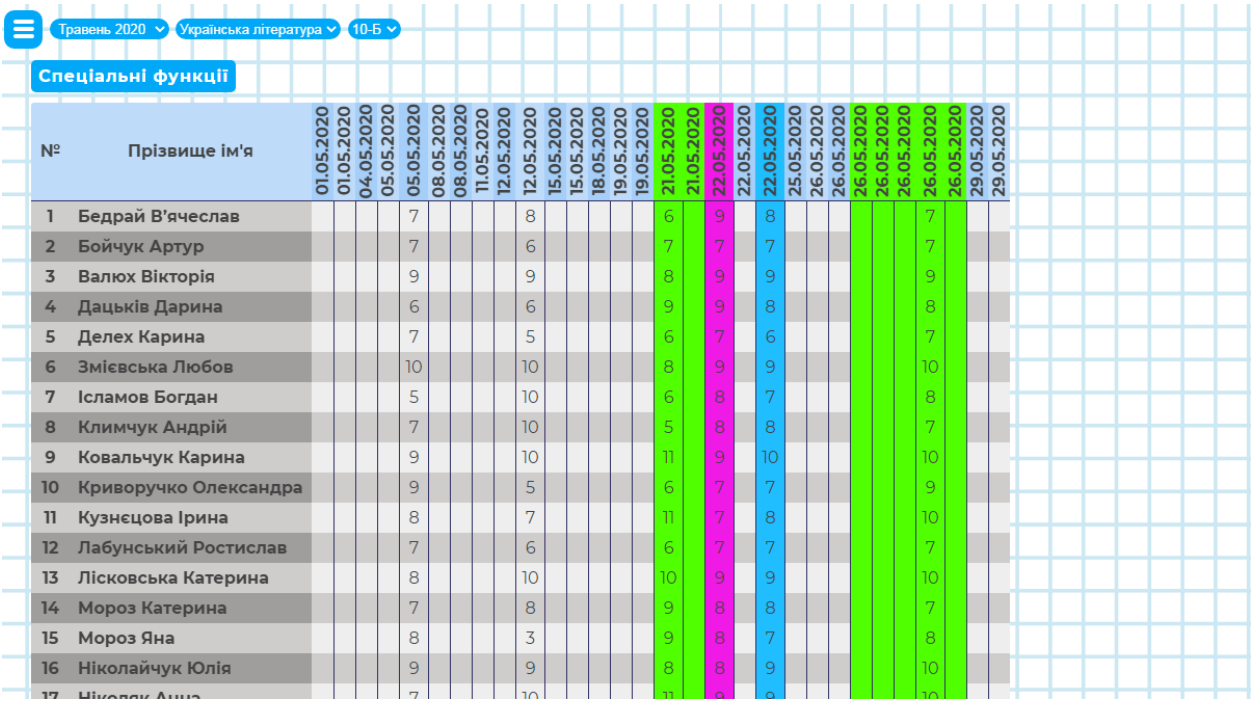
Але, звичайно, найбільше функцій, які має система, розміщуються на сторінці журналу класу (рис. 4.5).

Перш за все варто звернути увагу на рядок з елементами навігації, тут представлено: кнопка виклику меню режимів, випадаючі списки для вибору місяця, предмету та класу, для якого варто відображати головну таблицю з усіма оцінками.

Майже кожна комірка таблиці наділена певним функціоналом. При натисненні на комірку з датою (у першому рядку) буде відкрито модальне вікно, в якому можна ввести дані про урок саме цієї дати: задати тип уроку (звичайний, контрольна робота), задати домашнє завдання та записати тему уроку. Щоб підтвердити ввід необхідно натиснути кнопку «ОК». Для

скасування натиснути «Відміна», або просто клікнути за межами модального вікна.

Після натиснення на будь-яку комірку, що знаходиться на перетині двох осей, з’явиться модальне вікно виставлення оцінки обраному учню в обрану дату.



№	Прізвище ім'я	01.05.2020	01.05.2020	04.05.2020	05.05.2020	05.05.2020	08.05.2020	08.05.2020	11.05.2020	12.05.2020	12.05.2020	15.05.2020	15.05.2020	18.05.2020	19.05.2020	19.05.2020	21.05.2020	21.05.2020	22.05.2020	22.05.2020	22.05.2020	25.05.2020	26.05.2020	26.05.2020	26.05.2020	26.05.2020	26.05.2020	26.05.2020	29.05.2020	29.05.2020
1	Бедрай В'ячеслав					7				8							6		9		8							7		
2	Бойчук Артур					7				6							7		7		7							7		
3	Валюх Вікторія					9				9							8		9		9							9		
4	Дацьків Дарина					6				6							9		9		8							8		
5	Делех Карина					7				5							6		7		6							7		
6	Змієвська Любов					10				10							8		9		9							10		
7	Ісламов Богдан					5				10							6		8		7							8		
8	Климчук Андрій					7				10							5		8		8							7		
9	Ковальчук Карина					9				10							11		9		10							10		
10	Криворучко Олександра					9				5							6		7		7							9		
11	Кузнєцова Ірина					8				7							11		7		8							10		
12	Лабунський Ростислав					7				6							6		7		7							7		
13	Лісковська Катерина					8				10							10		9		9							10		
14	Мороз Катерина					7				8							9		8		8							7		
15	Мороз Яна					8				3							9		8		7							8		
16	Ніколайчук Юлія					9				9							8		8		9							10		

Рисунок 4.5 – Веб-сторінка журналу класу

Також на сторінці присутня кнопка «Спеціальні функції», що викликає модальне вікно з кнопками (рис. 4.6), що дозволяють автоматично вирахувати та виставити тематичні, семестрові, річні оцінки за допомогою вбудованих алгоритмів. А також саме тут присутня кнопка для додавання колонки з «особливим» статусом.

Рисунок 4.6 – Модальне вікно спеціальних функцій

Отже, ця інформаційна система надає користувачам статусу «Вчитель» чи «Адміністрація» великий арсенал можливостей, які призвані спростити життя як самим викладачам, так і учням, адже всі дії вчителів миттєво відображаються в учнів. Це позбавляє необхідності останніх вести звичайний паперовий щоденник та дізнаватися інформацію від третіх осіб про свою успішність при неможливості фізичного відвідування школи.

4.3 Інструкція по використанню пристрою автоматичної ідентифікації учнів/викладачів

Кожен учень чи вчитель, що використовують пристрій автоматичної ідентифікації, змушені мати при собі свою власну, зареєстровану в системі, RFID мітку, яка фізично може бути вмонтована в ID картку, брелок, браслет чи навіть кільце.

Для ідентифікації в системі користувач має піднести свою RFID мітку паралельно до лицевої сторони пристрою на відстань, що не перевищує п’яти сантиметрів.

Після проходження часу утримання мітки, що не перевищує трьох секунд (залежить від сили сигналу Wi-Fi мережі, де знаходиться пристрій, та завантаженості сервера інформаційної системи), на лицевій панелі в лівому верхньому куті повинен на одну секунду засвітитися світлодіод. Зелений колір свідчить про успішну ідентифікацію, а червоний про помилку знаходження ідентифікатора цієї мітки в системі, що може бути спричинене неякісним зчитуванням, тому варто спробувати ще раз.

4.4 Заходи щодо забезпечення безпеки використання системи

Звичайно, така серйозна інформаційна система, що містить у собі величезні масиви персональних даних, потребує належної системи захисту від атак зловмисників.

Першим етапом захисту є хешування паролів з додаванням «солі». Це запобігає передачі мережею паролю та подальше його зберігання у чистому вигляді.

Другим етапом є використання на етапі ідентифікації та автентифікації не звичайного запиту, що виступає у вигляді звичайного рядка з усіма необхідними аргументами у середині, а спеціального, аргументи в який передаються окремими параметрами. Цей крок призваний захистити систему від SQLін’єкції.

Третім етапом є перевірка при кожному запиті до сервера, чи дані, що бажає отримати клієнтська сторона належать ідентифікованому в системі користувачу. Цей крок призваний не дозволити зловмиснику, який заволодів обліковим записом одного із користувачів й потрапив у систему, отримати дані, що виходять за рамки прав ідентифікованого користувача.

Також до способів захисту варто віднести JWT-токени, що автоматично стають невалідними після проходження 15хв від останньої активності на сайті. А також регулярні резервні копії бази даних, що робляться сервером також автоматично.

Висновки до розділу

Розділ містить повний набір інструкцій для користувачі різних типів. Описано та проілюстровано алгоритми доступу до розкладу занять та журналу вчителя та адміністратора, а також послідовність кроків для відкриття щоденника й таблиць успішності учня чи батька. Ще в розділі описано інструкцію по взаємодії з пристроєм автоматичної ідентифікації, що є спільною як для учнів так і для викладачів.

Важливим елементом наповнення цього розділу є опис заходів що були здійснені для забезпечення безпеки використання розробленої інформаційної системи електронного документообігу.

ВИСНОВОК

Ведення документації лише в паперовому вигляді з кожним днем відштовхує сучасну систему освіти на декілька кроків назад, адже не сприяє оптимізації часу на роботу з документами, а навпаки його тільки забирає. Також непередбачувана епідеміологічна ситуація в світі показує, що методи контактної взаємодії з документами можуть бути не лише ускладненими в екстремальних ситуаціях, а й навіть небезпечними. Тому заклади загальної середньої освіти несамовито потребують інформаційної системи електронного документообігу.

В ході виконання завдань кваліфікаційної роботи було досліджено теоретичні основи побудови інформаційної системи електронного документообігу, проаналізовано програмні та технічні рішення подібних уже існуючих систем, що надало розуміння принципів побудови інформаційних систем, а також показало напрямок руху в галузі досягнення досконалості власної системи, адже всі існуючі мають ті чи інші недоліки: мають обмежений функціонал, володіють недостатньо простим і зрозумілим інтерфейсом чи взагалі є платними. Щоб спробувати уникнути всі вище зазначені недоліки, було ретельно спроектовано, реалізовано та протестовано систему, застосовуючи наступні технології: HTML, CSS, JavaScript, jQuery, AJAX, ASP.NET MVC Framework, Microsoft SQL Server, Google Chrome Developer Tools тощо. Визначено, що система відповідає усім закладеним вимогам та дозволяє зручно зберігати, модифікувати та поширювати інформацію, а також заощаджує людський час, адже виконує частину рутинних задач в автоматизованому режимі.

Реалізована система зарекомендувала себе як багатофункціональна, зручна, проста в налаштуванні та використанні. Вона матиме велике практичне значення для закладів, де буде використовуватися, адже системи подібного призначення є незамінною складовою будь-якої сучасної загальноосвітньої школи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Передпроектне обстеження при розробці інформаційної системи. *Habr*: веб-сайт. URL: https://habr.com/ru/post/432844/.
2. Дакетт Д. HTML и CSS. Разработка и дизайн веб-сайтов / Д. Дакетт. – Москва: Эксмо, 2019. – 480 с. – (Мировой компьютерный бестселлер).
3. ДакеттД. Javascriptи jQuery. Интерактивная веб-разработка / Д. Дакетт. – Москва: Эксмо, 2018. – 640 с. – (Мировой компьютерный бестселлер).
4. Васильев А. Н. Программирование на C# для начинающих / А. Н. Васильев. – Москва: Бомбора, 2018. – 592 с. – (Российский компьютерный бестселлер).
5. Кариев Ч. А. Технология Microsoft ADO .NET. Учебное пособие / Ч. А. Кариев. – Москва: Бином. Лаборатория знаний, 2017. – 543 с. – (Основы информационных технологий).
6. Белов А. Программирование ARDUINO. Создаем практические устройства / А. Белов. – Санкт Петербург: Наука и Техника, 2018. – 272 с.
7. Чим відрізняється розробка ASP.NET MVC від ASP.NET API. *Habr Q&A*: веб-сайт. URL: https://qna.habr.com/q/254139.
8. HTML. *Вікіпедія*: веб-сайт. URL: https://uk.wikipedia.org/wiki/HTML.
9. Простою мовою про HTTP. *Habr*: веб-сайт. URL: https://habr.com/ru/post/215117/.
10. CSS. *Вікіпедія*: веб-сайт. URL: https://uk.wikipedia.org/wiki/CSS.
11. CSS Selector Reference. *W3Schools:* веб-сайт. URL: https://www.w3schools.com/cssref/css_selectors.asp.
12. Як працює CSS Flexbox. *Tproger*: веб-сайт. URL: https://tproger.ru/translations/how-css-flexbox-works/.
13. JavaScript. *Вікіпедія*: веб-сайт. URL: https://uk.wikipedia.org/wiki/JavaScript.

14. jQuery. *Вікіпедія*: веб-сайт. URL: https://uk.wikipedia.org/wiki/JQuery.
15. AJAX. *Вікіпедія*: веб-сайт. URL: https://uk.wikipedia.org/wiki/AJAX.
16. ASP.NET Core WEB API. Введення в WEB API. *Metanit*: веб-сайт. URL: https://metanit.com/sharp/aspnet5/23.1.php.
17. Вивчаємо ASP.NET MVC 5. *Professorweb*: веб-сайт. URL: https://professorweb.ru/my/ASP_NET/mvc/level1/.
18. Платформа ASP.NET. *Metanit*: веб-сайт. URL: https://metanit.com/sharp/mvc.php.
19. C Sharp. *Вікіпедія*: веб-сайт. URL: https://uk.wikipedia.org/wiki/C_Sharp.
20. ADO.NET. *Вікіпедія*: веб-сайт. URL: https://uk.wikipedia.org/wiki/ADO.NET.
21. Введення в MS SQL Server і T-SQL. Що таке SQL Server і T-SQL? *Metanit*: веб-сайт. URL: https://metanit.com/sql/sqlserver/1.1.php.
22. SQL. *Вікіпедія*: веб-сайт. URL: https://ru.wikipedia.org/wiki/SQL.
23. Профілактика SQL-ін'єкцій. *Habr*: веб-сайт. URL: https://habr.com/ru/post/87872/.
24. З чого складається IoT. *Habr*: веб-сайт. URL: https://habr.com/ru/post/436708/.
25. Публікація на веб-сервері. *Metanit*: веб-сайт. URL: https://metanit.com/sharp/mvc/13.2.php.
26. Збережені функції. За і проти. *Habr*: веб-сайт. URL: https://habr.com/ru/post/210920/.
27. Оптимізація збережених процедур в SQL Server. *Habr*: веб-сайт. URL: https://habr.com/ru/post/85370/.
28. C++. *Вікіпедія*: веб-сайт. URL: https://uk.wikipedia.org/wiki/C%2B%2B.
29. Вбудована система. *Вікіпедія*: веб-сайт. URL: https://uk.wikipedia.org/wiki/%D0%92%D0%B1%D1%83%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0.

30. Inretnet of Things: все, що потрібно знати про Інтернет речей і про майбутнє сучасної цивілізації. Everest: веб-сайт. URL: https://www.everest.ua/ru/internet-of-things-vse-chto-nuzhno-znat-ob-ynternete-veshhej-y-o-budushhem-sovremennoj-czyvylyzaczyy.
31. Мультиметр. *Вікіпедія*: веб-сайт. URL: https://en.wikipedia.org/wiki/Multimeter.
32. Осцилограф. *Вікіпедія*: веб-сайт. URL: https://uk.wikipedia.org/wiki/%D0%9E%D1%81%D1%86%D0%B8%D0%BB%D0%BE%D0%B3%D1%80%D0%B0%D1%84.
33. Логічний аналізатор. *Вікіпедія*: веб-сайт. URL: https://en.wikipedia.org/wiki/Logic_analyzer.
34. UART. *Вікіпедія*: веб-сайт. URL: https://uk.wikipedia.org/wiki/UART.
35. SPI. *Вікіпедія*: веб-сайт. URL: https://uk.wikipedia.org/wiki/Serial_Peripheral_Interface.
36. Початок роботи з ESP8266 NodeMCU v3 Lua з Wi-Fi. *Arduini Master:* веб-сайт. URL: https://arduinomaster.ru/platy-arduino/esp8266-nodemcu-v3-lua/.
37. Радіочастотна ідентифікація. *Вікіпедія*: веб-сайт. URL: https://en.wikipedia.org/wiki/Radio-frequency_identification.
38. RFID-модуль RC522. *3DiY*: веб-сайт. URL: https://3d-diy.ru/wiki/arduino-moduli/rfid-modul-rc522.
39. ESP8266 прошивка, програмування в Arduino IDE. *Habr*: веб-сайт. URL: https://habr.com/ru/post/371853/.
40. Шкільні зміни: теорія і практика. Коментар до змін в початковій школі психолога Катерини Гольцберг [Електронний ресурс] // Міністерство освіти і науки України. – 2016. – Режим доступу до ресурсу: https://mon.gov.ua/ua/osvita/zagalna-serednya-osvita/pochatkovashkola/poradi-psihologa/shkilni-zmini-teoriya-i-praktika-kometar-do-zmin-v-pochatkovij-shkolipsihologa-katerini-golcberg.

ДОДАТКИ

ДОДАТОК А

Контролери системи з переліком усіх вбудованих функцій

1) HomeController – містить усі методи для серверної валідації введених користувацьких полів на сторінці авторизації. Також саме тут проходять процеси ідентифікації та автентифікації, після успішного проходження яких створюється сесія та переміщення користувача на одну з сторінок теки View, сторінка визначається по результатам отриманого статусу користувача із БД.
1) PupilController – зберігає методи що можуть викликатися тільки зі сторінки авторизованого користувача статусу «Учень/Батько». Методи контролера, що викликаються напряму користувачем: DisplayDayBook, DisplayPageLeft, DisplayPageRight, GetScriptWithPupilTable, OnChangeInDataSelect. Допоміжні методи: DisplayDayBookOnWeek, CheckWorkDay, CheckNextAndPrevWeeks, FindNearestDate, GetScriptWithDayBook, DisplayPupilTableOnMonth, FillDateSelect, FillSubjectSelect, ScriptForBackgroundColourOfMarks, ScriptForErasingAllNavigationTools.
2) TeacherController – тут знаходяться методи, які можуть викликатися користувачем статусу «Вчитель/Адміністрація». Методи контролера, що викликаються з клієнтської сторони: DisplayForTeacher, DisplayPageLeft, DisplayPageRight, GetScriptWithTeacherTable, OnChangeInDateSelect, OnChangeInSubjectSelect, ScriptForLessonMenu, ScriptForMarkMenu, ScriptForSetMarkMenu, ScriptForSetLessonMenu, ScriptForSpecialBtns. Допоміжні методи: DisplayDayBookOnWeek, FindNearestDate, GetScriptWithDayBook, FillDateSelect, FillSubjectSelect, FillClassPartSelect, ScriptForSetTematichna, ScriptForSetSemestrova, ScriptForSetRichna, ScriptForSetSpecialMark.