

DestinEase Installation Manual

Table of Contents

1. Introduction
2. System Requirements
3. Installation Procedures
4. Cloning Git Repository
5. Firebase Integration
6. Starting the Server

Introduction

The DestinEase application is a modern web platform designed to help users discover and plan their perfect travel destinations.

It combines a React-based frontend with a robust Node.js/Express backend to deliver personalized travel recommendations, real-time weather integration, and flight search functionality.

Key Features:

- - Personalized travel recommendations based on user preferences
 - Real-time weather data integration
 - Flight search and pricing information
 - Secure user authentication system
 - Interactive destination cards with detailed insights
 - Smart filtering and sorting options

System Requirements

Ensure your system meets these requirements:

Hardware Requirements:

- Processor: 1.8 GHz or faster
- RAM: 4 GB minimum (8 GB recommended)
- Storage: 1 GB of available space
- Internet Connection: Active

Software Requirements:

- Node.js: Version 14.0.0 or higher
- npm: Version 6.0.0 or higher
- Git: Installed
- Web Browser: Modern browsers like Chrome, Firefox, Safari, or Edge
- Code Editor: Visual Studio Code (recommended)

Installation Procedures

Follow these steps to set up your development environment and install necessary dependencies.

Frontend Dependencies:

1. Install core React packages:
`npm install react react-dom react-router-dom`
2. Install UI components and utilities:
`npm install @lucide-react`
`npm install recharts`
`npm install tailwindcss postcss autoprefixer`
3. Install network and state management libraries:
`npm install axios cors`

Backend Dependencies:

1. Install server framework and middleware:
`npm install express body-parser cors`
2. Install authentication packages:
`npm install jsonwebtoken bcryptjs`
3. Install additional utilities:
`npm install axios dotenv`

Cloning Git Repository

To set up the project repository:

1. Open your terminal/command prompt.
2. Clone the repository:
`git clone https://github.com/htmw/2024F-Visionary-Techs`
`cd travel-recommendations-app`
3. Install project dependencies:

npm install

4. Verify the project structure:

- Ensure all React components are in src/components.
- Check for configuration files: package.json, tailwind.config.js, server.js
- Confirm the presence of required modules in node_modules.

Firestore Integration

The application can be configured to use Firestore for authentication. Follow these steps:

1. Firestore Project Setup:

- Visit Firestore Console.
- Create a new project.
- Enable Authentication services.

2. Configure Firestore in the Application:

Update the Firestore configuration in your code:

```
const firestoreConfig = {  
  apiKey: "your-api-key",  
  authDomain: "your-domain.firebaseio.com",  
  projectId: "your-project-id",  
  storageBucket: "your-bucket.appspot.com",  
  messagingSenderId: "your-sender-id",  
  appId: "your-app-id"  
};
```

3. Environment Configuration:

- Create a .env file in the root directory.
- Add Firestore credentials.
- Update authentication providers in the backend.

Starting the Server

Follow these steps to start the servers:

1. Start the Backend Server:

```
cd backend
```

```
node server.js
```

The backend server will run on port 5980 by default.

2. Start the Frontend Development Server:

```
cd frontend
```

```
npm start
```

The frontend server will be accessible at <http://localhost:3000>.

3. Verify Installation:

- Backend Health Check: Visit <http://localhost:5980/api/health>.
- Frontend Access: Go to <http://localhost:3000>.
- Test user authentication.
- Verify API connections (Weather API, Flight API, User Preferences).

4. Test the Application:

- Create a new user account.
- Set travel preferences.
- View personalized recommendations.
- Test flight search functionality.
- Confirm weather data display.