🧳 Enterprise Travel & Expense Management System

A robust MERN stack application designed for enterprises to streamline travel and expense tracking. It features multi-level approvals, role-based access control, real-time notifications, and secure authentication, ensuring compliance and transparency in corporate workflows.

Comprehensive System Overview

This Enterprise Travel & Expense Management System is engineered for modern organizations to efficiently handle travel requests and expense claims. It supports a comprehensive set of features tailored for corporate environments.

Multi-Level Approvals

Streamlined, configurable approval workflows tailored for various organizational hierarchies.

Role-Based Dashboards

Customized dashboards and functionalities based on user roles (Employee, Manager, Admin).

Real-Time Notifications

Instant alerts and email notifications for status updates, ensuring timely responses.

Audit Logging & Compliance

Detailed audit trails of all activities for transparency and regulatory compliance.

Key Features at a Glance

Core Functionality

- **Role-Based Access Control:** Distinct roles for Employee, Manager, Admin, and Super Admin with tailored permissions.
- **Multi-Level Approval Workflow:** Configurable routing based on Manager, Admin, and Super Admin hierarchy.
- **Real-Time & Email Notifications:** Instant alerts via Socket.IO and comprehensive email updates through Nodemailer.
- **Audit Logging:** Detailed history of all requests and claims with advanced filtering capabilities.
- **CSV Export:** Convenient export of all or filtered data for reporting and analysis.

Security & User Experience

- **Secure Authentication:** Robust security measures including JWT, bcrypt, Helmet, CORS, and Rate Limiting.
- **Responsive UI & Dark Mode:** Optimized for various devices with a modern, user-friendly design and dark mode toggle.
- **Attachment Support:** Secure upload and management of bills and supporting documents.
- **Password Management:** Secure mechanisms for password changes and resets to maintain account integrity.

🔐 Role-Based Access & Workflow Logic

Employee Privileges

Submits new travel requests and expense claims, views the status of their personal submissions, and manages their profile.

Manager Responsibilities

Reviews, approves, or rejects requests submitted by their assigned team members. Managers cannot approve or reject their own requests.

Admin Capabilities

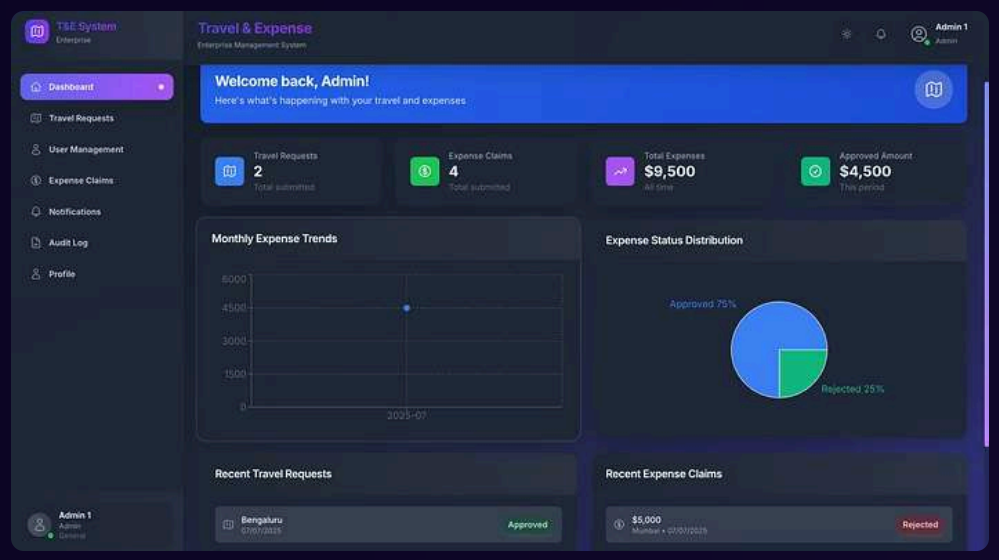
Manages all user accounts, oversees all requests and audit logs, and assigns roles. Once an Admin acts on a request, Managers cannot intervene.

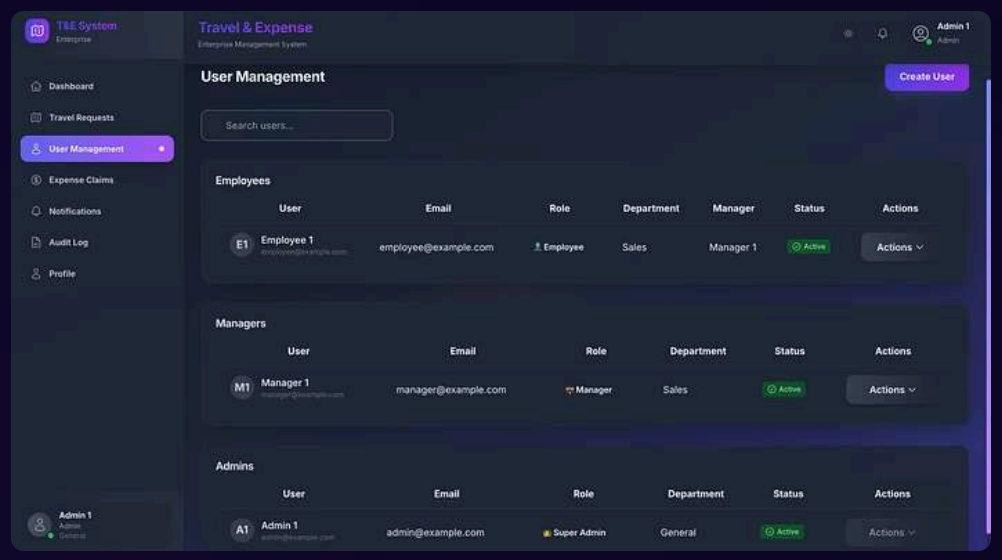
Super Admin Authority

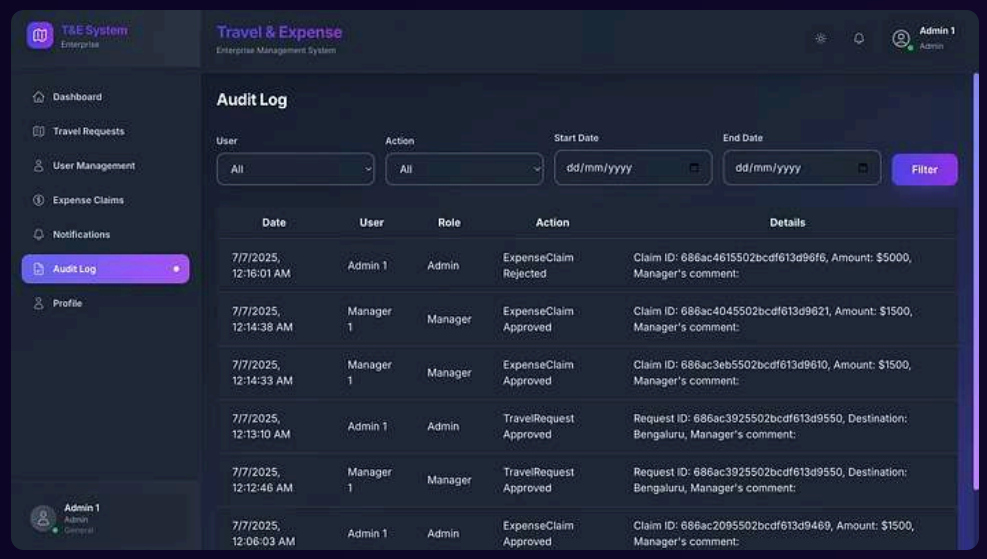
Possesses ultimate control, including the ability to promote any user to Admin. The Super Admin account is non-deletable for system integrity.

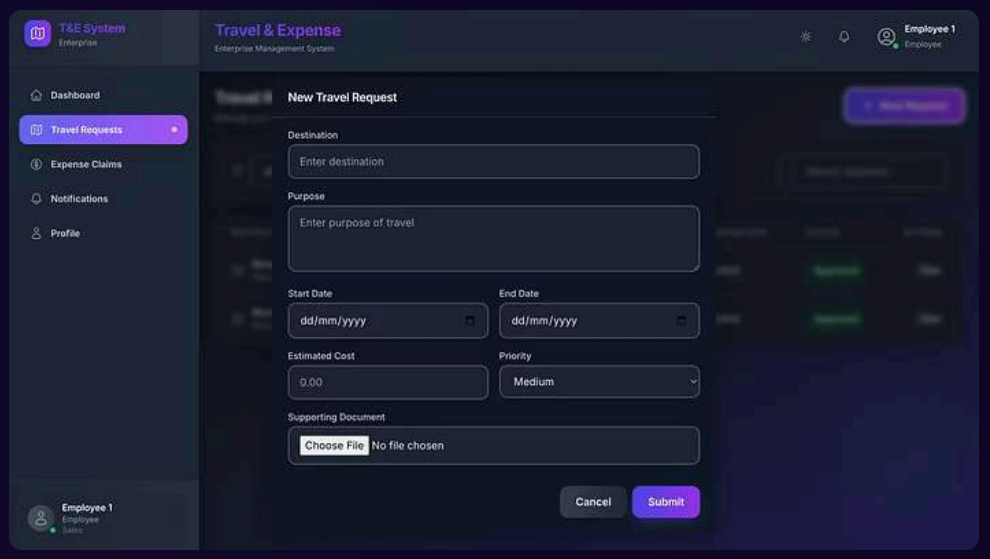
The system ensures robust security and validation. No past dates are accepted for new requests, and expense claims must align with approved travel dates. All required fields and documents are strictly enforced to maintain data integrity.

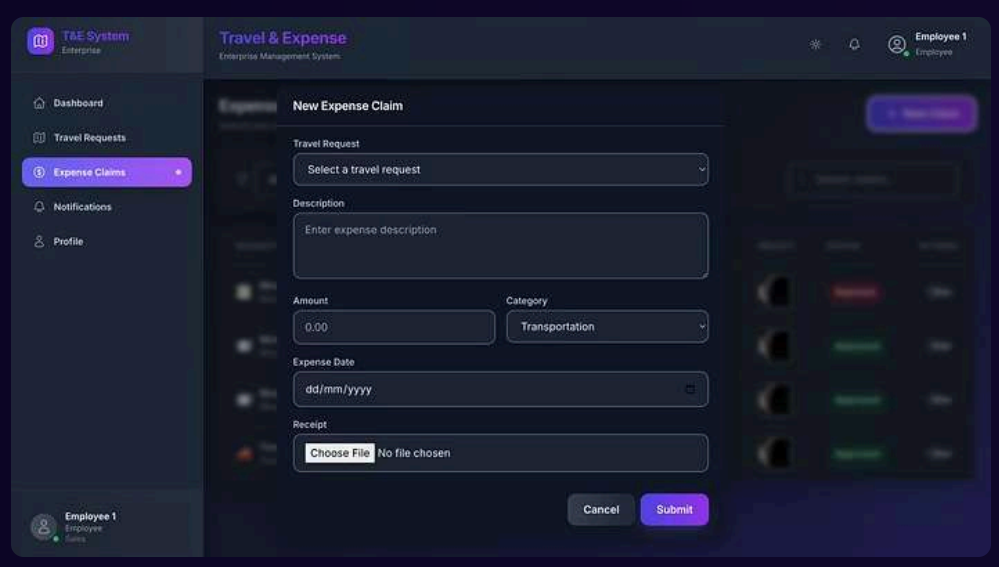
System Dashboards & Interfaces

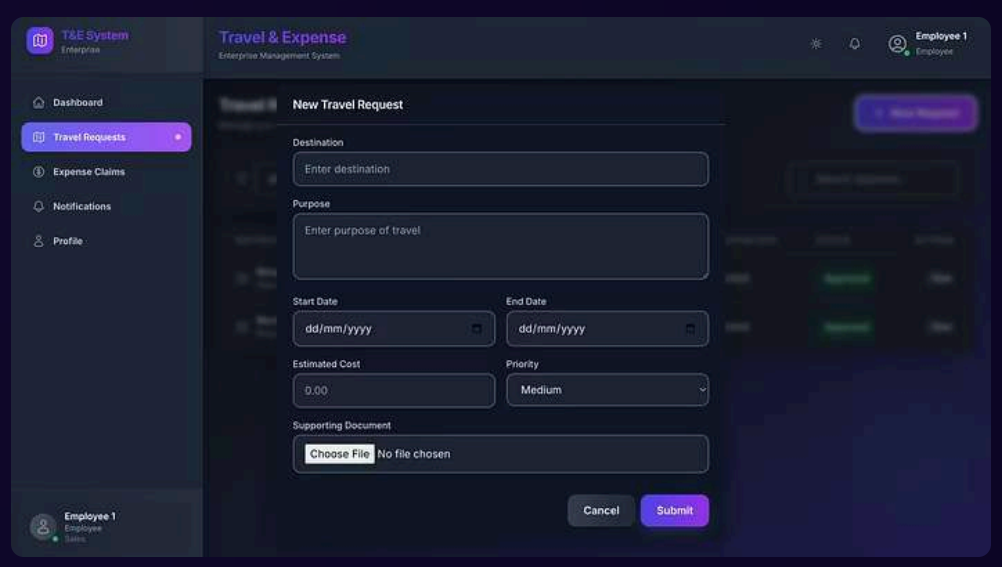










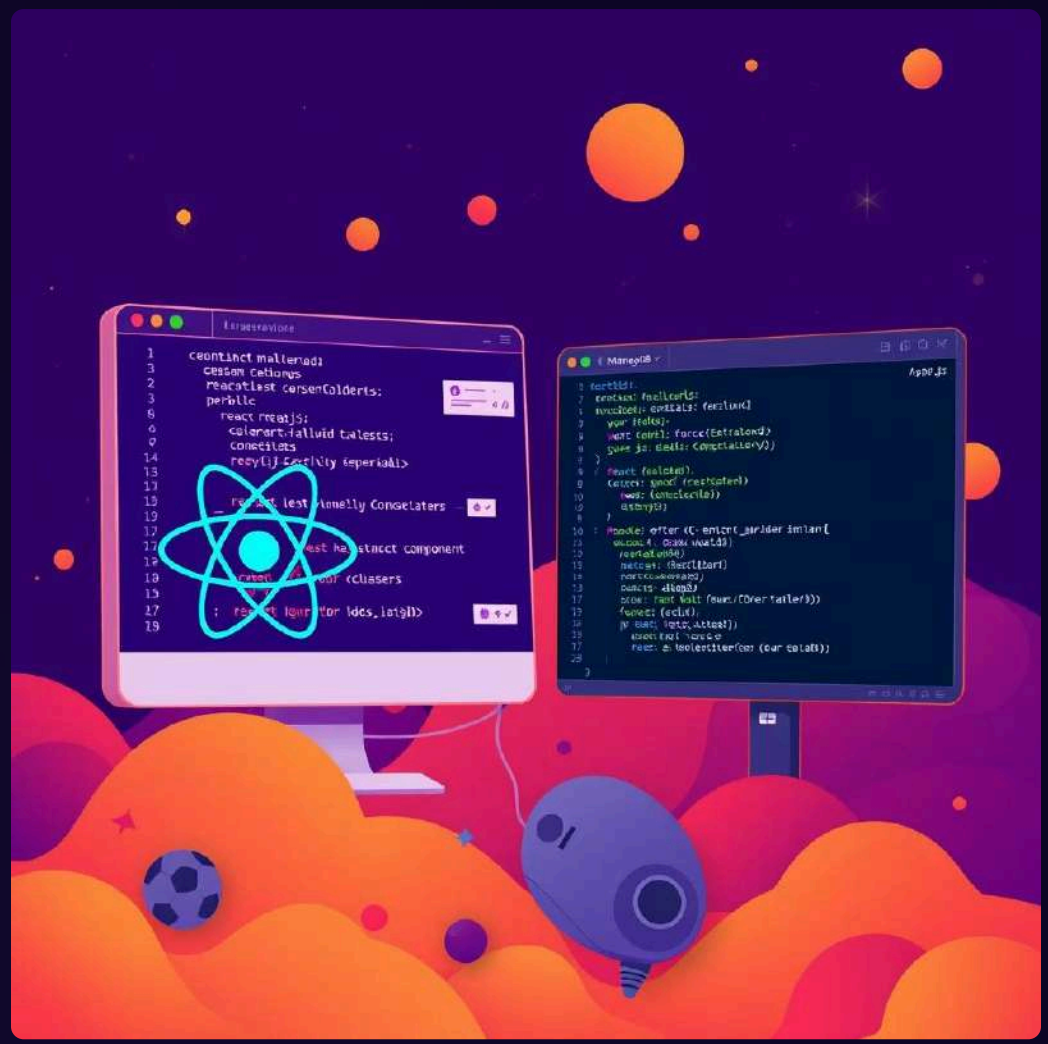


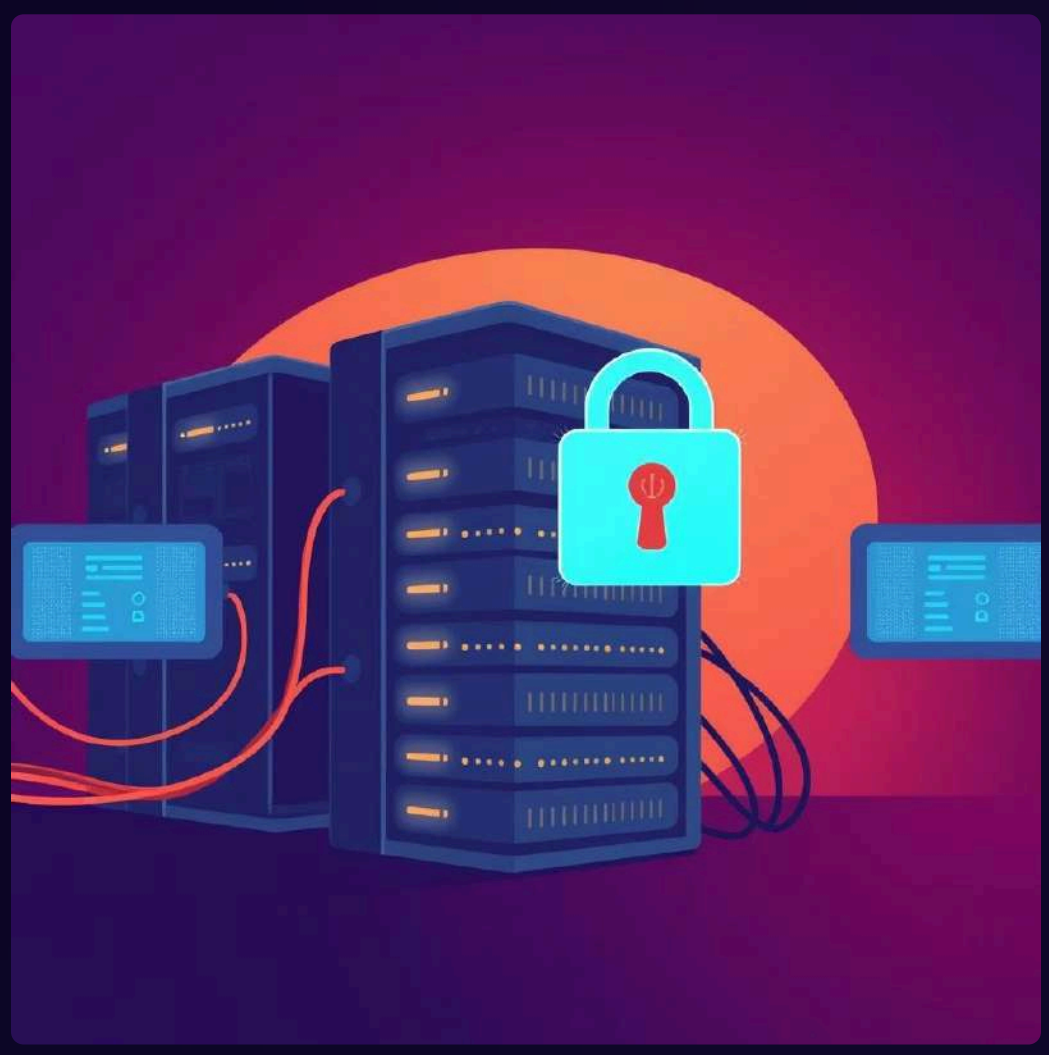
These screenshots illustrate the intuitive and role-specific user interfaces, from the comprehensive Admin Dashboard to the efficient Employee Expense Claim Form, showcasing the system's responsiveness and dark mode capabilities.

⚙️ MERN Stack Architecture

Frontend Technologies

- **React.js:** Dynamic UI with Hooks & Context API for state management.
- **Tailwind CSS:** Utility-first framework for rapid and responsive styling.
- **React Router:** Seamless single-page application navigation.
- **Axios:** Efficient HTTP client for API requests.
- **Socket.IO Client:** Enables real-time, bidirectional communication for instant notifications.
- **React Toastify:** Provides customizable, non-blocking user notifications.





Backend Technologies

- **Node.js & Express.js:** Robust runtime and flexible framework for building RESTful APIs.
- **MongoDB & Mongoose:** NoSQL database for scalable data storage and an ODM for simplified data interaction.
- **Socket.IO:** Facilitates real-time communication for instant alerts.
- **Nodemailer:** Integrated for reliable email notifications.
- **Security Stack:** JWT for authentication, bcrypt for password hashing, and Helmet, CORS, & Express Rate Limit for comprehensive API security.

🛠️ Installation & Setup

ⓘ To set up the Enterprise Travel & Expense Management System, follow these steps to clone the repository and install dependencies for both the backend and frontend.

```
git clone https://github.com/Kush-Varshney/Enterprise-Travel-Expense-System.git
cd Enterprise-Travel-Expense-System

# Server Setup
cd server
npm install

# Client Setup
cd ../client
npm install
```

Ensure you have Node.js and npm installed on your system before proceeding. This process will prepare your development environment to run the application.

🌐 Environment Variable Configuration

Proper configuration of environment variables is crucial for the system's functionality, especially for database connection, authentication, and email services.

Server Configuration (server/.env)

```
PORT=4000
MONGODB_URI=your_mongodb_connection_string
JWT_SECRET=your_secret_key
EMAIL_USER=your_email@example.com
EMAIL_PASS=your_email_password
CLIENT_URL=http://localhost:3000
EMAIL_FROM=Travel Expense System
<noreply@yourdomain.com>
```

MONGODB_URI: Your MongoDB connection string (e.g., from MongoDB Atlas or a local instance).

JWT_SECRET: A strong, random string for JWT token signing.

EMAIL_USER & EMAIL_PASS: Credentials for your email service (e.g., Gmail, SendGrid) used by Nodemailer.

Client Configuration (client/.env)

```
REACT_APP_API_URL=http://localhost:4000
```

REACT_APP_API_URL: The base URL of your backend API server. This should match the `PORT` configured in the server's `.env` file.

⚠ Ensure these files are created in their respective directories and are not committed to version control for security.

📂 Project Folder Structure

The project is organized into logical directories for clear separation of concerns between the client (frontend) and server (backend), along with dedicated folders for static assets and documentation.

```
Enterprise-Travel-Expense-System/
├── README.md
├── .gitignore
├── screenshots/
│   ├── admin-dashboard.png
│   └── ... (other screenshots)
├── client/                 # React Frontend
│   ├── package.json
│   ├── public/
│   ├── src/
│   │   ├── App.js
│   │   ├── components/     # Reusable UI components
│   │   ├── contexts/       # React Context API for global state
│   │   └── pages/          # Specific page components
│   └── .env.example
└── server/                 # Node.js/Express Backend
    ├── package.json
    ├── server.js           # Main server entry point
    ├── scripts/
    ├── middleware/         # Express middleware
    ├── models/             # Mongoose schemas for MongoDB
    ├── routes/             # API route definitions
    ├── utils/              # Utility functions (e.g., sendEmail)
    └── .env.example
```

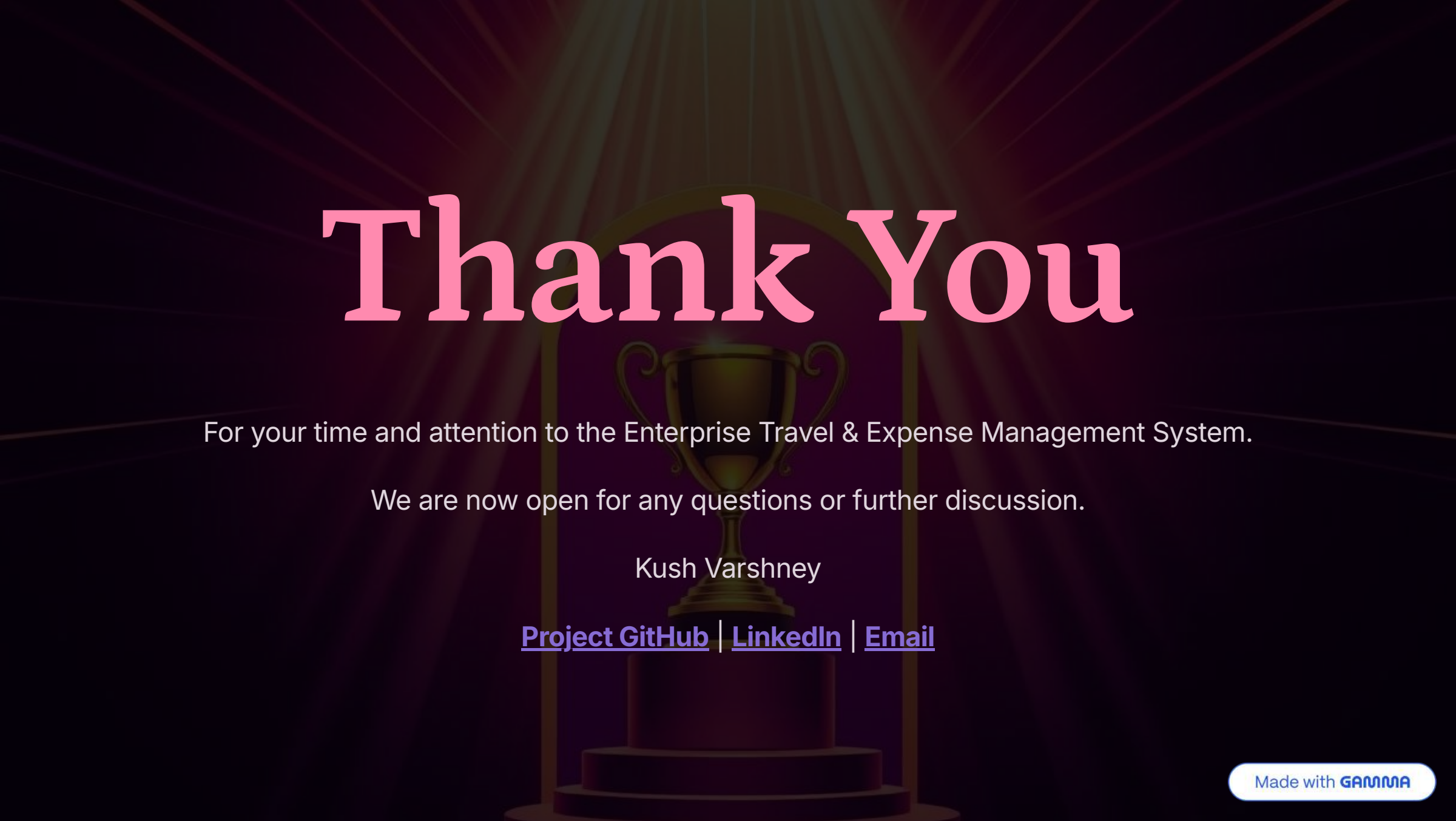
This structure facilitates modular development, maintainability, and scalability, allowing developers to quickly locate and manage specific parts of the application.

📮 Key API Endpoints

The system exposes a set of RESTful API endpoints to manage user authentication, travel requests, expense claims, and administrative tasks.

POST	/api/auth/register	Registers a new user account.
POST	/api/auth/login	Authenticates a user and returns a JWT token.
POST	/api/request/	Creates a new travel or expense request.
GET	/api/request/mine	Retrieves all requests submitted by the authenticated user.
PATCH	/api/request/:id/approve	Approves a specific travel or expense request by ID.
PATCH	/api/request/:id/reject	Rejects a specific travel or expense request by ID.
GET	/api/admin/users	Super Admin: Fetches a list of all registered users.
PATCH	/api/admin/user/:id/role	Super Admin: Changes the role of a specific user by ID.
GET	/api/export	Exports system data to a CSV file (all or filtered).

These endpoints provide the necessary interface for secure and efficient management of enterprise travel and expense operations, supporting all user roles and administrative functions.



Thank You

For your time and attention to the Enterprise Travel & Expense Management System.

We are now open for any questions or further discussion.

Kush Varshney

Project GitHub | LinkedIn | Email