# STUDENT–TEACHER BOOKING APPOINTMENT SYSTEM

**Full Stack Web Development Project Report**
*(Unified Mentor Internship)*

---

## 1. Title Page

**Project Title:** Student–Teacher Booking Appointment System
**Domain:** Education (Web-Based Appointment Management System)
**Technologies Used:** HTML, CSS, JavaScript, Firebase (Authentication & Database)
**Difficulty Level:** Easy
**Internship Organization:** Unified Mentor
**Intern Name:** *Vaibhav Thapliyal*
**Duration:** *Nov 2025 – Feb 2026*
**Github:** *https://github.com/vai938/student_teacher_booking*

---

## 2. Abstract

The Student–Teacher Booking Appointment System is a web-based application designed to streamline the process of scheduling appointments between students and teachers. The system enables students to search for teachers, book appointments, and send messages stating the purpose of meetings, while teachers can view, approve, or cancel appointment requests. An administrative module manages teacher records and student registrations.

The application is built using HTML, CSS, JavaScript, and Firebase for authentication and real-time database services. The system improves communication efficiency, reduces manual scheduling overhead, and provides accessibility across devices through a web interface. The project emphasizes modular code design, logging, maintainability, and deployment readiness.

---

## 3. Introduction

Appointment booking systems have become an integral part of modern service delivery in education, healthcare, and professional services. Traditional queue-based or manual scheduling approaches are inefficient and lead to increased waiting times and poor user experience.

This project was undertaken as part of the Unified Mentor internship program to design and implement a web-based appointment booking platform tailored for academic environments. The system allows students and teachers to manage appointments remotely using web-enabled devices, thereby enhancing accessibility and operational efficiency

Student-Teacher Booking Appoint…

.

---

## 4. Problem Statement

In academic institutions, scheduling meetings between students and teachers is often manual and time-consuming, leading to scheduling conflicts, communication gaps, and inefficiencies.

**Problem:**
Design and implement a web-based appointment booking system that enables students and teachers to schedule, approve, and manage appointments efficiently, with centralized administration, real-time updates, and accessibility across devices.

---

## 5. Objectives

The key objectives of this project are:

- To develop a web-based platform for booking student–teacher appointments.
- To implement role-based access for Admin, Teacher, and Student.
- To enable students to search for teachers and request appointments.
- To allow teachers to approve or cancel appointments.
- To provide messaging functionality for appointment purposes.
- To ensure data persistence and authentication using Firebase.
- To follow modular, testable, maintainable, and portable coding practices.

---

## 6. Scope of the Project

**In Scope:**

- Admin module for teacher management and student approval
- Teacher module for appointment approval/cancellation and viewing messages
- Student module for searching teachers, booking appointments, and sending messages
- Authentication and role-based access control
- Firebase-backed real-time data storage
- Logging of user actions
- Public GitHub repository with proper documentation

**Out of Scope (Future Enhancements):**

- Calendar synchronization (Google Calendar, Outlook)
- Video meeting integration
- SMS/Email notifications
- Advanced analytics dashboards

---

## 7. Literature Review (Brief)

Web-based appointment systems are widely used in education and healthcare to improve service efficiency. Cloud-based backends such as Firebase simplify authentication, real-time data synchronization, and deployment for small-to-medium scale applications. Role-based access control and modular frontend architecture are standard practices for maintainable and scalable web systems.

---

## 8. System Architecture

**High-Level Architecture:**

```
Client (Browser)
    ↓
HTML/CSS UI
    ↓
JavaScript Controllers
    ↓
Firebase Authentication
Firebase Firestore / Realtime DB
    ↑
Real-Time Sync & Logging
```

**System Modules:**

- Admin Module
- Teacher Module
- Student Module
- Authentication Module
- Logging Module

---

## 9. Technology Stack

| Layer | Technology Used |
|---|---|
| Frontend | HTML5, CSS3 |
| Scripting | JavaScript (ES6) |
| Backend (BaaS) | Firebase (Auth, Firestore/DB) |
| Hosting | Firebase Hosting / Cloud Platform |
| Tools | VS Code, GitHub |

---

## 10. Functional Requirements

- The system shall allow users to register and log in.
- The system shall allow Admin to add, update, and delete teachers.
- The system shall allow Admin to approve student registrations.
- The system shall allow students to search teachers and book appointments.
- The system shall allow students to send messages with appointment purpose.

- The system shall allow teachers to approve or cancel appointments.
- The system shall allow teachers to view all appointments and messages.
- The system shall maintain logs for user actions.

---

## 11. Non-Functional Requirements

- **Security:** Firebase authentication and role-based access.
- **Maintainability:** Modular code structure and clean architecture.
- **Testability:** Unit-testable business logic.
- **Portability:** Browser-based compatibility across platforms.
- **Reliability:** Consistent real-time updates.
- **Scalability:** Cloud backend to support multiple users.

---

## 12. Implementation Details

### 12.1 Frontend

The frontend provides separate dashboards for Admin, Teacher, and Student with role-based navigation and access control. Responsive layouts ensure compatibility across devices.

### 12.2 Backend (Firebase Integration)

Firebase services are used for:

- User authentication (Email/Password)
- Real-time database for appointments and messages
- Role-based access control
- Hosting and deployment

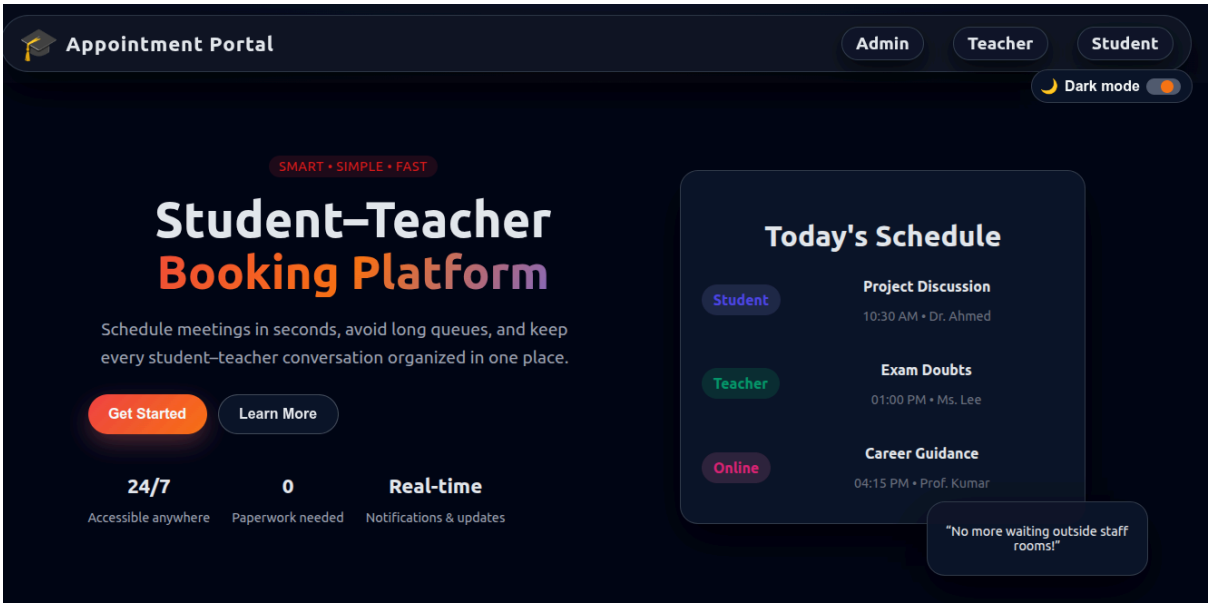
### 12.3 Logging and Code Quality

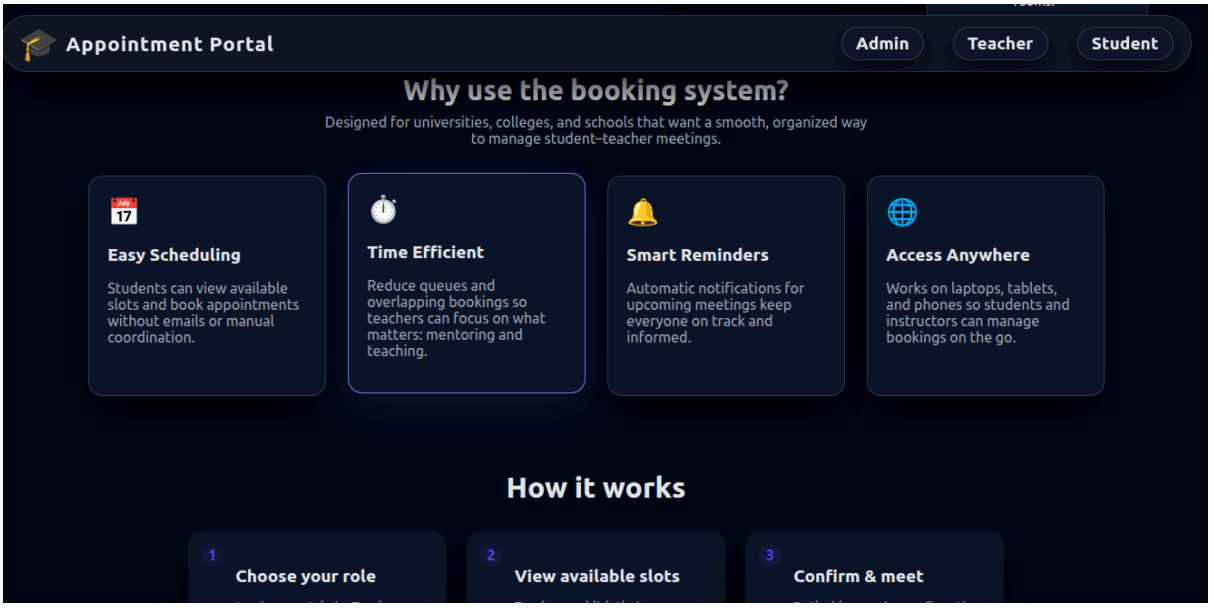
- Logging is implemented for user actions such as login, appointment booking, approval, and cancellation.

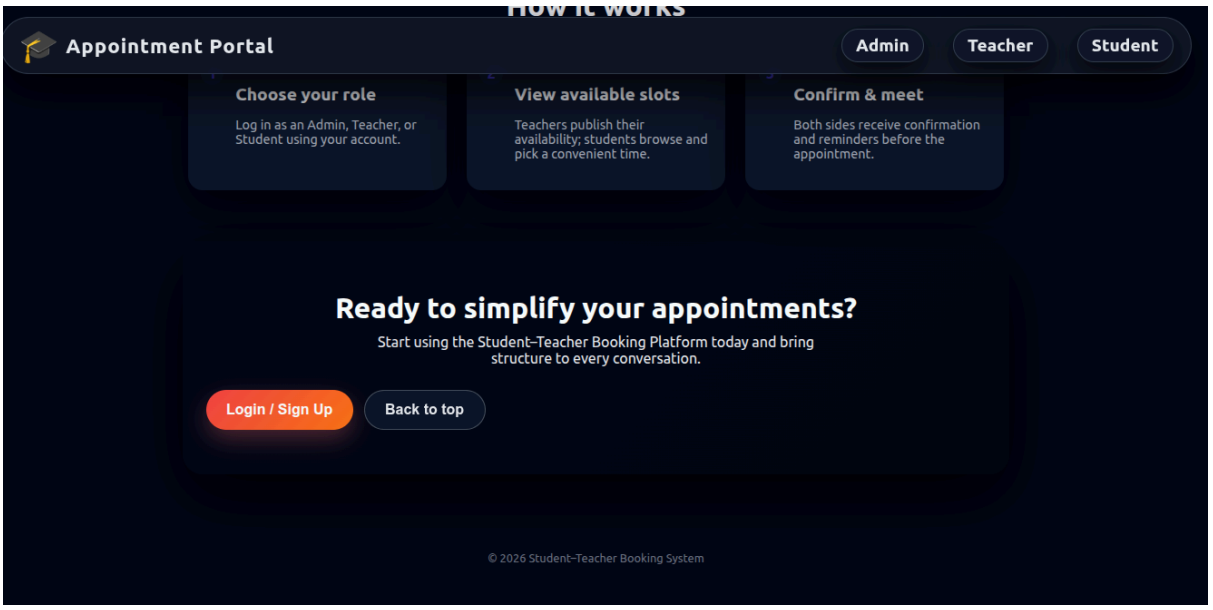
- The codebase follows modular design principles:
    - Safe
    - Testable
    - Maintainable
    - Portable

---

## 13. Output Screenshot






- Login & Registration Screens
- Admin Dashboard
- Teacher Appointment Management
- Student Booking Interface

---

## 14. Testing

**Test Cases:**

- User registration and login
- Role-based access validation
- Appointment booking workflow
- Approval and cancellation flow
- Message sending and retrieval
- Logging verification
- Firebase connectivity testing

---

## 15. Challenges Faced

- Designing role-based access control
- Integrating Firebase authentication with the frontend
- Maintaining real-time updates across users
- Ensuring secure data access rules

- Structuring modular and maintainable code

---

## 16. Results and Outcomes

The system successfully enables students and teachers to manage appointments efficiently with real-time updates and role-based workflows. The application meets all functional requirements and adheres to code quality guidelines, including modularity, logging, and maintainability. The project enhanced practical knowledge of cloud-backed web application development and authentication systems.

---

## 17. Future Enhancements

- Calendar integration and reminders
- Email and push notifications
- Video conferencing integration
- Advanced reporting and analytics
- Mobile application version

---

## 18. Conclusion

The Student–Teacher Booking Appointment System demonstrates the effective use of frontend technologies combined with cloud-based backend services to solve real-world scheduling problems in the education domain. The project fulfills the defined objectives and provides a scalable foundation for future enhancements, aligning well with professional software development practices learned during the Unified Mentor internship.

---