

Milestone 3: GUI Development & Prompt Engineering Report

Project: Travel Agency Management System

Team: Jana Farghal, Ali Osman, Abdulrahman Moustafa

1. Objective

The objective of Milestone 3 was to build a Graphical User Interface (GUI) that connects seamlessly with the backend database. This phase utilizes prompt engineering techniques to develop a user-friendly website that satisfies the functional requirements.

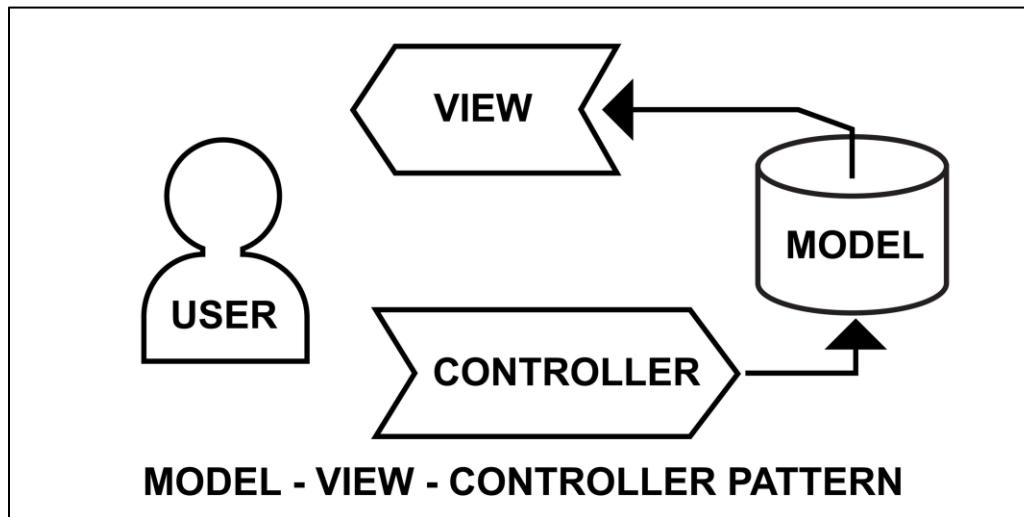
2. System Architecture (MVC)

The system is built using the Model-View-Controller (MVC) design pattern⁷:

Model: The relational database (Supabase/PostgreSQL) designed in Milestone 2.

View: The front end developed using HTML, CSS, and JavaScript.

Controller: The backend integration (Node.js/Express) that manages API calls and database connections.



3. GUI Design and Functionality

The interface allows users to perform core database operations as defined in the project requirements:

View Data (REQ_002): A dashboard allowing customers to view available trips.

The screenshot shows a web browser window for 'Wanderlust Travel' at the URL 127.0.0.1:5500/CCAS-4.3-Final-Project-main/frontend/GUI.html#. The page has a purple header with the logo and navigation links: Home, Manage Trips, Manage Bookings, View Trips, and View Bookings. The main content area is titled 'Available Trips' with the sub-instruction 'Browse all available travel packages'. Below this is a search bar labeled 'Search trips...'. A table lists three travel packages with columns: ID, DESTINATION, DURATION, PRICE, START DATE, END DATE, CAPACITY, and DESCRIPTION. The data is as follows:

ID	DESTINATION	DURATION	PRICE	START DATE	END DATE	CAPACITY	DESCRIPTION
1	Paris, France	7 days	\$1299.00	Mar 15, 2026	Mar 22, 2026	20	Explore the city of lights
2	Tokyo, Japan	10 days	\$1899.00	Apr 10, 2026	Apr 20, 2026	15	Experience Japanese culture
3	Bali, Indonesia	5 days	\$899.00	May 1, 2026	May 6, 2026	25	Tropical paradise

Insert Data (REQ_005): Admin forms to add new trips to the database.

The screenshot shows a web browser window for 'Wanderlust Travel' at the URL 127.0.0.1:5500/CCAS-4.3-Final-Project-main/frontend/GUI.html#. The page has a purple header with the logo and navigation links: Home, Manage Trips, Manage Bookings, View Trips, and View Bookings. The main content area is titled 'Manage Trips' with the sub-instruction 'Add, update, or delete travel packages'. Below this is a form titled 'Add / Update Trip' with fields for Destination, Duration, Price, Start Date, End Date, Max Capacity, and Description. At the bottom are 'Save Trip' and 'Clear Form' buttons.

Form Fields:

- Destination *: e.g., Paris, France
- Duration (Days) *: e.g., 7
- Price (\$) *: e.g., 1299.99
- Start Date *: dd/mm/yyyy
- End Date *: dd/mm/yyyy
- Max Capacity *: e.g., 20
- Description:
Enter trip description...

Buttons:

- Save Trip
- Clear Form

Update Data (REQ_006): Interfaces to modify existing trip records, such as updating prices or dates.

The screenshot shows a web browser window for 'Wanderlust Travel'. The title bar displays the URL '127.0.0.1:5500/CCAS-4.3-Final-Project-main/frontend/GUI.html#'. The navigation bar includes links for Home, Manage Trips, Manage Bookings (which is highlighted in blue), View Trips, and View Bookings. The main content area has a heading 'Manage Bookings' and a sub-instruction 'Add, update, or delete customer bookings'. Below this is a form titled 'Add / Update Booking' with fields for Customer Name, Email, Phone, Select Trip, Number of People, Booking Date, Status, Total Amount, and Special Notes. At the bottom are 'Save Booking' and 'Clear Form' buttons.

Delete Data (REQ_006): Options for administrators to remove trips from the system.

The screenshot shows a 'All Trips' page. At the top is a search bar with placeholder text 'Search trips by destination...'. Below it is a table with columns: ID, DESTINATION, DURATION, PRICE, START DATE, END DATE, CAPACITY, DESCRIPTION, and ACTIONS. The table contains three rows of trip data:

ID	DESTINATION	DURATION	PRICE	START DATE	END DATE	CAPACITY	DESCRIPTION	ACTIONS
1	Paris, France	7 days	\$1299.00	Mar 15, 2026	Mar 22, 2026	20	Explore the city of lights	<button>Edit</button> <button>Delete</button>
2	Tokyo, Japan	10 days	\$1899.00	Apr 10, 2026	Apr 20, 2026	15	Experience Japanese culture	<button>Edit</button> <button>Delete</button>
3	Bali, Indonesia	5 days	\$899.00	May 1, 2026	May 6, 2026	25	Tropical paradise	<button>Edit</button> <button>Delete</button>

4. Prompt Engineering & API Utilization

We utilized GPT-3.5/4 (OpenAI API) to assist in generating the GUI and refining the design.

4.1 Stages of Prompt Refinement

We followed an iterative prompting process to ensure quality and accuracy:

Stage 1: Zero-Shot (Skeleton Structure)

Goal: To generate the basic HTML and CSS structure for the system's core pages.

Prompt: "Generate a clean, professional HTML and CSS skeleton for a Travel Agency Management System dashboard. Include a navigation bar with links for 'Home', 'Manage Trips', and 'Handle Bookings'. Use a grid layout to display trip packages as cards with placeholders for Trip Name, Destination, and Price."

Stage 2: Refined Prompting (Responsive Design & Functional UI)

Goal: To ensure the system is accessible via modern web browsers (REQ_015) and features a simple, easy-to-use interface (REQ_019).

Prompt: "Update the previous CSS to make the layout fully responsive for mobile and desktop using Flexbox. Additionally, create a modal popup form for 'Add New Trip' that includes input fields for Trip Name, Destination, Start Date, End Date, Price, and Available Seats, matching my SQL schema."

Stage 3: Validation & Technical Integration (Fetch API)

Goal: To connect the frontend forms to the Node.js backend and Supabase database using the Fetch API.

Prompt: "Write a JavaScript function using the Fetch API to handle the 'Add New Trip' form submission. The function should send a POST request to 'http://localhost:3000/api/trips' with the form data in JSON format. Ensure it handles success and error responses by alerting the user, satisfying the requirement for system feedback."

5. Requirement-Query Matrix (Milestone 3)

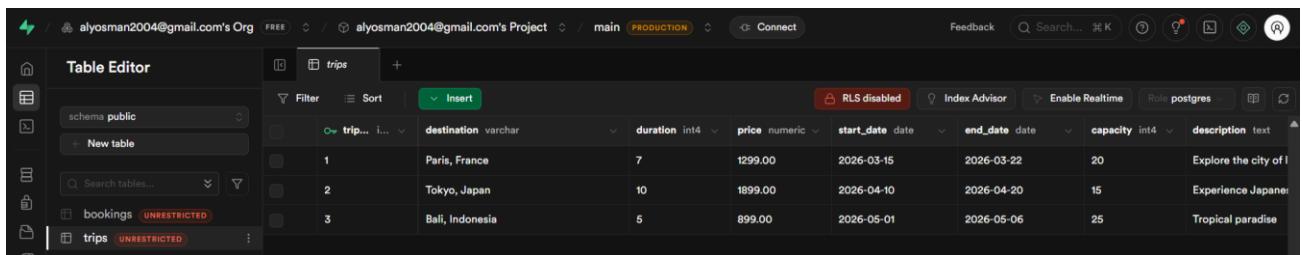
This matrix matches the functional requirements from Milestone 1 with the SQL queries from Milestone 2 now implemented in the GUI.

Requirement ID	Description	Query ID
REQ_001	User registration and login 22	QUR_001 23
REQ_002	View available trips 24	QUR_005 25
REQ_005	Add new trips (Admin) 26	QUR_002 27
REQ_006	Update/Delete trips (Admin) 28	QUR_003 / QUR_004 29292929

6. Implementation

To confirm the integration between the GUI and the database, we verified the following:

Database Sync: Data entered in GUI forms is correctly stored in the Supabase tables.



The screenshot shows the Supabase Table Editor interface for the 'trips' table. The table has columns: destination (varchar), duration (int4), price (numeric), start_date (date), end_date (date), capacity (int4), and description (text). There are three rows of data:

	destination	duration	price	start_date	end_date	capacity	description
1	Paris, France	7	1299.00	2026-03-15	2026-03-22	20	Explore the city of light
2	Tokyo, Japan	10	1899.00	2026-04-10	2026-04-20	15	Experience Japanese culture
3	Bali, Indonesia	5	899.00	2026-05-01	2026-05-06	25	Tropical paradise

7. Conclusion

Milestone 3 successfully transformed the conceptual database design into a functional web application. The use of prompt engineering accelerated the development of a user-friendly interface that adheres to the project's security and performance requirements.