**REVISION NO:01**

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

**Architecture Document**

**TEACHER’S CONNECT**

**Architecture**

**Teachers Connect**

**3**

# 

Architecture-1.0v

**Teachers Connect**

**3**

**Architecture**

**Architecture**

**3**

**Teachers Connect**

## Contents

1.Introduction ……………………………………………………………..3

2.System Architecture ……………………………………………………………..3

3.Architecture Design ……………………………………………………………..3

4.Data Flow ……………………………………………………………..3

5.Api Design ……………………………………………………………..3

6.Business Logic …………………………………………………………….. 4

7.Deployment …………………………………………………………….. 4

# Architecture Design Document

**Teachers Connect**

**3**

**Architecture**

## 1. Introduction

This document outlines the architecture design for the project, inspired by the Teacher Connect interface. It includes detailed descriptions of system components, data flow, business logic, and deployment strategies.

## 2. System Architecture

The system employs a modern web-based architecture leveraging the MERN stack (MongoDB, Express.js, React.js, Node.js). It is designed for scalability, maintainability, and optimal performance.

Key components include:

1. Frontend: Built using React.js for a dynamic and responsive user experience.

2. Backend: Developed with Node.js and Express.js for robust server-side logic.

3. Database: MongoDB for flexible and scalable data storage.

4. Hosting: Hosted on platforms like AWS or Render for reliability and scalability.

## 3. Architecture Description

The architecture is divided into three primary layers:

1. Presentation Layer: Handles user interfaces and interactions via React.js.

2. Business Logic Layer: Processes core server-side operations using Node.js.

3. Data Layer: Manages data persistence and retrieval using MongoDB.

## 4. Data Flow

The data flow between the layers ensures seamless interaction:

1. User inputs data via the frontend interface, which sends HTTP requests to the backend.

2. The backend processes these requests, applying business logic, and interacts with the database.

3. Responses are sent back to the frontend, which updates the UI accordingly.

## 5. API Design

Key API endpoints include:

/api/v1/auth: Handles user authentication.

/api/v1/users: Manages user profiles.

/api/v1/appointments: Handles appointment bookings.

**Teachers Connect**

**3**

**Architecture**

/api/v1/messages: Enables user communication.

## 6. Business Logic

The system incorporates robust algorithms and functions, including:

1. User authentication using JWT.

2. Role-based access control for admin, teacher, and student functionalities.

3. Appointment scheduling with conflict prevention.

## 7. Deployment

The deployment process ensures high availability and scalability:

1. Development Environment: Local setup for React and Node.js.

2. Production Environment: Hosted on Render or AWS with MongoDB Atlas for database management.

3. CI/CD Pipeline: Automates building, testing, and deployment using GitHub Actions.