Site : https://travel-agency-vsg2.vercel.app/

Admin id and password:

id: admin

password: password@123

Travel Agency Website

Project Overview

Welcome to the **Travel Agency Website**, a dynamic and fully functional platform designed to provide users with an intuitive and seamless experience in exploring travel packages, destinations, and services. Whether you're looking to book a vacation or manage your travel business, this website offers all the features you need to get started.

With this application, users can easily browse through available travel packages, explore detailed information about various destinations, download invoices, and enjoy a smooth, responsive design across all devices. Additionally, the admin panel allows authorized users to manage and configure travel-related data.

Key Features

1. User Authentication

- Secure login and admin access using a **protected route system**.
- Only authenticated users can access the **admin dashboard**, ensuring secure management of travel packages and user data.

2. Explore Destinations & Packages

- Users can **view different travel packages** and explore detailed information about destinations.
- **Unique package URLs** with pricing, itineraries, and in-depth descriptions allow users to make informed decisions.

3. Admin Dashboard

- **Admin users** can manage and configure travel packages, destinations, and invoices with ease.
- **Protected routes** ensure only authorized admins can access the dashboard and modify data.

4. Invoices

- **Generate and download invoices** as PDFs for users, providing easy access to booking receipts.

5. Fully Responsive Design

- Built using **modern web technologies** for a smooth user experience across desktops, tablets, and mobile devices.
- The website is fully optimized for all screen sizes, ensuring that users can access the site from anywhere.

6. Modern Routing with React Router

- A clean and efficient **routing mechanism** using **React Router** for seamless navigation across the site.
- Direct access to **package details**, **invoice management**, and more, without unnecessary page reloads.

7. User-Friendly UI

- An **interactive and engaging interface** with a simple layout, intuitive navigation, and clear calls-to-action.
- The homepage showcases **popular destinations** and **featured packages** to captivate users and guide them through the site.

Tech Stack

Frontend:

- **React**: Building dynamic and fast user interfaces that provide a seamless experience.
- **Tailwind CSS**: Utility-first CSS framework for creating responsive designs quickly and efficiently.

Backend:

- **Node.js**: Handles authentication, database interactions, and server-side operations.
- **MongoDB**: A NoSQL database used to store user data, travel packages, and invoices.

Setup Instructions

Follow the steps below to set up and run the Travel Agency Website project locally on your machine.

1. Clone the Repository

First, clone the repository to your local machine using the following command:

```
git clone https://github.com/your-username/travel-agency.git
```

2. Navigate to the Project Directory

Once cloned, navigate into the project directory:

```
cd travel-agency
```

3. Install Dependencies

The project uses **Node.js** and **npm** for package management. To install the required dependencies, run:

```
npm install
```

This will install both frontend and backend dependencies as specified in the package.json file.

4. Set Up Environment Variables

To run the project locally, you will need to set up environment variables (e.g., for the database connection, authentication tokens, etc.). Create a .env file in the root directory of your project and add the following variables:

```
MONGODB_URI=mongodb+srv://sahilchhabra1551:v83T6YtW11Eg2oGo@cluster0.ga9vhez.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0

CLOUD_API_SECRET = tKmdFOD4gILsIiv79XdCYc2lFc0

CLOUD_NAME = dtbqj0ymg

CLOUD_API_KEY =793858121451954
```

5. Run the Development Server

Now, you can start the local development server for the frontend. In the project root directory, run:

```
npm run dev
```

This will start the React development server and you can access the website at http://localhost:5173 (or the port specified in the terminal output).

6. Run the Backend ()

Ensure that your backend server is running as well. Navigate to the backend directory and run:

```
Node Server.js
```

This will start the server and it will handle requests to your API, such as user authentication and package management.

7. Access the Application

Once the development server is running, you can access the application on your browser by visiting:

http://localhost:5173

8. Admin Login

To access the admin dashboard, you will need to authenticate with the admin credentials. Visit the **/admin** route to log in as an admin user.