**Full Stack Development with MERN**

**API Development and Integration Report**

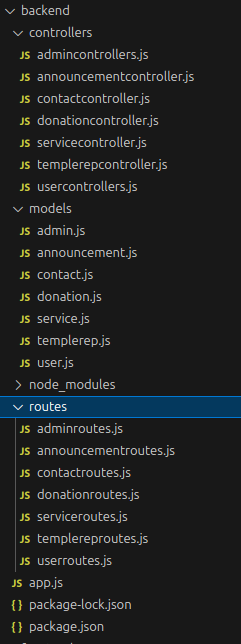
| Date | 18/07/2024 |
| --- | --- |
| Team ID | SWTID1719999205 |
| Project Name | Darshan Ease |
| Maximum Marks |  |

**Project Title:** Darshan Ease   
**Date:** 18/07/2023  
**Prepared by:** [Sashidhar Reddy, Adhvik Sai, A Bharath, Yuva Sai Rithvik]

**Objective**  
The objective of this report is to document the API development progress and key aspects of the backend services implementation for the Darshan Ease project.

**Technologies Used**

* **Backend Framework:** Node.js with Express.js
* **Database:** MongoDB
* **Authentication:** Bcrypt

**Project Structure**  


**Key Directories and Files**

1. **/controllers**
   * Contains functions to handle requests and responses.
2. **/models**
   * Includes Mongoose schemas and models for MongoDB collections.
3. **/routes**
   * Defines the API endpoints and links them to controller functions.
4. **/middlewares**
   * Custom middleware functions for request processing.

**API Endpoints**

The API endpoint used is http://localhost:5500

**User Authentication**

* **POST /api/user/register** - Registers a new user.
* **POST /api/user/login** - Authenticates a user and returns a token.

**Admin Authentication**

* **POST /api/admin/register** - Registers a new admin.
* **POST /api/admin/login** - Authenticates a admin and returns a token.

**Temple RepresentatveAuthentication**

* **POST /api/templerep/register** - Registers a new templerep.
* **POST /api/templerep/login** - Authenticates a templrerp and returns a token.

**User Management**

* **GET /api/user/-** Retrieves user information by ID.
* **PUT /api/user/**- Updates user information by ID.
* **DELETE /api/user –** Deletes posts by user ID.

**Admin Management**

* **GET /api/admin/-** Retrieves admin information by ID.
* **PUT /api/admin/**- Updates admin information by ID.
* **DELETE /api/admin –** Deletes posts by admin ID.

**Temple Representative Management**

* **GET /api/templerep/-** Retrieves templerep information by ID.
* **PUT /api/templerep/**- Updates templerep information by ID.
* **DELETE /api/templerep –** Deletes posts by representative ID.

**Integration with Frontend**  
The backend communicates with the frontend via RESTful APIs. Key points of integration include:

* **User Authentication:** Tokens are passed between frontend and backend to handle authentication.
* **Data Fetching:** Frontend components make API calls to fetch necessary data for display and interaction.

**Error Handling and Validation**  
Describe the error handling strategy and validation mechanisms:

* **Error Handling:** Centralized error handling using middleware.
* **Validation:** Input validation using libraries like express-validator.

**Security Considerations**  
Outline the security measures implemented:

* **Authentication:** Secure token-based authentication.
* **Data Encryption:** Encrypt sensitive data at rest and in transit.