**High Level Design (HLD**

**REVISION NO:01**

**LAST DATE OF REVISION:18/11/2024**

**HIGH LEVEL DESIGN (HLD)**

**TEACHER’S CONNECT**

**Teachers Connect**

**1**

**High Level Design (HLD**

**1**

**Teachers Connect**

**DOCUMENT VERSION CONTROL**

|  |  |  |  |
| --- | --- | --- | --- |
| **DATE ISSUED** | **VERSION** | **DESCRIPTION** | **AUTHOR** |
| 18/11/2024 | 1 | Initial HLD-v1.0 | Theja Ashwin H,  Sowdeshwari B Maheshwar R, |
|  |  |  |  |
|  |  |  |  |

**Admin Login Feature - High-Level Design**

**Purpose**

Allow an admin to log in with specific credentials for managing the application (e.g., creating or editing tournaments, viewing user data).

**Feature Details**

1. **Admin Credentials:**
   * **Email:** admin@admin.com
   * **Password:** admin123
2. **Security Measures:**
   * Store the password as a **hashed value** in the database (use bcrypt or a similar library).
   * Use **JWT (JSON Web Token)** for session management.
   * Secure endpoints accessible by the admin with authentication middlewar

**Teachers Connect**

**2**

**Teachers Connect**

**1**

# 

**High Level Design (HLD**

**Contents**

1. Abstract …………………………………………………………………………. 4  
2. Introduction ……………………………………………………………………..5  
3. General Description …………………………………………………………..5  
4. Problem Statement ……………………………………………………………6  
5. Proposed Solution………………………………………………………………6  
6. Further Improvements………………………………………………………..7  
7. Technical Requirements………………………………………………………7  
8. Data Requirements………………………………………………………………8  
9. Tools Requirements …………………………………………………………….8  
10. Tools Used ………………………………………………………………………..8  
11. Constraints ……………………………………………………………………….9  
12. Assumptions ………………………………………………………………………9  
13. Design Details……….....................................................................................9  
14. Process Flow ……………………………………………………………………….10  
15. Proposed Methodology …………………………………………………………10  
16. Deployment Process ……………………………………………………………11  
17. Event Log …………………………………………………………………………..11  
18. Error Handling …………………………………………………………………..11  
19. Dashboards ……………………………………………………………………….12  
20. Key Performance Indicators (KPIs) ……………………………………..14  
21. Conclusions ……………………………………………………………………….15

**3**

**Teachers Connect**

**High Level Design (HLD**

## Abstract

This document provides a detailed overview of the design and implementation of the **Student-Teacher Booking Appointment System**, a modern solution to streamline scheduling within educational institutions. Built with technologies such as **MongoDB**, **Express.js**, **ReactJS**, **Node.js**, and **TailwindCSS**, the system offers a user-friendly platform for managing appointments efficiently.

The system addresses inefficiencies in traditional scheduling by enabling **real-time appointment booking**, **messaging**, and **centralized administrative control**. Students can easily search for teachers and book slots, while teachers can manage their schedules effectively. Administrators oversee the platform, ensuring seamless operations and maintaining user roles.

With features like email notifications and real-time updates, the system enhances communication between students and teachers, fostering improved engagement. Its scalability and accessibility across devices ensure that users can interact with the platform conveniently. This solution not only simplifies scheduling but also lays the foundation for future enhancements, such as mobile applications and advanced analytics.

**4**

**Teachers Connect**

**High Level Design (HLD**

## Introduction

A High-Level Design Document (HLD) serves as a blueprint for understanding and implementing a project, offering a comprehensive overview of the system's architecture, functionality, and technical specifications. It bridges the gap between the initial idea and detailed technical implementation, providing clarity for stakeholders, developers, and project teams.

This document outlines the **Student-Teacher Booking Appointment System**, detailing its design, functionalities, and technological foundation. The HLD includes an overview of system modules, process flows, technical and data requirements, and deployment strategies. It serves as a guide to align the project team with the project’s objectives, ensuring a clear understanding of how the system will function and meet its goals.

The HLD focuses on:

* Defining the problem the system addresses.
* Describing the proposed solution and its components.
* Highlighting the roles of various stakeholders, including students, teachers, and administrators.
* Providing a framework for the system’s technical architecture, tools, and processes.

This document is essential for ensuring that all project contributors are aligned on the high-level aspects of the system, fostering collaboration, and enabling effective execution from concept to deployment.

## Scope

* The increasing reliance on digital solutions in education has highlighted the need for efficient scheduling
* systems. The Student-Teacher Booking Appointment System caters to this requirement by enabling seamless
* communication and appointment management. It offers a centralized platform accessible via web and mobile   
  devices, simplifying the coordination between students and teachers.

## General Description

The system defines three key user roles, each with distinct responsibilities:

* **Admin:** Manages teacher profiles, approves student registrations, and ensures system integrity.
* **Teacher:** Handles scheduling, approves or cancels appointments, and communicates with students.
* **Student:** Registers on the platform, searches for teachers, books appointments, and exchanges messages.

**Teachers Connect**

**5**

**High Level Design (HLD**

## Problem Statement

* Traditional appointment systems in education are often inefficient.
* Inefficiencies result in communication gaps and missed opportunities for academic guidance.
* Existing systems lack solutions tailored to the specific needs of students and teachers.
* This project bridges the gap by providing a robust web-based platform.
* The platform focuses on streamlining scheduling and effectively managing appointments.

## Proposed Solution

The proposed system offers a comprehensive and efficient platform designed to address the challenges of traditional appointment systems in educational settings. By leveraging advanced technologies, the platform simplifies scheduling and fosters effective communication among students, teachers, and administrators. Key features of the system include:

* **Centralized Teacher and Student Management by Admins:**  
  Admins play a pivotal role in ensuring the smooth operation of the system. They can:
  + Create and manage teacher profiles, including details such as name, department, and subjects taught.
  + Approve or deny student registrations to maintain a secure and organized user base.
  + Oversee the platform's overall functionality to uphold its integrity and efficiency.
* **Appointment Scheduling and Approval Workflows for Teachers:**  
  Teachers have access to a user-friendly interface that enables them to:

**Teachers Connect**

**6**

**High Level Design (HLD**

* + Set their availability for appointments based on their schedules.
  + Approve, reschedule, or cancel student appointment requests.
  + Communicate directly with students regarding appointment details or changes.
* **Seamless Search and Booking Functionalities for Students:**  
  Students benefit from an intuitive interface that allows them to:
  + Search for teachers based on criteria such as name, department, or subject expertise.
  + Book appointments in real-time, ensuring immediate confirmation or feedback.
  + Send messages or queries to teachers, providing clarity on appointment purposes or requirements.
* **Real-Time Updates and Email Notifications:**  
  The system keeps all users informed through automated email alerts and real-time updates, minimizing miscommunication and ensuring that appointments are effectively managed.

## Further Improvements

* Future iterations of the system can include:  
  - Integration with institutional calendars to avoid scheduling conflicts.
* Development of a dedicated mobile application for better accessibility.
* Incorporation of AI for optimal time slot suggestions.
* Analytics dashboards for insights into user behavior and system performance.

## Technical Improvements

* **Frontend:** Built with **React.js** for dynamic and interactive user interfaces, styled using **TailwindCSS** for modern, responsive, and highly customizable designs.
* **Backend:** Powered by **Node.js** and **Express.js**, ensuring efficient and reliable server-side operations.
* **Database:** Utilizes **MongoDB** for high-performance, scalable, and flexible data management.
* **Email Services:** Employs **Nodemailer** for seamless real-time email notifications and communication.
* **Hosting:** Deployed on **Render**, offering reliable scalability and easy deployment for consistent performance.

**Teachers Connect**

**7**

**High Level Design (HLD**

## Data Requirements

* **User Information:** Includes details such as names, email addresses, and roles for identification and access control.
* **Appointment Records:** Tracks appointments with status updates (e.g., scheduled, completed, or canceled) and associated timestamps for efficient scheduling and history management.
* **Messaging Logs:** Maintains logs of all communications for clear tracking and effective user interaction

## Tools Requirements and Tools Used

* **Development:**
  + **VS Code** for code editing and development.
  + **Git** for version control and collaborative workflows.
* **Testing:**
  + **Postman** for comprehensive end-to-end testing.

**Tools Used**

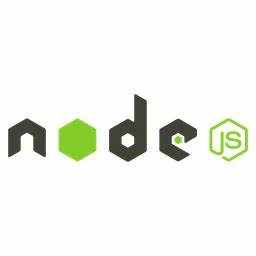
* **Frontend:**
  + **Vite for** a blazing fast frontend build tool powering the next generation of web applications.
  + **React.js** for building dynamic user interfaces.
  + **TailwindCSS** for responsive and customizable styling.
* **Backend:**
  + **Express.js** for server-side logic.
  + **MongoDB** for efficient and scalable data storage.
* **Utilities:**
  + **Nodemailer** for sending real-time email alerts.
* **Deployment:** **Render** for hosting and ensuring scalability.

**8**

**Teachers Connect**

**High Level Design (HLD**

A blue and white symbol

Description automatically generated   A logo with black and yellow text

Description automatically generated

A blue waves on a black background

Description automatically generated A logo of a company

Description automatically generated  A blue ribbon on a white background

Description automatically generated

## Constraints and Assumptions

**Constraints:**

* **Hosting Resources:** Limited server and storage resources during the initial deployment phase.
* **Internet Dependency:** The system relies on consistent and reliable internet connectivity for seamless operation.

**Assumptions:**

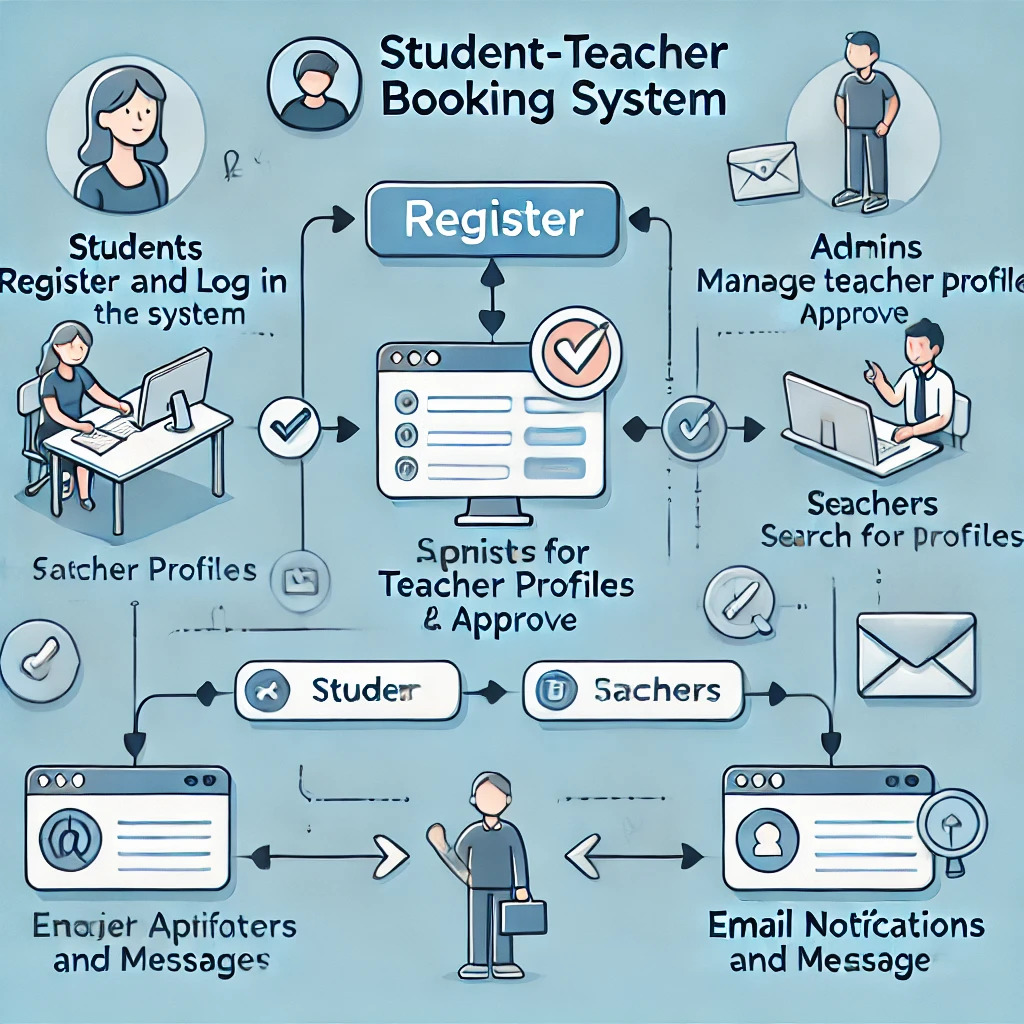
* **User Proficiency:** Users possess basic technical skills required to navigate and interact with the system.
* **Email Validity:** Email addresses provided by users are accurate and functional for notification purposes.
* **Schedule Compliance:** Teachers will adhere to their predefined schedules for accurate planning and execution.

**9**

**Teachers Connect**

**High Level Design (HLD**

## Process Flow



1. **User Registration and Login:**
   * Students create accounts and securely log in to access the system.
2. **Teacher Profile Management:**
   * Admins manage teacher profiles, verify credentials, and approve student registrations.
3. **Appointment Booking:**
   * Students search for teachers based on preferences (e.g., subject, availability) and book appointments.

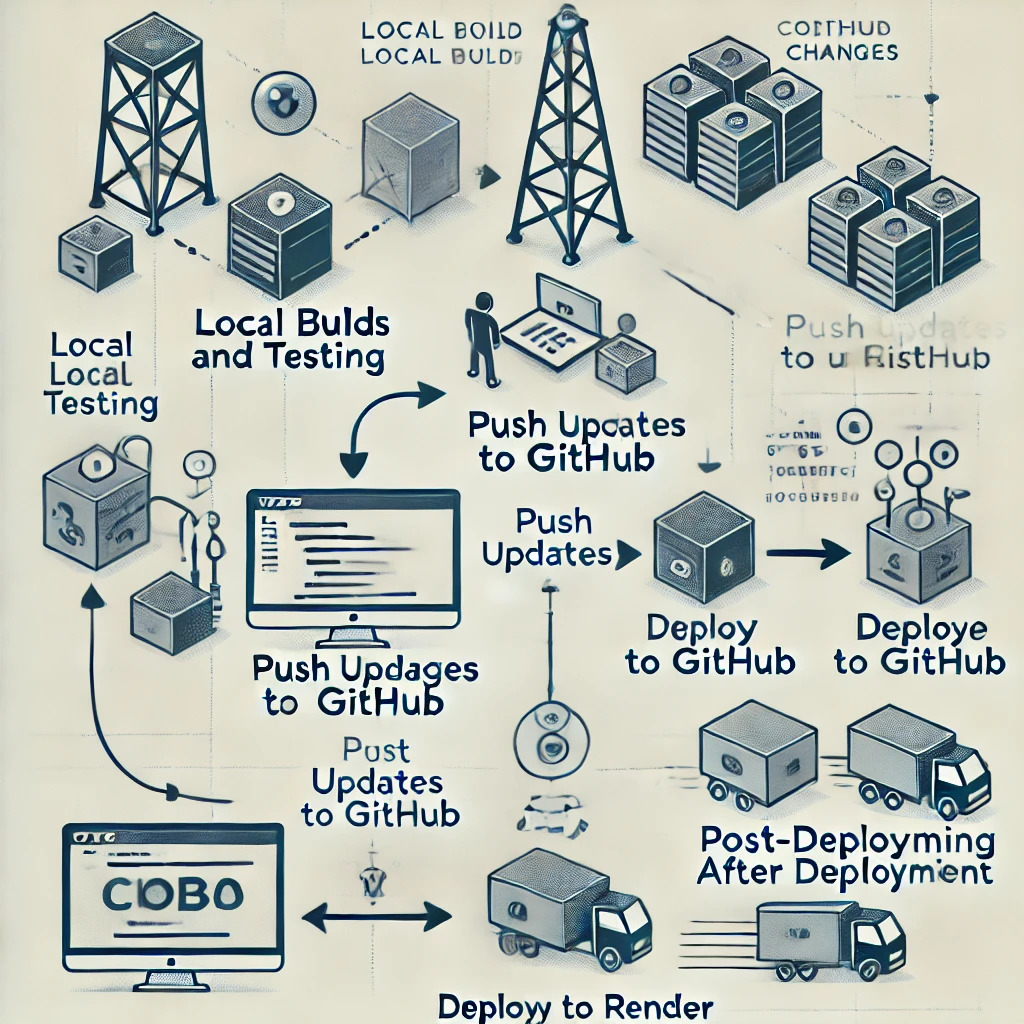
**10**

**Teachers Connect**

**High Level Design (HLD**

1. **Appointment Management:**
   * Teachers review, approve, or reschedule appointments as needed to align with their availability.
2. **Communication and Notifications:**
   * The system sends automated email notifications and facilitates direct messaging for effective communication between users

## Deployment Process



**11**

**Teachers Connect**

**High Level Design (HLD**

1. **Local Builds and Testing:**
   * Perform thorough local builds and run tests to ensure functionality and quality before deployment.
2. **Push Updates to GitHub:**
   * Push the latest code changes and updates to the GitHub repository for version control and collaboration.
3. **Deploy to Render:**
   * Deploy the system on the Render platform, ensuring that all configurations are properly set for smooth deployment.
4. **Post-Deployment Monitoring:**
   * Monitor the system’s performance after deployment to identify and address any potential issues or bottlenecks.

## Error Handling

* **Input Validation**: Ensure user inputs are thoroughly validated to prevent incorrect or malicious data submission, enhancing system security and integrity.
* **User-Friendly Error Messages**: Display clear, concise, and actionable error messages that guide users in resolving issues, improving their experience.
* **Server-Side Recovery**: Implement automatic failover and retry mechanisms to handle server-side failures, ensuring minimal disruption and maintaining system stability.
* **Logging and Monitoring**: Integrate comprehensive logging and real-time monitoring to detect, track, and resolve issues proactively, preventing future occurrences.
* **Graceful Degradation**: Ensure the application can still provide limited functionality during errors, maintaining usability even during unexpected failures.
* **Error Reporting**: Provide users with an easy way to report errors, helping the development team gather information for faster resolution.

**12**

**Teachers Connect**

**High Level Design (HLD**

## Dashboards and KPIs

## Dashboard

**Admin Dashboard:**

* **Manage User Profiles: Admin can manage teacher and student profiles (create, edit, remove).**
* **Monitor Activities: Oversee all appointments and user activities.**
* **Analytics: Display key metrics such as bookings, teachers, and students using React and Charts.js to visualize data through charts like line or bar graphs.**

**Teacher Dashboard:**

* **View Appointments & Manage Availability: Teachers can manage their schedules and availability.**
* **Track Progress & Session History: Access detailed student progress and session history.**
* **Set Preferences: Teachers can configure their preferred time slots and cancellation policies.**

**Student Dashboard:**

* **Book, Reschedule, Cancel Appointments: Students can book or modify appointments with teachers.**
* **Track Appointments: View upcoming and past appointments.**
* **Session History: Access notes, progress updates, and session history.**

**By integrating React with Charts.js, the dashboards offer real-time data visualization, helping admins track performance, and enabling teachers and students to manage their activities efficiently.**

**Teachers Connect**

**13**

**High Level Design (HLD**

KPIs:

* **Number of Successful Bookings**:
  + Track the total number of confirmed appointments to gauge the platform's usage and demand.
* **Average Time to Confirm Appointments**:
  + Measure the time it takes from booking a session to confirmation, indicating system efficiency and responsiveness.
* **User Satisfaction Ratings**:
  + Gather feedback through surveys to assess user satisfaction with the booking process, appointments, and overall experience.

**14**

**Teachers Connect**

**High Level Design (HLD**

## Conclusions

* The Student-Teacher Booking Appointment System is a scalable, efficient, and user-centric solution for educational scheduling challenges.
* Utilizes advanced web technologies to ensure seamless communication, accessibility, and operational efficiency.
* Future enhancements will refine the system, adapting it to the evolving needs of the education sector.
* Ongoing improvements will focus on enhancing the user experience and system functionality.

**15**

**Teachers Connect**