**CityWanderLust Project Documentation**

**Overview**

**CityWanderLust** is a web application that allows users to explore information about cities worldwide. The application provides a user-friendly interface for browsing city details, latitude, longitude, landmarks, and population data. Search Bar is provided to search a city of your choice. User can login using Sign in button. User should provide valid Email address for successful login. Internationalisation is provided to choose language and region of your choice.

**Table of Contents**

[1. Getting Started 2](#_Toc157542113)

[2. Project Structure 2](#_Toc157542114)

[3. Frontend 2](#_Toc157542115)

[4. Backend 2](#_Toc157542116)

[5. Dependencies 2](#_Toc157542117)

[6. Testing 3](#_Toc157542118)

[7. Deployment 3](#_Toc157542119)

[8. Scope 3](#_Toc157542120)

# Getting Started

To get started with the CityWanderlust project, follow these steps:

Clone the repository from GitHub: git clone <https://github.com/shivaniatos8/OttonovaChallenge>

Install dependencies:

Frontend: npm install

Backend: npm install

Run the development server:

Frontend: node server.js

Backend: node server.js

Access the application in your web browser at **http://localhost:3000**.

# Project Structure

The project is structured into two main directories: frontend and backend.

frontend: Contains the frontend codebase built using HTML, CSS, and JavaScript.

backend: Contains the backend codebase built using Node.js and Express.js.

# Frontend

The frontend of the City Wanderlust project is built using JavaScript, CSS, HTML a popular. The frontend directory structure is as follows:

views: Contains the static assets and HTML template for the application. Contains the source code for the frontend application.

index.html: Entry point for rendering the application.

Server.js: contains Api endpoint.

# Backend

The backend of the City Wanderlust project is built using Node.js and Express.js, providing RESTful API endpoints for serving city data to the frontend. The backend directory structure is as follows:

server.js: Contains controller functions for handling API requests and business logic. It is the main entry point for the backend application.

# Dependencies

The City Wanderlust project has the following dependencies:

Frontend: JavaScript, HTML and CSS.

Bootstrap: Frontend framework for responsive design.

Backend:

Node.js: JavaScript runtime environment.

Express.js: Web application framework for Node.js.

# Testing

The City Wanderlust project uses Jest.

# Deployment

The City Wanderlust project can be deployed to various platforms such as AWS, or Azure. Ensure that environment variables are configured for production deployment.

# Scope

The scope of the City Wanderlust website encompasses the features and functionalities it offers to users. Below is the scope of the website

* City Information:

Display detailed information about various cities worldwide. Include data such as city name, country, population, and landmarks.

* Search Functionality:

Allow users to search for cities by name. Provide auto-complete suggestions while typing in the search bar.

* Responsive Design:

Ensure the website is responsive and works seamlessly across different devices and screen sizes.

* User Interaction:

Enable users to click on city cards to view detailed information about a specific city.

Include interactive elements such as buttons and forms for user engagement.

* User Authentication :

Implement user authentication functionality to allow registered users to log in and access additional features.

Include user registration and login forms with appropriate validation.

* Internationalization (Optional):

Provide options for users to select their preferred language and display city information accordingly.

Include flags or language icons for language selection.

* Customization (Optional):

Allow users to customize their experience by choosing preferred themes or color schemes.

Provide options for users to set preferences such as default language or currency display.

* Integration with APIs (Optional):

Integrate with external APIs to fetch additional data such as weather information, local events, or points of interest in cities.

Display relevant information alongside city details to enhance user experience.

* Social Media Integration (Optional):

Include social media sharing buttons to allow users to share city information with their networks.

Integrate with social media APIs for features such as social login or user-generated content sharing.

* Admin Panel (Optional):

Implement an admin panel for managing city data, user accounts, and website settings.

Allow administrators to add, edit, or delete cities and their information.

* Analytics and Reporting (Optional):

Integrate with analytics tools to track user behaviour, website traffic, and user interactions.

Provide reporting features for administrators to analyse website performance and user engagement metrics.

Conclusion: The scope of the CityWanderLust website can be adjusted based on project requirements, timelines, and available resources. Optional features can be included or excluded based on priority of a developer.