Detailed Project Report - Student-Teacher Booking Appointment

Project Title: Student-Teacher Booking Appointment

GitHub Repository: https://github.com/Vengatesh521/Full-Stack-Student-Teacher-Booking-Appointment

1. Problem Statement

The traditional methods of booking appointments are often inefficient and time-consuming. This project proposes a web-based solution tailored for the educational domain, allowing students to book appointments with teachers online. It simplifies scheduling, minimizes waiting times, and enables better communication. Students can also send messages specifying the purpose of their appointments.

2. Technologies Used

- Frontend: React.js, HTML5, CSS3, Bootstrap
- Backend: Node.js, Express.js
- Database: MongoDB
- Authentication: JWT (JSON Web Token)
- Deployment: Render (Backend) + Netlify/Vercel (Frontend)
- Version Control: Git, GitHub
- Logging: console.log (extendable to winston/morgan)
- Hosting Justification: Cloud platforms provide accessibility, scalability, and cost-effective solutions.

3. System Modules

Admin:
- Add/Update/Delete Teacher
- Approve/Reject Student Registrations

Teacher:
- Login/Logout
- View Appointments
- Schedule Availability
- Approve or Cancel Appointments
- View Messages

Student:
- Register/Login
- Search Teachers
- Book Appointments
- Send Message (with purpose & time)

4. System Architecture

Frontend: SPA using React with Axios for API communication
Backend: REST API using Express.js
Database: MongoDB
Authentication: JWT
Architecture Type: MVC pattern (Model-View-Controller)

5. Solution Design (LLD)

- Users Collection: Stores student and teacher data
- Appointments Collection: Stores appointment details
- Messages Collection: Stores messages linked with appointments
- APIs: /api/users, /api/appointments, /api/messages
- Components: Login, Register, AdminDashboard, TeacherDashboard, StudentDashboard, AppointmentForm, MessageForm

6. Optimization Strategy

- Code Level: Modular React components, database indexing
- Architecture Level: Separated concerns for scalability
- Performance: Pagination, lazy loading
- Security: JWT, input validation, CORS, request sanitization

7. Project Evaluation Metrics

- Safe: Secured routes, input validation
- Testable: APIs tested via Postman
- Maintainable: Structured modular code
- Portable: Works across OS platforms with Node and MongoDB

8. Test Cases (Sample)

Test Case: Register new student
Input: Valid name, email, password
Expected Output: Success, JWT token returned

Test Case: Login teacher
Input: Correct credentials
Expected Output: Redirect to Teacher Dashboard

Test Case: Book appointment
Input: Date, time, purpose
Expected Output: Appointment saved, pending approval

Test Case: Approve appointment (teacher)
Input: Click 'Approve'
Expected Output: Status updated to 'Approved'

Test Case: Unauthorized access
Input: No token
Expected Output: 401 Unauthorized error

9. Deployment

- Frontend: Netlify/Vercel (To be added)
- Backend: Render (To be added)
- MongoDB: Hosted on MongoDB Atlas

10. ReadMe File Checklist

Detailed Project Report - Student-Teacher Booking Appointment

- Project Overview
- Technologies Used
- Setup Instructions
- Folder Structure
- Deployment Links
- Screenshots (if possible)
- Contribution Guidelines
- License

Conclusion

This project demonstrates a practical and modern solution to appointment management between students and teachers using the MERN stack. It aligns with industry standards and is optimized for clarity, security, and future scalability.

Demo Credentials

Admin: rahul / 123456
Teacher 1: teacher02 / TeachStrong456
Teacher 2: raj / 123456
Student: lax3 / 123456

GitHub Repository

https://github.com/Vengatesh521/Full-Stack-Student-Teacher-Booking-Appointment

Live Demo

https://full-stack-student-teacher-booking.vercel.app/login