Student-Teacher Appointment Booking System

# 1. Introduction

The Student-Teacher Appointment Booking System is a web-based application that allows students to book appointments with teachers. The platform ensures a streamlined process by incorporating role-based access (Admin, Student, and Teacher), secure authentication via Firebase, and real-time data management using Firestore.

# 2. Technologies Used

- Frontend: HTML, CSS, JavaScript

- Backend & Database: Firebase (Firestore, Authentication)

- Hosting: Firebase Hosting (optional)

- Authentication: Firebase Auth (Email/Password)

# 3. System Features

## 3.1. User Roles

- Admin: Manages user approvals and teacher metadata.

- Student: Can register, log in (after admin approval), and request appointments with teachers.

- Teacher: Can register (login only if added to teacher list), view and manage appointments.

# 4. Functional Modules

## 4.1. Registration & Login

Users register with name, email, password, and role. Admin must approve students and teachers before login is permitted. Admin can log in without needing approval.

## 4.2. Admin Dashboard

View and approve/reject user registration requests. On approving a teacher, admin is prompted for name, subject, and branch (stored in 'teachers' collection). View all approved students and teachers.

## 4.3. Student Dashboard

View list of approved teachers with subject and branch. Book appointments with selected teachers. View status of all appointments: pending, approved, and rejected.

## 4.4. Teacher Dashboard

View appointments assigned to them, categorized by status. Approve or reject appointment requests.

# 5. Database Structure

## Collections

- users: Stores name, email, role, and approved boolean.

- teachers: Stores name, subject, branch, email, and userId.

- appointments: Stores studentEmail, teacherEmail, date, and status.

# 6. Key Features

- Secure login system using Firebase Auth.  
- Role-based access control to segregate features.  
- Live data sync with Firestore database.  
- UI enhancements optimized for laptop view with padding, spacing, and styled cards.

# 7. Challenges & Solutions

|  |  |
| --- | --- |
| Challenge | Solution |
| Multiple users with same name | Emails used as unique identifiers |
| Preventing teachers from logging in before admin addition | Checked if email exists in 'teachers' collection |
| Firestore document reference error | Ensured even segments by not nesting emails in path |
| Displaying clean UI | Used Flexbox and CSS cards for clean layout |

# 8. Conclusion

This project effectively demonstrates the use of Firebase services in building a real-time, scalable, and secure appointment booking platform. The separation of roles and clean UI design make it user-friendly for academic environments.