order number: ....

Project report submitted to the

UNIVERSITY OF MOHAMED BOUDIAF – MSILA

1985

جامعة محمد بوضياف - المسيلة

University of Mohamed Boudiaf-Msila

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

In partial fulfillment of the requirements for the degree of

License in **Computer science**

By

Oussama Boubaaya

Imad Baadji

Mouad Amran

Title of the project

FACULTY PLANNING/SCHEDULING PLATFORM

Under the supervision of

Dabba Ali

Acknowledgments

Praise be to God, abundant and blessed praise. Now then, this may be our last year at this university. We would like to extend our sincere thanks to our esteemed professors for the effort they have put into providing us with knowledge. Let us not forget our supervising professor, Dabba Ali, who has satiated us with his guidance. May God protect and care for him, and may God bless you all.

Contents

List of figures

General Introduction

With the rapid technological development witnessed worldwide, information management and scheduling systems have become essential elements that contribute to improving the quality of administrative and educational organization. Scheduling is a task that requires high precision and tight coordination, especially in educational institutions that rely on the arrival of materials and professors at specific times to ensure a smooth and effective educational process.

With the increasing number of professors and educational materials, the need for automated digital solutions emerges that help organize schedules more efficiently and accurately, reduce errors resulting from manual intervention, and save the time and effort spent preparing schedules manually. Here, we leveraged our faculty as a test subject to develop this project, which is a web application called “**StaffWise”** that manages schedule creation based on data entered by administrators, such as professor’s free time, the number of days allowed for teaching, and the types of subjects they teach, or an Excel file containing all the data.

The report's methodology is structured into three chapters:

Chapter One: A comprehensive introductory chapter that presents basic concepts and an overview of the project.

Chapter Two: This chapter presents the design and modeling of the most important diagrams that illustrate the implementation process.

Chapter Three: Introduction to the most important tools and programming languages used to complete the project.

Chapter 1: Introduction and Project Overview

1. Introduction

In the digital age, educational institutions face increasing demands to streamline administrative processes while maintaining high standards of efficiency and accuracy. Information management systems have emerged as critical tools for achieving these goals, transforming how data is stored, analyzed, and utilized. These systems not only enhance decision-making but also reduce redundancies and errors inherent in manual workflows. In educational contexts, effective management of resources—such as faculty, classrooms, and course materials—is pivotal to delivering a seamless academic experience.

2. The Critical Role of Scheduling

Scheduling serves as the backbone of operational efficiency in educational institutions. It involves coordinating multiple variables: professor availability, classroom allocation, course timings, and student enrollment. Traditional manual scheduling methods, often reliant on spreadsheets or physical planners, struggle to accommodate the growing complexity of modern academic environments. Challenges such as overlapping sessions, underutilized resources, and last-minute changes underscore the limitations of these approaches. Errors in scheduling can disrupt teaching activities, strain faculty workloads, and ultimately compromise educational outcomes.

2.1. Challenges of Manual Scheduling Systems

Manual scheduling is labor-intensive, time-consuming, and prone to human error. Administrators must reconcile conflicting constraints, such as professors’ availability, subject-specific requirements, and institutional policies. For instance, a professor teaching advanced courses may have limited time slots due to research commitments, while laboratory-based subjects require specialized rooms. Without automation, resolving these conflicts becomes a tedious trial-and-error process. Additionally, scaling manual methods to accommodate larger institutions or dynamic academic calendars is impractical, leading to inefficiencies and frustration.

2.2. The Need for Automated Scheduling Solutions

To address these challenges, automated scheduling systems leverage algorithms and digital tools to optimize resource allocation. Such systems minimize human intervention, reduce errors, and adapt to changing requirements. By automating repetitive tasks, institutions can reallocate time and effort toward strategic initiatives, such as curriculum development or student engagement. Furthermore, digital solutions enable real-time updates, ensuring schedules remain accurate and responsive to unforeseen disruptions.

2.3. Introducing StaffWise: A Modern Scheduling Solution

This project proposes **StaffWise**, a web-based application designed to revolutionize scheduling in educational institutions. StaffWise empowers administrators to generate optimized schedules by inputting criteria such as:

- Professors’ available time slots.
- Maximum teaching days per faculty member.
- Subject-specific requirements (e.g., lab sessions, lecture halls).
- Bulk data uploads via Excel files for scalability.

The application’s algorithms prioritize fairness, resource efficiency, and adherence to institutional policies. By automating the scheduling process, StaffWise aims to eliminate bottlenecks, enhance transparency, and ensure alignment with academic goals.

3. Oreview

This chapter has outlined the foundational concepts driving the development of StaffWise, emphasizing the transition from manual to automated systems in education. Subsequent chapters will delve into the technical design, implementation methodologies, and tools employed to bring this solution to fruition. Chapter Two explores the system’s architecture through UML diagrams and workflow models, while Chapter Three details the programming languages, frameworks, and technologies underpinning the application.

This structured approach ensures a comprehensive understanding of both the theoretical and practical aspects of building an effective scheduling system tailored to modern educational needs.

Chapter 2: Design

1. Introduction

After completing the theoretical study of how to establish a time management system, in this chapter we will explain the steps we will follow to design our website in detail. We will use the Unified Modeling Language (UML) to ensure a logical and comprehensive visualization of the system. We will define its functional requirements and accurately document all the details and information necessary for its design. We will also conduct a detailed study of all phases of the implementation and carefully analyze a set of related plans.

2. UML Diagrams

Unified Modeling Language (UML) is a standardized general-purpose modeling language in software engineering that provides a set of graphical notation techniques to create visual models of object-oriented systems. It is used to specify, visualize, construct, and document the artifacts of software systems.

We'll be focusing on three key diagrams: the Class Diagram, Authentication Sequence Diagram, and Use Case Diagram. These diagrams collectively illustrate the system's structure, workflows, and user interactions.

2.1. Class Diagram

This class diagram represents a faculty scheduling and management system. The system manages users, faculties, departments, courses, and scheduling resources.

Core Entity Structure

Administrative Hierarchy:

- Faculty - Represents an academic faculty/school with properties like opening/closing times
- Department - Academic departments within a faculty
- Section - Subdivisions within departments (likely academic levels/years)
- StudentGroup - Groups of students (like classes or cohorts) within sections

Course Management:

- Subject - Academic subjects offered by faculties
- Course - Actual course instances with scheduling information
- Semester - Academic terms with start/end dates for planning

Resources:

- Room - Physical spaces used for courses
- Timeslot - Time periods when courses can be scheduled

User Management:

- User - System users with assigned roles (teachers, administrators)
- EmailVerification - For verifying user email addresses
- ResetCode - For handling password resets
- InvalidatedToken - to blacklist expired tokens

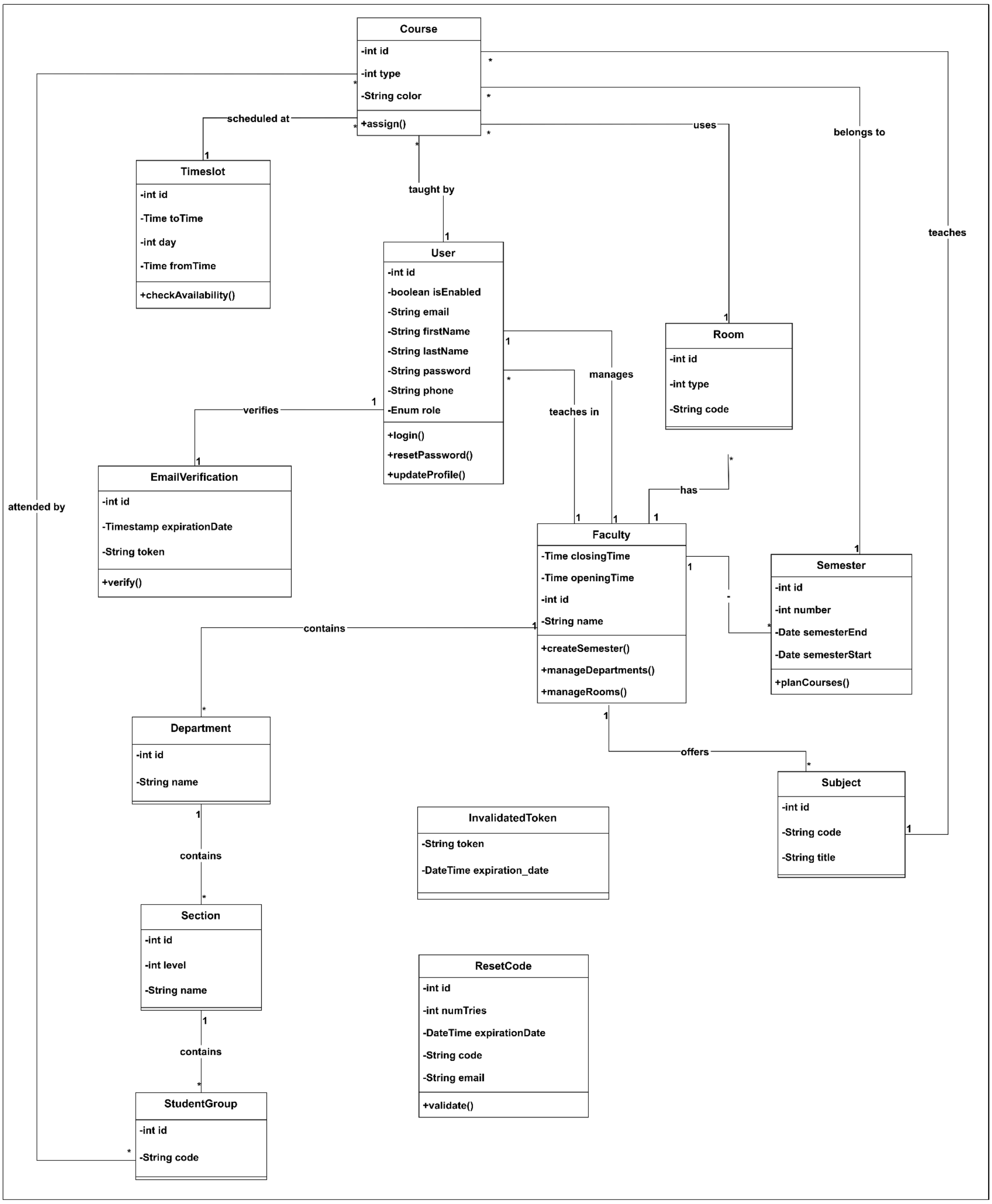


figure1. Class Diagram

2.2. Use Case Diagram

The use case diagram identifies the different actors in your system (Admin, Teacher, New User) and the functionality available to each. It highlights the core features of StaffWise like user management, course scheduling, and reporting.

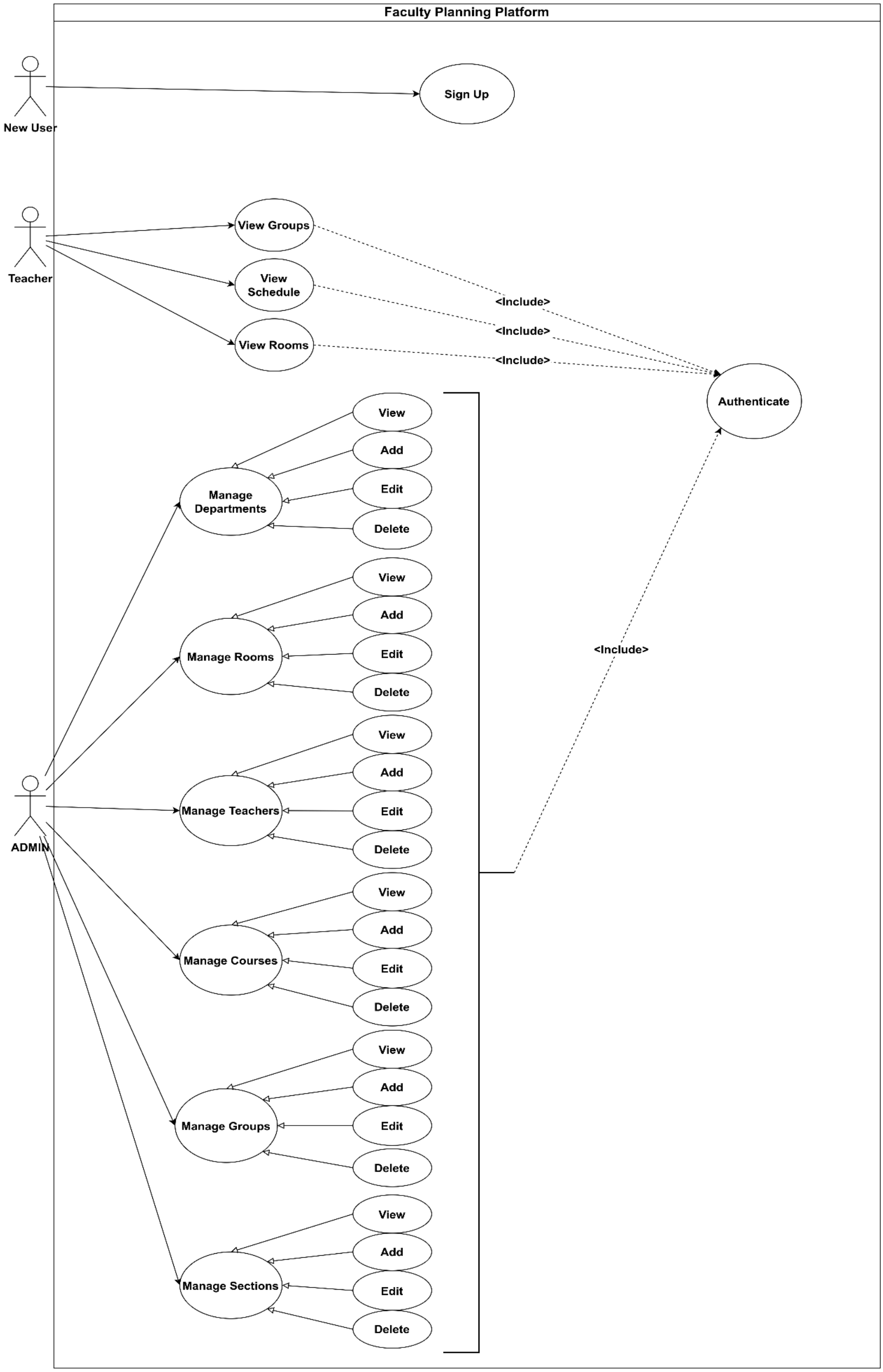


figure2. Use Case Diagram

2.3. Authentication Sequence Diagram

The Authentication sequence diagram shows the login process where:

- User enters credentials
- System verifies those credentials
- If valid, JWT token is generated and user is redirected to dashboard
- If invalid, an error message is displayed

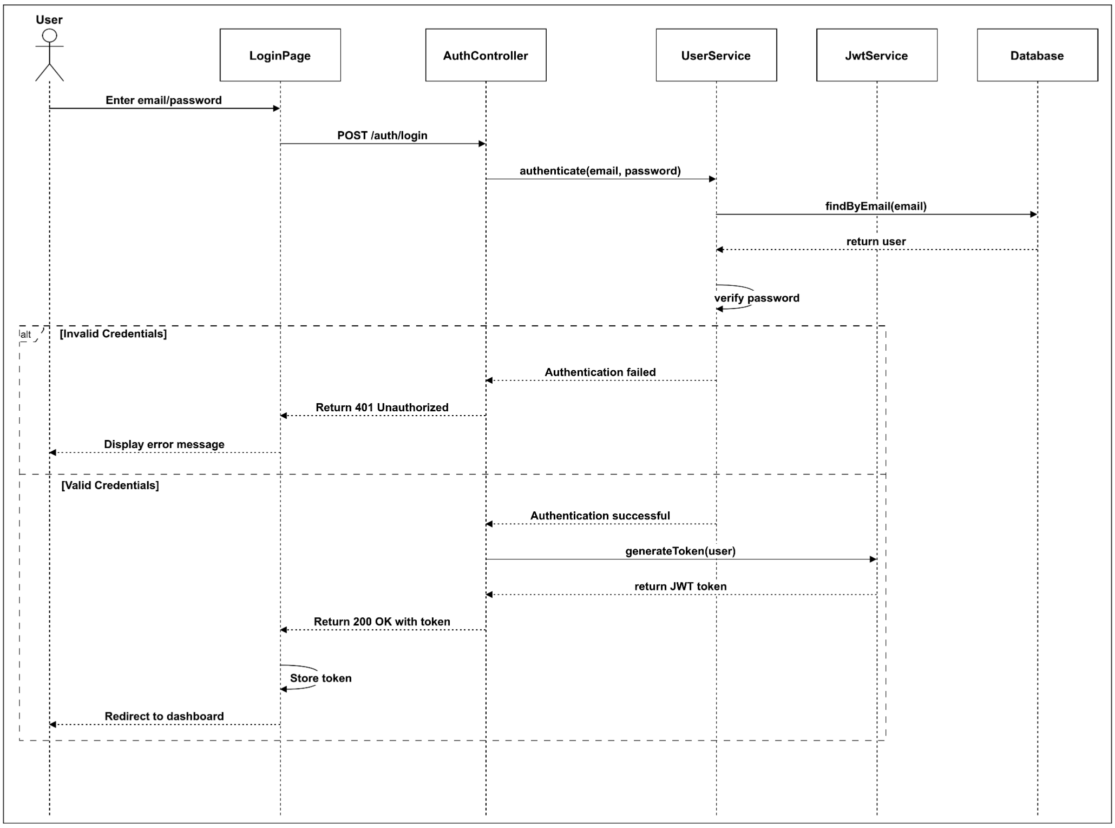


figure3. Authentication Sequence Diagram

2.4. Activity Diagram

The activity diagram shows all possible user activities inside the application including the authentication process and role-based actions.

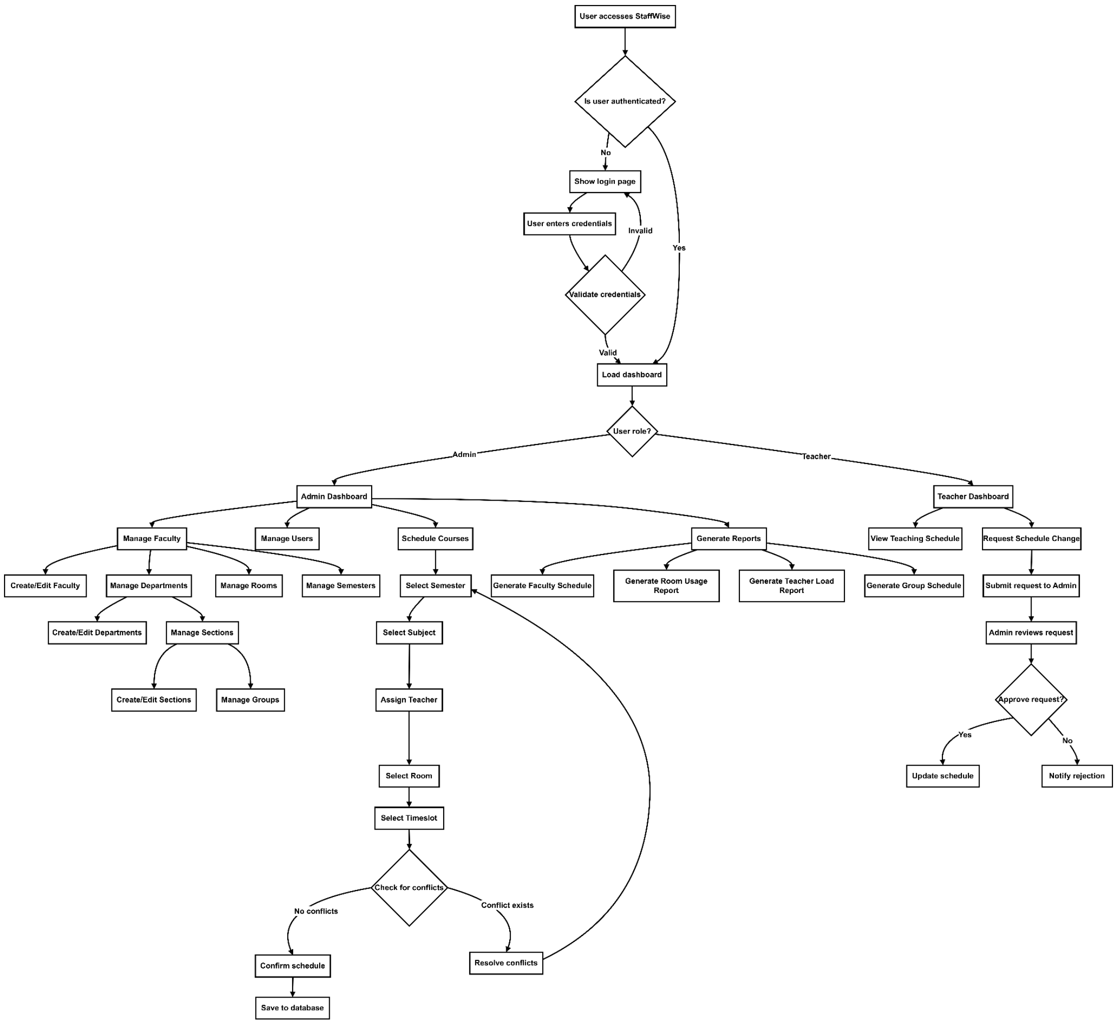


figure4. Activity Diagram

3. Oreview

The class diagram ensures data integrity through normalized relationships, while the sequence diagram prioritizes security via JWT. The use case diagram emphasizes role-based access, aligning with StaffWise's goal of reducing manual effort and ensuring transparency. Together, these diagrams provide a blueprint for scalable, maintainable, and user-centric scheduling automation.

Chapter 3: Development

1. Introduction

After completing the theoretical study and design phases in the first and second chapters, we will move on to the actual implementation phase of the project. This chapter details the technology stack, frameworks, and tools employed to develop **StaffWise**, aligning with the design principles outlined in Chapter Two. We will review the work environment in which the website was developed. We will discuss the database we created, as well as the interfaces we designed and will develop in the future. We will also highlight the general features of the website.

2. Technical Stack

1. Backend: Java with Spring Boot
2. Frontend: Next js, Tailwind css
3. Database: MySQL , XAMPP, phpMyAdmin
4. Authentication: JWT-based token authentication

2.1. Backend Implementation

Spring Boot Framework

Role: Powers the business logic, authentication, and API endpoints.

Key Features:

Spring Security: Manages JWT-based authentication and role-based access control (Admin vs. Teacher).

Spring Data JPA: Simplifies database operations via ORM (Object-Relational Mapping).

REST Controllers: Handle HTTP requests (e.g., AuthController, UserController).

API Endpoints examples:

POST /auth/login: Authenticates users and returns JWT.

PATCH /courses/assign: Assigns a room and timeslot to a course, implements conflict detection.

GET /faculties/{id}/users: Retrieves the given faculty’s Professors.

2.2. Database Design with MySQL

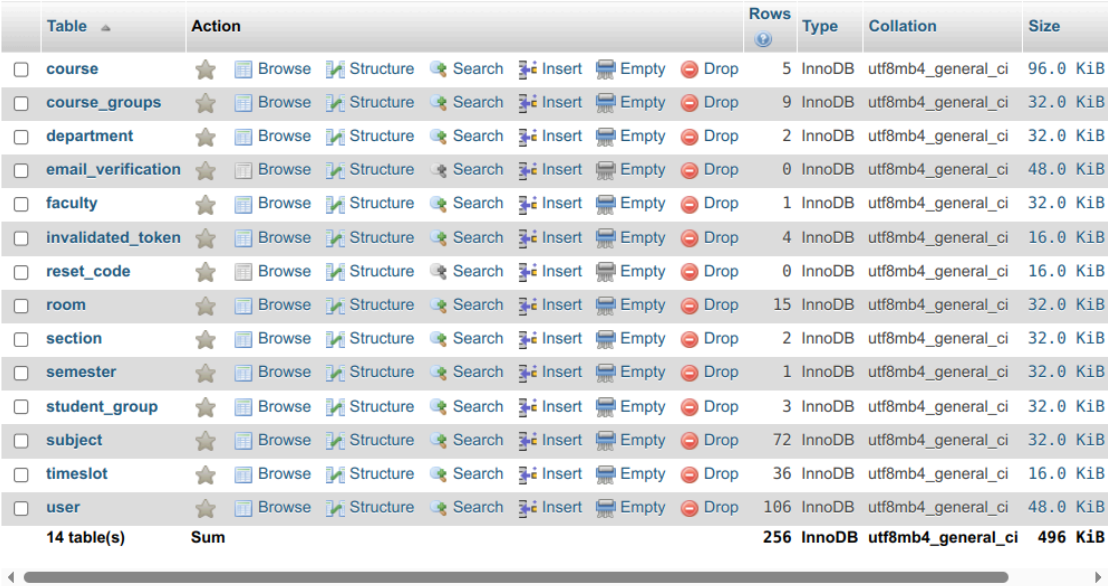


	Table	Action							Rows	Type	Collation	Size
□	course	★	Browse	Structure	Search	Insert	Empty	Drop	5	InnoDB	utf8mb4_general_ci	96.0 KiB
□	course_groups	★	Browse	Structure	Search	Insert	Empty	Drop	9	InnoDB	utf8mb4_general_ci	32.0 KiB
□	department	★	Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	utf8mb4_general_ci	32.0 KiB
□	email_verification	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB
□	faculty	★	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	utf8mb4_general_ci	32.0 KiB
□	invalidated_token	★	Browse	Structure	Search	Insert	Empty	Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB
□	reset_code	★	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB
□	room	★	Browse	Structure	Search	Insert	Empty	Drop	15	InnoDB	utf8mb4_general_ci	32.0 KiB
□	section	★	Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	utf8mb4_general_ci	32.0 KiB
□	semester	★	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	utf8mb4_general_ci	32.0 KiB
□	student_group	★	Browse	Structure	Search	Insert	Empty	Drop	3	InnoDB	utf8mb4_general_ci	32.0 KiB
□	subject	★	Browse	Structure	Search	Insert	Empty	Drop	72	InnoDB	utf8mb4_general_ci	32.0 KiB
□	timeslot	★	Browse	Structure	Search	Insert	Empty	Drop	36	InnoDB	utf8mb4_general_ci	16.0 KiB
□	user	★	Browse	Structure	Search	Insert	Empty	Drop	106	InnoDB	utf8mb4_general_ci	48.0 KiB
	14 table(s)	Sum							256	InnoDB	utf8mb4_general_ci	496 KiB

figure5. Database Tables In phpMyAdmin

2.3. Frontend Implementation

2.3.1. Next.js Framework:

Role: Serves as the foundation for building the user interface, offering server-side rendering (SSR) and static site generation (SSG) for optimized performance.

Key Features:

- Dynamic routing for pages like /dashboard, /timetable, and /settings.
- API routes to proxy requests to the Spring Boot backend.
- State management using React Context API for global data (e.g., user roles, faculty id).

2.3.2. Tailwind CSS:

Role: A utility-first CSS framework for rapid UI development.

Advantages:

- Responsive design for cross-device compatibility.
- Pre-styled components (e.g., tables, forms) for consistency.
- Custom themes to align with institutional branding.

2.3.3. Web App Interface Screenshots:

In this section we review the main interfaces of the StaffWise system. The interfaces are designed to be user-friendly and to serve the various target groups—administrators and professors alike.

1. Sign-In Interface

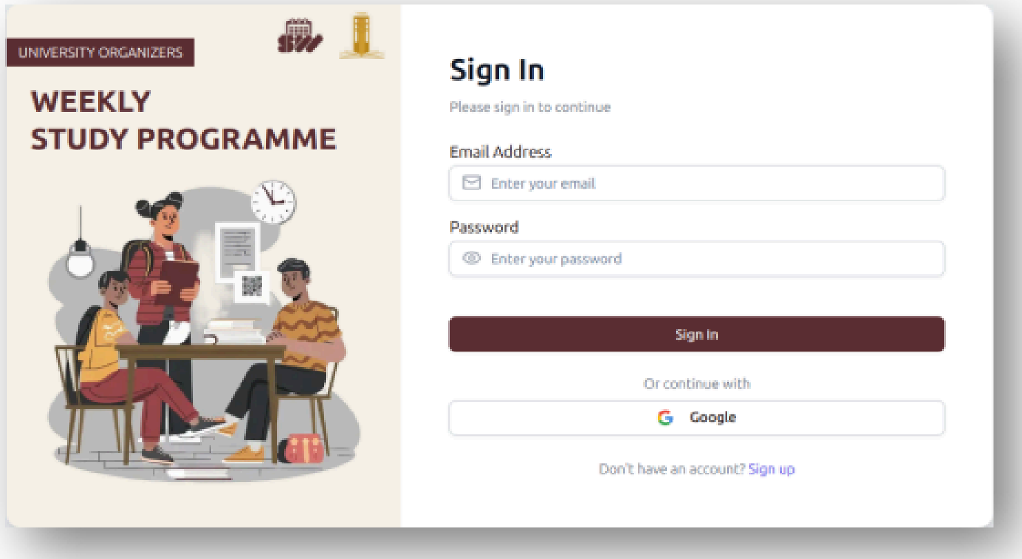


figure6. Sign-In Page

Purpose:

This interface is used to authenticate users, whether professors or administrators, by entering their email and password, or by using their Google account.

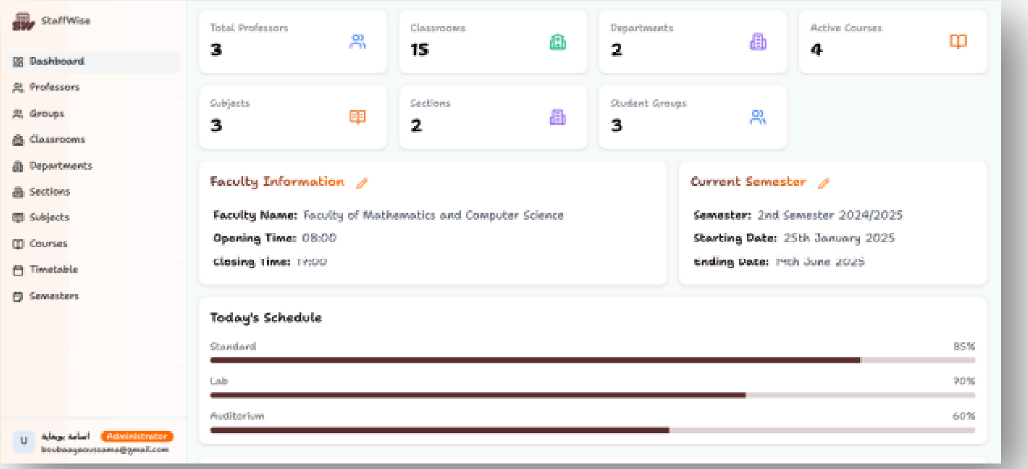
Description:

Displays fields for entering email and password, plus buttons for signing in with Google or creating a new account.

2. Dashboard

Purpose:

Provides an overview of the system's key academic resources and metrics related to the current semester, such as the number of professors, rooms, and subjects, along with occupancy statistics.



Total Professors 3

Classrooms 15

Departments 2

Active Courses 4

Subjects 3

Sections 2

Student Groups 3

Faculty Information

Faculty Name: Faculty of Mathematics and Computer Science
Opening Time: 08:00
Closing Time: 17:00

Current Semester

Semester: 2nd Semester 2024/2025
Starting Date: 25th January 2025
Ending Date: 14th June 2025

Today's Schedule

Standard 85%
Lab 70%
Auditorium 60%

figure7. Dashboard

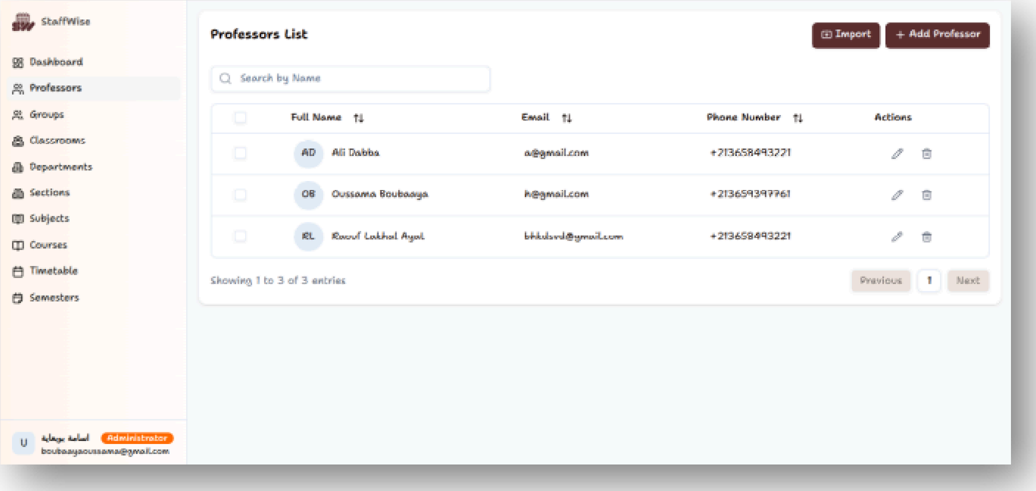
3. Professors Management (Professors List)

Purpose:

Allows the administrator to add, edit, or delete professor records.

Description:

A table lists each professor’s Name, Email, and Phone Number. An “Add Professor” button opens a form for entering a new professor’s details.



Professors List

□	Full Name ⇅	Email ⇅	Phone Number ⇅	Actions
□	AD Ali Dabba	a@gmail.com	+213658493221	✎ 🗑
□	OB Oussama Boubaaya	h@gmail.com	+213659397761	✎ 🗑
□	RL Raouf Lakhal Ayat	bhkdsvd@gmail.com	+213658493221	✎ 🗑

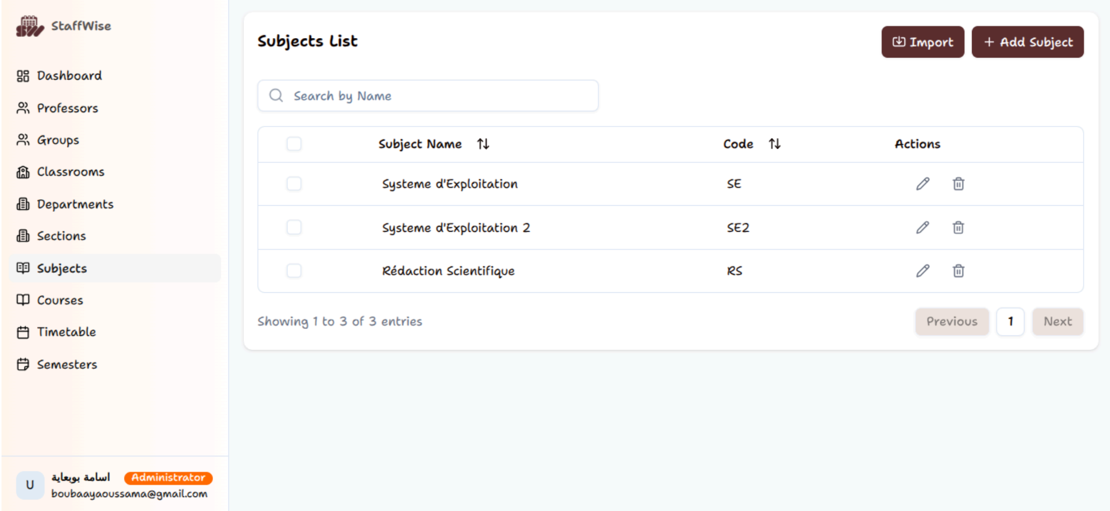
Showing 1 to 3 of 3 entries

figure8. Professors Page

4. Subjects Management (Subjects List)

Purpose:

Used to register academic subjects and link them to the appropriate departments, with the option to designate each subject as theoretical or practical.



Subjects List

□	Subject Name ⇅	Code ⇅	Actions
□	Systeme d'Exploitation	SE	✎ 🗑
□	Systeme d'Exploitation 2	SE2	✎ 🗑
□	Rédaction Scientifique	RS	✎ 🗑

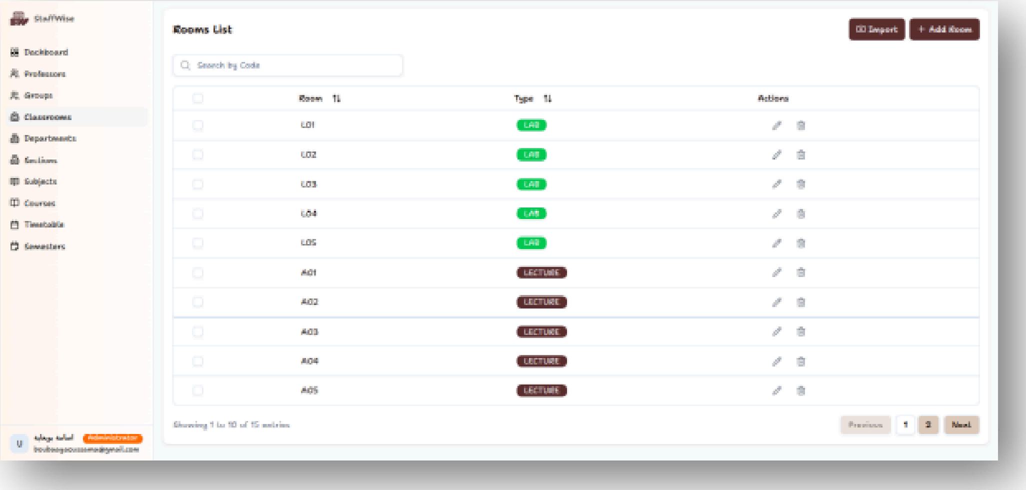
Showing 1 to 3 of 3 entries

figure9. Subjects Page

5. Classrooms Management

Purpose:

Used to allocate and categorize classrooms, whether lecture halls or laboratories.



Rooms List

□	Room ⇅	Type ⇅	Actions
□	L01	LAB	✎ 🗑
□	L02	LAB	✎ 🗑
□	L03	LAB	✎ 🗑
□	L04	LAB	✎ 🗑
□	L05	LAB	✎ 🗑
□	A01	LECTURE	✎ 🗑
□	A02	LECTURE	✎ 🗑
□	A03	LECTURE	✎ 🗑
□	A04	LECTURE	✎ 🗑
□	A05	LECTURE	✎ 🗑

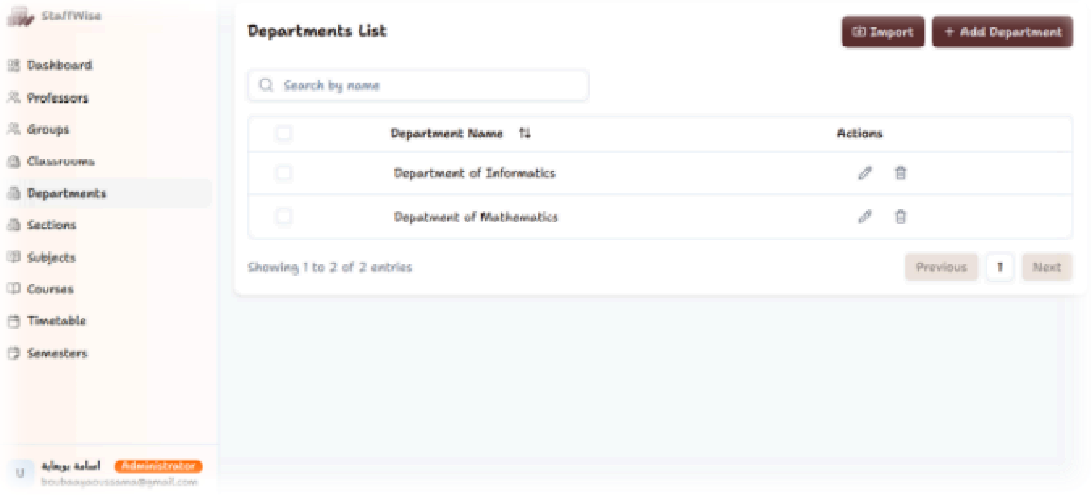
Showing 1 to 10 of 15 entries

figure10. Classrooms Page

6. Departments Management (Departments List)

Purpose:

Organizes the institution’s academic structure by registering its various departments.



Departments List

□	Department Name ⇅	Actions
□	Department of Informatics	✎ 🗑
□	Depatment of Mathematics	✎ 🗑

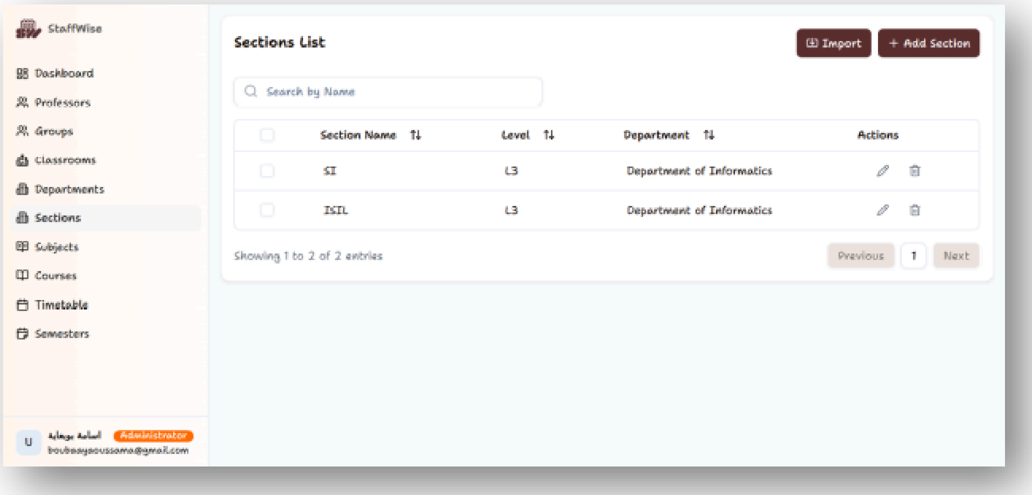
Showing 1 to 2 of 2 entries

figure11. Departments Page

7. Sections Management (Sections List)

Purpose:

Enables the user to manage class sections, including naming them and linking each to its academic level and department.



Sections List

□	Section Name ⇅	Level ⇅	Department ⇅	Actions
□	SI	L3	Department of Informatics	✎ 🗑
□	ISIL	L3	Department of Informatics	✎ 🗑

Showing 1 to 2 of 2 entries

figure12. Sections Page

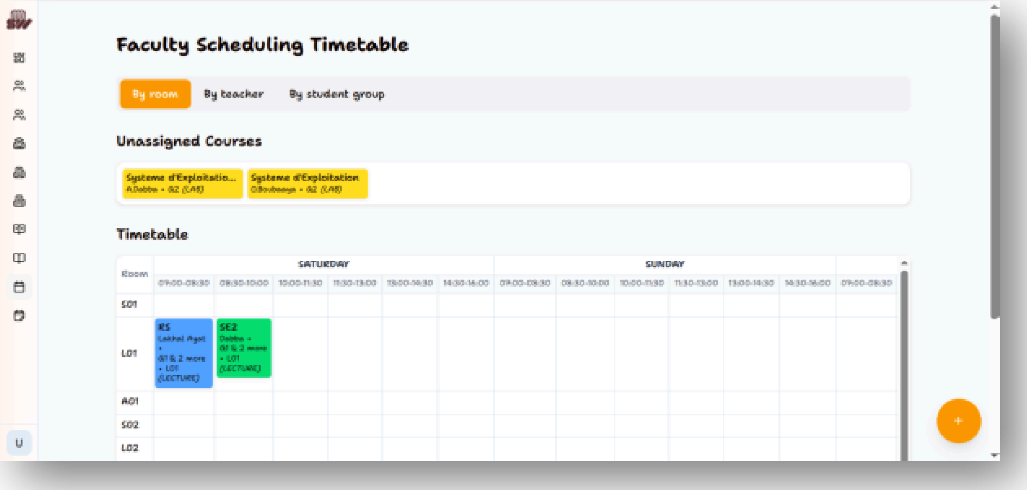
8. Timetable Management

Purpose:

The core of the system—allows for interactive creation and modification of weekly schedules.

Description:

Displays a weekly timetable where classes can be dragged and dropped into time slots, with easy controls to adjust times and distribute courses.



Faculty Scheduling Timetable

Unassigned Courses

Systeme d'Exploitatio... A.Dabba • G2 (LAB)

Systeme d'Exploitation O.Boubaaya • G2 (LAB)

Timetable

Room	SATURDAY						SUNDAY						
	07:00-08:30	08:30-10:00	10:00-11:30	11:30-13:00	13:00-14:30	14:30-16:00	07:00-08:30	08:30-10:00	10:00-11:30	11:30-13:00	13:00-14:30	14:30-16:00	07:00-08:30
S01													
L01	RS Lakhal Ayat • G1 & 2 more • L01 (LECTURE)	SE2 Dabba • G1 & 2 more • L01 (LECTURE)											
A01													
S02													
L02													

figure13. Planning Timetable Page

3. Oreview

In this chapter, we present the results of the practical phase of the project and present the final program, which meets the requirements specified in the previous chapters. We focus on the work environment in which the site was developed, in addition to explaining the database and providing a comprehensive overview of the interfaces that have been designed.

General Conclusion

StaffWise delivers a modern, automated solution to streamline scheduling in educational institutions, addressing inefficiencies in manual processes through a robust web application built with Next.js, Spring Boot, and MySQL. By integrating role-based access, JWT authentication, and dynamic scheduling algorithms, the system minimizes errors, optimizes resource allocation, and empowers administrators to focus on strategic tasks. Its scalable architecture ensures adaptability to diverse institutional needs while maintaining security and usability.

To build on this foundation, future works could incorporate:

1. AI-driven optimization for smarter conflict resolution and workload balancing.
2. Mobile compatibility to enable on-the-go schedule access and updates.
3. Third-party integrations (e.g., student portals, calendar apps) for seamless interoperability.
4. Predictive analytics to forecast resource demands and staffing needs.

Continuous user feedback and iterative testing will ensure StaffWise evolves as a cornerstone of educational efficiency.

Bibliography

[1] J. Smith and A. Johnson, Automated Scheduling Systems in Education: A Comprehensive Review, 2nd ed. New York, NY: Academic Press, 2020.

[2] R. Brown et al., "Optimizing University Timetabling Using Constraint Satisfaction Algorithms," IEEE Transactions on Education, vol. 65, no. 3, pp. 321-335, Aug. 2022, doi: 10.1109/TE.2022.3056789.

[3] K. Taylor and L. Martin, "Database Optimization Techniques for Scalable Educational Systems," IEEE Journal of Database Management, vol. 14, no. 2, pp. 78-92, May 2021, doi: 10.1109/JDM.2021.3071234.

[4] A. Clark and B. Roberts, "Future Trends in AI-Powered Scheduling Systems," IEEE Intelligent Systems, vol. 38, no. 1, pp. 45-57, Jan. 2024, doi: 10.1109/MIS.2024.3352101.

[5] Next.js Documentation, Server-Side Rendering and Static Site Generation, Vercel, 2023. [Online]. Available: https://nextjs.org/docs

[6] Spring Boot Official Documentation, Securing REST APIs with JWT, VMware, 2023. [Online]. Available: https://spring.io/projects/spring-boot

[7] MySQL 8.0 Reference Manual, Indexing and Query Optimization, Oracle, 2023. [Online]. Available: https://dev.mysql.com/doc/refman/8.0/en/

Abstract:

StaffWise tackles the problem of manual class scheduling in schools and colleges by creating an automated system that balances teacher availability, course needs, and classroom resources. The project uses smart methods to build timetables that avoid overlaps, ensure fair workloads for teachers, and make the best use of classrooms. By replacing time-consuming manual planning, it reduces mistakes and saves time for staff. The system also allows schools to adapt schedules easily when changes occur, like a teacher's sudden absence or new course additions. Future improvements could include predicting future needs (like more classrooms during busy terms), schedule generation using AI or connecting with other school tools (like attendance systems). This project shows how automation can turn scheduling from a stressful task into a tool that supports better teaching and learning.

الملخص:

مشروع StaffWise يُعالج مشكلة الجدولة اليدوية للفصول في المدارس والكليات عن طريق نظام آلي يوازن بين توفر المدرسين، احتياجات المواد الدراسية، وموارد القاعات. يستخدم المشروع أساليب ذكية لبناء جداول تمنع التداخلات، تضمن توزيعًا عادلًا لأعباء العمل على المدرسين، وتُحسن استخدام الفصول. باستبدال التخطيط اليدوي المرهق، يقلل النظام من الأخطاء ويوفر وقت الموظفين. كما يسمح للمؤسسات بتعديل الجداول بسهولة عند حدوث تغييرات مفاجئة، مثل غياب مدرس أو إضافة مواد جديدة. يمكن أن تشمل التطويرات المستقبلية توقع الاحتياجات المستقبلية (مثل الحاجة لقاعات إضافية خلال الفصول المزدحمة)، إنشاء الجداول الزمنية باستخدام الذكاء الاصطناعي أو الربط مع أنظمة المدرسة الأخرى (مثل أنظمة الحضور). يُظهر هذا المشروع كيف يمكن للأتمتة تحويل الجدولة من مهمة مرهقة إلى أداة تدعم التعليم والتعلم بشكل أفضل."