WebX Lab CA - Prerequisites Document

Name of Student	**Rohan Lalchandani**
Class Roll No	**D15A_25**
D.O.P.	
D.O.S.	
Sign and Grade	

TITLE: Smart Travel Planner

INTRODUCTION

The Smart Travel Planner is a comprehensive Flask-based web application designed to streamline and enhance the travel planning experience. The application features a robust architecture with distinct components handling user authentication, trip management, destination exploration, and social interactions.

At its core, the application utilizes MongoDB for data persistence, managing various collections including users, destinations, trips, and travel tips. The backend is built with Flask, running on port 5000, and implements a well-structured MVC (Model-View-Controller) pattern. The models are clearly defined in models.py, handling entities such as User, Trip, Destination, and JournalEntry with proper data validation and relationships.

Key features include comprehensive trip planning capabilities, allowing users to create detailed itineraries, track expenses, and maintain travel journals. The social aspects of the platform enable trip sharing and collaboration through a well-implemented invitation system. The application also integrates weather information for destinations and provides curated travel tips categorized by topics such as packing, safety, and local customs.

SYSTEM REQUIREMENTS

1. Hardware Requirements
    - Processor: Intel Core i5 or higher
    - RAM: Minimum 8GB (Recommended: 16GB for smooth performance)
    - Storage: At least 10GB free disk space
    - Operating System: Windows 10/11, macOS, or Linux

2. Software Requirements
    - Node.js (v18.16.1 or later) – Required for running the React frontend
    - npm (v9.5.1 or later) – For managing frontend dependencies
    - Python (3.11.9 or later) – Required for Flask backend
    - MongoDB Atlas – Cloud-based NoSQL database
    - VS Code – Recommended IDE for both frontend and backend development
    - Postman – For API testing

TECHNOLOGY STACK

1. Frontend:

- HTML5, CSS3
- Bootstrap CSS Framework (Dark Theme)
- Font Awesome Icons
- JavaScript
- Custom CSS for styling

2. Backend:

- Python 3.11+
- Flask 3.1.0 (Web Framework)
- Flask Extensions:
    - Flask-Login (Authentication)
    - Flask-SQLAlchemy (ORM)
    - Flask-PyMongo (MongoDB Integration)
    - Flask-WTF (Forms)
    - Werkzeug (WSGI Utilities)

3. Database:

- MongoDB (Primary Database)

4. APIs & Services:

- OpenWeather API (Weather Information)
- SendGrid (Email Services)

5. Development & Deployment:

- Gunicorn 23.0.0 (WSGI Server)
- Replit (Development & Hosting Platform)
- Git (Version Control)

6. Additional Libraries:

- Requests 2.32.3 (HTTP Client)
- WTForms 3.2.1 (Form Validation)
- Email-validator 2.2.0
- psycopg2-binary 2.9.10 (PostgreSQL Adapter)

SETUP INSTRUCTIONS

1. Install Node.js and NPM
   Download and install Node.js from Node.js Official Website.
   Verify installation:
   node -v
   npm -v

2. Install Python and Flask
   Download and install Python from the Python Official Website. Verify installation:
   python --version pip -version
   Install Flask and required dependencies:
   pip install flask flask-cors pymongo

3. Setup MongoDB
   Use MongoDB Atlas for cloud storage or install MongoDB locally from MongoDB Official Website.
   If using a local setup, start MongoDB using:
   mongod --dbpath /path/to/data

PROJECT STRUCTURE

Backend	Frontend
├── app.py ├── config.py ├── models.py ├── forms.py ├── db.py ├── weather.py └── main.py	├── templates/ │ ├── base.html │ ├── index.html │ ├── dashboard.html │ └── profile.html │ ├── static/ │ ├── css/ │ │ └── custom.css │ ├── js/ │ │ └── main.js │ └── uploads/ │ └── data/ ├── destinations.json ├── trips.json

CONCLUSION

This document provides the necessary prerequisites for setting up the Smart Travel Planner project, including hardware/software requirements. Ensuring all dependencies and configurations are correctly set up will help in running the project efficiently.