

Full Stack Development with MERN

API Development and Integration Report

Date	19-07-24
Team ID	SWTID1720166168
Project Name	Project - Journify
Maximum Marks	

Project Title:Journify

Date: 19-07-24

Prepared by: SWTID1720166168

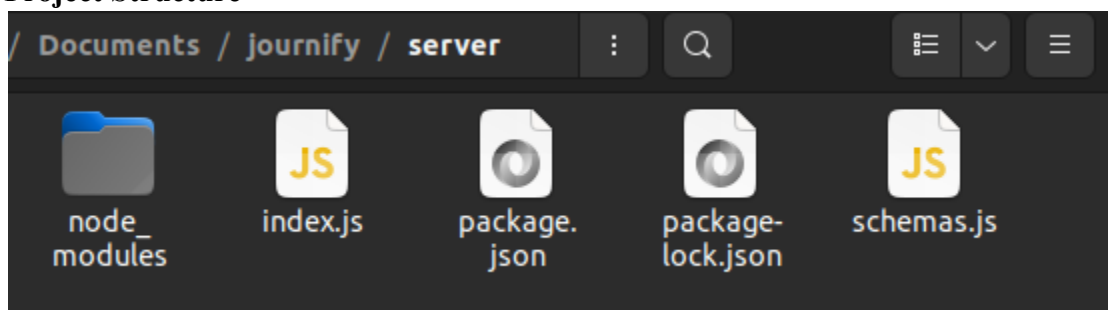
Objective

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

Technologies Used

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

Project Structure



Key Directories and Files

1. **/schemas**
 - o Includes Mongoose schemas and models for MongoDB collections.
2. **/index.js**
 - This file includes all the routes -

/register
/login
/approve-operator
/reject-operator
/fetch-user/:id
/fetch-users
/add-flight
/update-flight
/fetch-flights
/fetch-flight/:id
/fetch-bookings
/book-ticket
/cancel-ticket/:id

API Endpoints

A summary of the main API endpoints and their purposes:

User Authentication

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

Operator Approval

- **POST /approve-operator**- Approves an operator's request.
- **POST /reject-operator**- Rejects an operator's request.

User Management

- **GET /fetch-user/:id** - Retrieves a specific user's details by ID.
- **GET /fetch-users** - Retrieves a list of all users.

Flight Management

- **POST /add-flight** - Adds a new flight.
- **PUT /update-flight** - Updates an existing flight.
- **GET /fetch-flights** - Retrieves a list of all flights.
- **GET /fetch-flight/:id** - Retrieves details of a specific flight by ID.

Booking Management

- **GET /fetch-bookings** - Retrieves a list of all bookings.
- **POST /book-ticket** - Books a new ticket.
- **DELETE /cancel-ticket/:id** - Cancels a ticket by 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:** Errors are handled at each point using try-catch blocks.
- **Validation:** Input validation at each step .

Security Considerations

Outline the security measures implemented:

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