**Kwantlen Polytechnic University**

**Melville School of Business**

**Computer Information Systems Department**

**Travel Guide System with AI-Powered Itinerary Generator**

**Phase 3 (Design)**

**Submitted By**

Dakshdeep Singh (100452228)

Anmoldeep Singh (100450649)

Manshu Goyal (100447957)

Harjot Singh (100452598)

**Supervised By**

Prof. Mayyadah Al-Ani

**June 29, 2025**

**Table of Contents**

Contents

[1. Introduction 2](#_Toc202043125)

[2. Project Overview and Goals 3](#_Toc202043126)

[3. System Architecture 3](#_Toc202043127)

[4. Component Breakdown 4](#_Toc202043128)

[5. Data Design 5](#_Toc202043129)

[6. UI Design 6](#_Toc202043130)

[7. Technology Stack & Justification 6](#_Toc202043131)

[8. Security Design 7](#_Toc202043132)

[9. Additional Information 8](#_Toc202043133)

# Introduction

In the world that is increasingly digital, travel has become more available and irresistible than ever. Many people are visiting countries in the world due to the cheap transportation, the availability of numerous resources on the Internet, and the continued expansion of social media. Nevertheless, what is practiced to plan is very old-fashioned and useless, in spite of the advancement of technology in the travel industry. To access the incomplete information about destinations, most travellers still use generic travel websites, blogs, or vlogs. These sources rarely provide personal recommendations depending on specific preferences, budget, and schedule of a traveller. Due to this, one will always end up being too much information or stuck with lousy plans that cannot be associated with the goals of the vacation at hand.

To address this growing challenge, we propose the development of a system that produces an itinerary based on an artificial intelligence algorithm in what can be referred to as a *Travel Guide System* that features an *Artificial Intelligence Algorithm-based Itinerary Generator*. The purpose of this project is to develop a web application that presents an interesting, entertaining and efficient method of planning trips. The system can generate customized travel itineraries in real time, synthesizing artificial intelligence-based parameters supplied by the user, such as location, length, travel dates, interests and budget. Combining AI and modern online technology, the proposed approach will transform travel planning into an easy, personalized and automated experience, eliminating all cumbersome and manual processes.

The system is based on an AI engine that uses models of natural language processing, like the ones used by OpenAI, GPT, or Google PaLM. These models, in real time, produce smart and relevant itineraries depending on what the user feeds. The frontend presents an easy, receptive experience that is compatible with numerous gadgets due to the structure used, ReactJS, and styling, Tailwind CSS. In order to enhance their travel plans, the users can review day-by-day schedules, interactive maps, and intelligently selected information. Admin-controlled blogs also allow posting articles with professional travel recommendations and characteristics of a destination, which makes the web resource entertaining and instructive.

This Software Design Document (SDD) describes the development of the planned system in detail. It outlines the technology stack, user interface design, data models, how components work, system architecture and security issues. To ensure successful design, which can deliver on user needs and the academic requirements, research and comments collected in the earlier stages of the project are also included in the document. Finally, the SDD demonstrates how innovative AI technology can be used to develop a user-centred, scalable, and efficient travel planning tool.

# Project Overview and Goals

The aim of the travel guide system with an AI-powered itinerary generator would be to transform the process of travel planning. Conventional planning of vacations involves wastage of infinite time on the internet, reviewing, comparing prices and plotting things manually. Apart from being time-consuming, this process gives irregular and often inappropriate itineraries. To enable us to take care of this problem, we have a solution, whereby we will use artificial intelligence to automate the creation of the itineraries whilst ensuring that the recommendations we offer are relevant to the interests of the traveller.

The users of the platform will begin by entering critical data of the trip, including the planned destination, the dates of arrival and departure, the duration of the visit, the desired type of trip (adventure, culture or leisure), and the amount of money to be spent on the trip in general. The system applies this data to AI models to come up with an entire and personalized plan complete with recommended activities, estimated budgets, and travel itinerary, as well as a day-to-day outline of the journey. These capabilities enable users to draw informed decisions, update plans before departing and get a visual picture of their journey.

Among the goals of the project, we will develop an interface, which would be friendly and easy to use and provides interactive feedback or recommendations on the places of interest to see. The other objectives are to create backend services to deal with AI interaction, user data analysis, and provide responsive and safe support. Companies will be able to make their blogs into an e-learning platform by creating a location guide, safety travel tips, and cultural insights that the administrators can build to spotlight a specific topic in an integrated blog platform.

Travel Guide System is designed to simplify the travel planning process, eliminate the presence of various third-party services, and offer customers a one place where they can receive well-thought-out personalized intelligent travel recommendations by integrating user-provided data and AI-generated decision-making.

# System Architecture

To ensure the Travel Guide System can be scaled, maintained, and with a proper flow of data, its architecture incorporates a tier-based architecture that further splits the responsibilities of every component. Its developers are also free to scale up and manage the system by merely augmenting it, though not at the expense of its overall design.

Depending on the technology, the presentation layer is developed with the help of Tailwind CSS and ReactJS. This layer controls all user interactions, such as in the form of input, itinerary, maps and blog material. It will be responsible for collecting user information and presenting the AI-generated content in a logical and pleasing interface.

The application layer is enabled by Node.js and Express.js. It is a connector between the AI services and the frontend. This layer does this before returning results to the frontend; it includes business logic processing, input validation, route control and formatting. It also provides support with integrating several APIs, such as Google Maps and AI models.

Through the AI integration layer, external AI APIs are communicated with directly, like those of Google PaLM or OpenAI. This layer processes the preferences of the user and transmits them to the AI model, which processes the request and delivers a personalized plan for the trip.

The inputs of the users and responses of the AI during an active session are saved in the data layer, which is supposed to be used as a temporary in-memory component of storage. Phase 1: There is no need at this stage to have persistence DB storage and log in to the system continuously. This makes the program fast and light and does not betray the privacy of the user.

The security and performance layer consists of the use of TLS encryption, input validation, error handling, and rate limitation. Such protection ensures that the data of the users remains safe, that the system can cope with a load, and that there is no way to protect against common threats, such as malicious input or bots.

The interaction of these layers and sections to provide a smooth user environment can be observed in the above-provided system architecture diagram that was provided earlier.

# Component Breakdown

The six major components of the Travel Guide System are supposed to execute different duties in the greater structure.

The UI Component serves a front-facing interface in which a user enters their preferences in regard to the trip and sees the generated itineraries. This element, developed using ReactJS, ensures that the interaction of the user is fast, convenient, and works on different devices.

The backend server handles input validation management, data processing of the users, coordination of services and consistent and organized outputs of the backend server. It is created based on Node.js and Express.js, providing a solid backbone for multiple simultaneous requests.

The Map Integration component displays the route, landmarks, as well as locations visually through the Google Maps API. This adds the geospatial element to the itinerary and helps consumers understand the flow of the journey.

The Blog Manager provides administrative users with an admin interface to post, edit and manage entries on the blog. The module ensures that relevant travel tips, news and advertising can be posted in real time.

Finally, since the Session Data Handler does not need user accounts or long-term storage, any preferences and other results the browser encounters are retained through the browser session by temporarily keeping user session data.

The modularity of every component provides the possibility to easily develop and grow, and enables updating or replacement independently.

# Data Design

The data model of the system Travel Guide System has five basic elements, that is, User, Itinerary, Destination, Activity, and Preferences. These entities follow a logical evolution of data entered into the output by an AI.

A user goes to a planning portion and puts in such details as what he or she is interested in, where he or she wants to visit, and the planned duration. The system then creates an itinerary for the session based on this data. Depending on the length or multiple cities involved in the tour, each tour can cover several destinations. Depending on the preferences that the user states, individual trips are grouped into activities such as local events, sightseeing, and dining.

This information stays in Phase 1 and is discarded after the session. The data architecture can be extended during later iterations to include the storage in a persistent form, so users can make some accounts and see past itineraries.

The data dictionaries include some variables such as CITY (String), DURATION (Integer), TRAVELERS (Integer), ID (Integer to authenticate the user as an admin), and PASSWORD (String to log in as an admin). The input and management logic of the system is comprised of these variables that form its core.

**Table 5.1: Sample Data Dictionary**

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| CITY | String | City the user wants to travel to |
| DURATION | Integer | Number of days for the trip |
| TRAVELERS | Integer | Number of people travelling |
| ID | Integer | Admin ID (for blog management) |
| PASSWORD | String | Admin login password |

# UI Design

The user interface was designed to keep responsiveness, accessibility, and clarity in mind. The user interface was tested by the WCAG accessibility standards and structured with regard to the core usability principles through Figma prototyping.

The entry point is the Home Page, where users can type in their trip options. To facilitate usage, the layout is simplified and has dropdown menus, date pickers and sliders to select the budget.

The Results Page displays the AI-generated itinerary with the help of Google Maps visual representations that appear in the form of a daily schedule. Users have an opportunity to observe the daily schedule, understand the way, and make changes when needed.

The Blog Page includes travel tips, cultural outlook, and articles by administrators. The users could find more relevant material as posts are divided into popularity or genre.

Trusted users get the ability to create, edit as well and delete blog posts through the exclusive content control system provided through the Admin Panel. Authentication and secure access to blog content only allow government-sanctioned users to manage this content.

Accessibility features include high contrast schemes, keyboard navigation, and semantic HTML to screen readers. The interface is responsive to a variety of platforms on tablets, smartphones, and PCs; their interface has been tested.

# Technology Stack & Justification

It uses a state-of-the-art, lightweight, scalable technological stack, custom-made to work in production, MVP, and academic environments.

They are implemented in the frontend that consists of Tailwind CSS and ReactJS. By utilizing components of React, modular design becomes possible, and Tailwind ensures visibly responsive styling with minimal overhead.

Node.js and Express.js give us the power of the backend. Such technologies are perfect for the development of scalable APIs and the control of asynchronous requests.

Integration of AI is carried out by using either OpenAI GPT models, depending on API availability or Google PaLM. These models brought their natural language processing abilities to the system and enabled it to present relevant and humanistic trip itineraries.

The seamless development process is guaranteed by the hosting of the application on Vercel, which provides continuous integration and deployment directly out of GitHub. GitHub takes care of version control, helping to track changes and work in a team.

Figma, an ecosystem tool of UI/UX design, enables an easy transition between design and development, user testing, and rapid prototyping.

This tech stack is ideal for building a Minimum Viable Product (MVP) and for academic evaluation purposes.

**Table 7.1: Stack Summary**

|  |  |  |
| --- | --- | --- |
| Section | Technology | Justification |
| Front-End | ReactJS + Tailwind | Modular, fast UI with responsive layout |
| Back-End | Node.js + Express | Scalable backend with efficient API management |
| AI Integration | Google PaLM / OpenAI | NLP-based personalized itinerary generation |
| Hosting | Vercel | Supports CI/CD for GitHub projects |
| Version Control | GitHub | Enables collaboration and code version history |
| Design Tool | Figma | Streamlined UI/UX prototyping and layout planning |

# Security Design

To ensure assurance to users, reliability, as well as security of data, the system relies on several layers of security at each stage, including the prototype.

TLS encryption keeps all client-server communications encrypted and protects against manipulation and data interception. There are input validation processes at an application level to ensure only anticipated and clean data is manipulated in an application.

The limiter protects against abuse by automated free runners or excessive use of APIs. It also ensures that, regardless of the amount of traffic, the system will remain accessible and responsive.

Administrative blog administration is secured with a simple authentication layer of limits access to the administration by certain users. Passwords are salted and stored in a secure manner, even in the case of a temporary module or admin module.

The application does not collect and store personally identifiable information (PII) according to the best privacy practices. Once the user exits the session, all the data is temporary and is disposed of. The system also adheres to GDPR-like recommendations, as it does not use third-party trackers and persistent cookies.

In case of the malfunction of the AI engine or if it provides an error, the fallback processes may be taken in order to ensure that the system continues to generate insightful answers. This will ensure that it is continually operational and gives the user a good experience.

# Additional Information

Improvements in the future will happen, although at the moment they are also on a minimal viable product. These involve the addition of third-party booking APIs of hotels and flights, allowing users to save and share their itineraries, integrate real-time functionality such as weather information or notifications of local events and allow users to log in and register.

By documenting and following this document in detail, the development team will know precisely how to develop an innovative, well-scaled, and customer-engaging artificial intelligence-driven trip guide system.