

# VOYAGE

YOUR PERFECT TRIP

## **Final Project**

By:

Roni Shapira (209122456)

Neta Werner (315326934)

Third year, B.Sc in Computer Science  
Afeka College of Engineering

## Table of Contents

<b><u>.1</u></b>	<b><u>Executive Summary</u></b>	<b>4</b>
<b>2.</b>	<b>Preliminary Planning</b>	<b>4</b>
a.	Problem Description	4
b.	Project Goal	5
c.	Project Objectives	5
d.	Measures Of Success	5
e.	Market Survey - Comprehensive Market Survey	7
f.	Literature review	10
<b>3.</b>	<b>High-Level Design</b>	<b>16</b>
a.	System & Technological Alternatives:	16
b.	System Requirements	18
c.	Use Cases	22
d.	ERD	24
e.	Flow Diagram	25
f.	API Definition	26
g.	Initial Screens Design	30
h.	Risk-Management	32
i.	Engineering Challenges	33
j.	Limitations and Mitigation Strategies	34
k.	Gant Charts	36
<b>4.</b>	<b>Products</b>	<b>37</b>
a.	Git code link:	37
b.	Final System Design	37
c.	Final Screens Design	39
d.	Software Tools	43
e.	Tests Coverage	45
<b>5.</b>	<b>Summary and Conclusion</b>	<b>45</b>
a.	Compliance	45
b.	Discussion	49
c.	Future Suggestions	50

d. Conclusion.....	52
--------------------	----

# 1. Executive Summary

**VOYAGE** is our web application designed to streamline vacation planning.

We built VOYAGE by combining prompt engineering with generative AI to create a tool that constructs and validates personalized itineraries based on end-user preferences.

This eliminates the need for extensive research – simply provide VOYAGE with your travel desires, and it will craft a verified itinerary packed with local experiences and hidden gems.

Our passion for supporting small businesses extends into VOYAGE's core functionality.

By joining our platform, small businesses from various travel sectors can promote their offerings through a seamless recommendation system that connects them with potential customers who align with their services.

## 2. Preliminary Planning

### a. Problem Description

Our project seeks to answer two main problems:

- 1) **Time-consuming Vacation Planning:** Currently, planning a vacation often involves extensive research, and sifting through countless websites and reviews. VOYAGE tackles this issue by leveraging generative AI and helps untheological users enjoy its benefits.
- 2) **Local Business Visibility:** Small and local businesses in the travel industry often struggle to compete with large, established players who dominate online advertising. VOYAGE provides a solution by offering these businesses a platform to promote their unique offerings. Through a seamless recommendation system, VOYAGE connects them with travelers actively seeking authentic experiences, ensuring their content reaches the right audience.

## **b. Project Goal**

- 1) **User goal:** Our website allows users to design personalized vacations based on their preferences and choices.
- 2) **Publishers (business client) goal:** Via our interface, you can advertise your content (restaurants, hotels, attractions) to potential customers.

## **c. Project Objectives**

- 1) Using guiding questions, the UI will assist the end user in generating the ideal prompt query, enabling the generative AI to propose the best trip plan that matches the user's preferences.
- 2) Create a simple and clear user interface, while providing a short response time
- 3) The API will provide a clear, reasonable, and reliable result that matches the user's request.
- 4) The application allows business- customers to publish their content in a way that seemingly integrates into the trip options.

## **d. Measures Of Success**

- 1) **Using guiding questions, the UI will assist the end user in generating the ideal prompt query, enabling the generative AI to propose the best trip plan that matches the user's preferences:**
  - Number of configurable parameters: Count the number of user-adjustable variables within the itinerary generation process (travel dates, budget, activity types, etc.).
  - Well-designed prompt template: define a prompt template to inject the user's properties to get reasonable and accurate answers from the generative AI.
  - Average number of itinerary variations: Track the average number of alternative suggestions offered for each element (accommodation, transportation, landmarks) within a generated itinerary.

**2) Create a simple and clear user interface, while providing a short response time:**

- Number of clicks to create an itinerary: Count the average number of clicks required to complete an itinerary from start to finish, indicating efficiency and simplicity.
- Number of user interface elements: Minimize the number of buttons, menus, and options to avoid overwhelming users.
- Average API response time under load: Test API performance under various user loads and aim for a consistent response time below 2 seconds.

**3) The API will provide a clear, reasonable, and reliable result that matches the user's request:**

- Third-party data validation: Conduct periodic audits to ensure data accuracy and adherence to quality standards if relying on external data sources.

**4) The application allows business- customers to publish their content in a way that integrates into the trip options:**

- Each generated trip will include at least two recommendations according to the data in the DB.
- Build a payment method that relies on buying credits that the DB manages.

## e. Market Survey - Comprehensive Market Survey

Name	Description	Relevant for comparison
<b>TripHobo</b>	Allows users to plan their trips with customizable itineraries, including attractions, accommodations, and activities.	The customer has to <u>decide on a specific place</u> , like cities, there is <u>no reference to budget/preferences</u> .
<b>RoutePerfect</b>	Helps users plan their trips by providing personalized itineraries based on their preferences, including destinations, attractions, and activities.	A good site for comparison, there are <u>many features</u> . The <u>interface is messy</u> and cumbersome
<b>Sygie Travel</b>	Offers trip planning with a focus on attractions, landmarks, and personalized itineraries.	A <u>map with attraction markings</u> , less a system for building a trip
<b>Roadtrippers</b>	Ideal for planning road trips, providing users with customizable routes, attractions, and points of interest	Seems to be <u>suitable for use within the US only</u> , refers to <u>road trips, caravan trips</u>
<b>Visit A City</b>	Helps users plan city-based trips with personalized itineraries, including attractions, dining, and activities.	A nice option for comparison, also focused on a <u>specific city</u> .
<b>Travefy</b>	A collaborative trip-planning platform that allows users to create and share itineraries with others.	Builder of a travel diary does not build you a trip.  like an electronic planner that is adapted for trips, you enter the information

After conducting the comprehensive market survey, we have chosen to showcase three websites/applications that can serve as a good source for comparison to our project.

Measures Of Success		Weight	VOYAGE	Trip Hobo	Visit A City	Route Perfect
Using guiding questions the UI will assist the end user to generate the ideal prompt query that will enable the generative AI to propose the best trip plan that matches the user's preferences.	Number of configurable parameters	20%	2	1	0	1
	Well-designed prompt template	10%	2	1	1	1
	Average number of itinerary variations	10%	1	1	2	2
Create a simple and clear user interface, while providing a short response time	Number of clicks to create an itinerary	10%	1	1	2	1
	Number of user interface elements	10%	2	1	1	0
	Average API response time under load	10%	1	0	1	1
The API will provide a clear, reasonable, and reliable result that matches the user's request	Third-party data validation	5%	1	1	1	1
The application allows business- customers to publish their content in a way that integrates into the trip options	Each generated trip will include at least two recommendations if possible according to the data in the DB.	10%	2	1	1	1



Measures Of Success		Weight	VOYAGE	Trip Hobo	Visit A City	Route Perfect
	Build a payment method that relies on buying credits that are managed by the DB.	15%	1	1	1	1
Calc score: $\sum_1^n (W * S)$			1.4	0.9	1	1

\*The sections marked in yellow cannot be measured on the websites/apps. This is why we gave them all the middle grade

1. None of the sites/apps have access to documentation.

## **f. Literature review**

During our research in the preparation process for the project, we used both professional and academic literature to examine the possible ways to implement our application and meet the objectives we set for ourselves. Below are the sources we used, the added value, the questions, and the conclusions we drew from them.

We began our research by investigating two challenges we had previously assumed and attempted to provide solutions for them through our application:

(Both of our assumptions, are based on the premise that people who can afford it themselves, would like to travel aboard or inboard but prefer not to waste their time on planning their trips – they prefer personalized, well-planned trips that are created in a “one-click”)

- 1) The primary hurdle faced by the tourism industry is the fact that it is dominated by large players such as Booking.com, restaurants, hotels, and attractions. These organizations have the financial resources to invest in expensive advertising platforms like Google Maps and others, which makes it difficult for smaller players to compete [1] + [2]. Our platform allows small players to pay only if their ads are seen by potential clients who already showed an interest in their domain.
- 2) The tourism industry is being impacted by Artificial Intelligence, and generative AI has the potential to enhance customer experiences, provide forecasting and operational efficiency, and contribute to sustainability in the industry. However, many private individuals lack the technical skills to use generative AI tools effectively, and the "black box" nature of AI models makes it difficult for them to understand how to prompt the chatbot. This can lead to irresponsible or unsustainable use of AI in tourism. To bridge this gap, our app will act as a mediator between the end-users and the generative AI tools, making it easier for all parties to benefit from AI responsibly and sustainably [3] + [4].

Once we confirmed our two fundamental assumptions, we continued our research by delving deeper into the subject matter. We began gathering academic materials and reading blogs and tutorials that could assist us in enhancing our application in the three key domain areas we had identified:

### **1) Best practices for UI (user interface) implementation:**

During our research on the implementation of the UI, we focused on one article [5] that presents the various General Principles of User Interface Design and their relevance for present-day web interfaces. The study mentions these aspects as major attention needed while implementing web application UI:

- Aesthetically Pleasing: Visually appealing design attracts users and conveys information.
- Clarity: Interface elements and wording should be understandable and unambiguous.
- Compatibility: Users should feel comfortable and in control within the interface.
- Comprehensibility: The interface should be easy to understand and follow.
- Configurability: Users should be able to personalize the interface to their preferences.
- Consistency: Similar elements should have similar appearance, function, and behavior.
- Control: Users should feel in charge and able to perform actions easily.
- Directness: Tasks should be performed directly with minimal steps and visible options.
- Efficiency: Eye and hand movements should be minimized, and navigation should be efficient.
- Familiarity: Use familiar language and real-world concepts for intuitive interaction.
- Flexibility: The interface should adapt to individual user preferences and interaction styles.

- Forgiveness: The interface should handle user errors gracefully and guide correction.
- Predictability: User actions should have consistent and expected outcomes.
- Recovery: Users should be able to undo or redo actions and recover from mistakes easily.
- Responsiveness: The interface should respond quickly to user input and provide feedback.
- Simplicity: Keep the interface clutter-free, prioritize common functions, and hide complex ones.
- Grouping: Structure elements with meaningful titles to aid information recall and navigation.

## **2) The fact that generative AI tools' reliability and credibility are questionable:**

One of the major concerns in the ethics section regarding the challenges posed to humanity by the increasing use of generative AI tools is over-reliance [6]. The apparent convenience and power of generative AI tools can lead users to rely too heavily on them, causing them to trust the answers they provide without question. In contrast to traditional search engines, which provide multiple information sources for users to make personal judgments and selections, generative AI generates specific answers for each prompt. While tools like chatGPT, Bard, and others can increase efficiency by saving time and effort, users may get into the habit of adopting the answers without rationalization or verification. Therefore, all users need to have AI literacy. Users should not blindly trust the answers provided by generative AI but should instead go through verification procedures before adopting them.

To overcome this challenge, our application will implement a third-party data validation after getting the generative AI response.

## **3) The right way to prompt:**

As mentioned earlier in the lecturer review, one of the fundamental assumptions behind the application idea is the need for a mediator between end-users and

generative AI tools to maximize the tool's potential. To generate optional raw itinerary programs using the generative AI API, we need to have a deep understanding of how to design our prompt. We plan to use a pre-built template and customize it with user preferences when a get request is made. To create this template, we will follow guidelines we found during our research, many of which were taken from articles that provide best practices for prompting generative AI tools [7]+[8]+[9].

Key conclusions for prompt guidelines:

- Clear and specific instructions: Ambiguity in the prompt can lead to responses that may not fulfill the user's expectations.
- Using explicit constraints: In some cases, it is beneficial to specify explicit constraints in your prompt, such as format, length, or scope.
- Leveraging System 1 and System 2 questions: System 1 questions need quick, intuitive answers, while System 2 questions require analytical problem-solving. Create prompts that cater to both types of questions to guide Chat effectively.
- Controlling output verbosity: You can control verbosity by explicitly requesting responses of a specific length or by asking the model to provide information.
- Iterative testing and refining: Refine prompts based on generated responses for better model behavior.
- Balancing user intent and model creativity: Ensure that the prompt addresses the user's needs while allowing the model to showcase its capabilities.
- Harnessing external resources and APIs: By using external resources and APIs, you can improve the performance of generative AI tools in tasks that require real-time or specialized data.
- Temperature and token control: Adjusting the temperature and token count of ChatGPT's responses can have a significant impact on the generated content. Lower temperature values result in more focused and deterministic outputs, while higher values lead to more diverse and creative responses.

- Adapting prompts for domain-specific applications: Tailoring prompts for specific domains or industries can help you obtain more accurate and relevant information from the chat.

### **sources of information:**

[1] “Global Payments: Airbnb, Expedia, Booking.com And The Travel Industry’s 2021 Prospects”, By Daniel Webber, Forbes magazine, Dec 2020.

<https://www.forbes.com/sites/danielwebber/2020/12/16/global-payments-airbnb-expedia-bookingcom-and-the-travel-industrys-2021-prospects/?sh=c4f52f154c09>

[2] “Numerous Parts of the Travel Industry are Rapidly Being Dominated by Monopolies—and Relief Does Not Seem at Hand”, by Arthur Frommer, 2015, Frommer’s magazine.

[https://www.frommers.com/blogs/arthur-frommer-online/blog\\_posts/numerous-parts-of-the-travel-industry-are-rapidly-being-dominated-by-monopolies-and-relief-does-not-seem-at-hand](https://www.frommers.com/blogs/arthur-frommer-online/blog_posts/numerous-parts-of-the-travel-industry-are-rapidly-being-dominated-by-monopolies-and-relief-does-not-seem-at-hand)

[3] Artificial Intelligence in the Tourism Industry: An Overview of Reviews, by Miguel-Ángel García-Madurga, and Ana-Julia Grilló-Méndez, July 2023.

<https://www.mdpi.com/2076-3387/13/8/172>

[4] “How Will Artificial Intelligence Impact The Travel Industry?” By Roger Sands, Nov 2023, Forbes magazine.

<https://www.forbes.com/sites/rogersands/2023/11/01/how-will-artificial-intelligence-impact-the-travel-industry/?sh=2adc3d0951d7>

[5] “General Principles of User Interface Design and Websites”, by Bhaskar N U, Prathap Naidu P, Ravi Chandra Babu S R & Govindarajulu P, International Journal of Software Engineering (IJSE), Volume (2) : Issue (3) : 2011

[https://d1wqtxts1xzle7.cloudfront.net/35211157/IJSE-54-libre.pdf?1413847032=&response-content-disposition=inline%3B+filename%3DGeneral\\_Principles\\_of\\_User\\_Interface\\_Des.pdf&Expires=1706955993&Signature=H~zF6BsfwjNe4lrO8dgTmP162yvsLWKtOp2PdtkWxw0~Vt5pujb7fR](https://d1wqtxts1xzle7.cloudfront.net/35211157/IJSE-54-libre.pdf?1413847032=&response-content-disposition=inline%3B+filename%3DGeneral_Principles_of_User_Interface_Des.pdf&Expires=1706955993&Signature=H~zF6BsfwjNe4lrO8dgTmP162yvsLWKtOp2PdtkWxw0~Vt5pujb7fR)

[6] Generative AI and ChatGPT: Applications, challenges, and AI-human collaboration, JOURNAL OF INFORMATION TECHNOLOGY CASE AND APPLICATION RESEARCH 2023, VOL. 25, NO. 3, 277–304.

<https://www.tandfonline.com/doi/epdf/10.1080/15228053.2023.2233814?needAccess=true>

[7] PROMPT ENGINEERING FOR CHATGPT, By ChatGPT4, Sabit Ekin (Texas University), 2023

<https://www.techrxiv.org/doi/full/10.36227/techrxiv.22683919.v2>

[8] “How to Prompt? Opportunities and Challenges of Zero- and Few-Shot Learning for Human-AI Interaction in Creative Applications of Generative Models”, By Hai Dang, Lukas Meck, Florian Lehmann, Sven Goller, Daniel Buschek, 2022.

<https://arxiv.org/pdf/2302.11382.pdf>

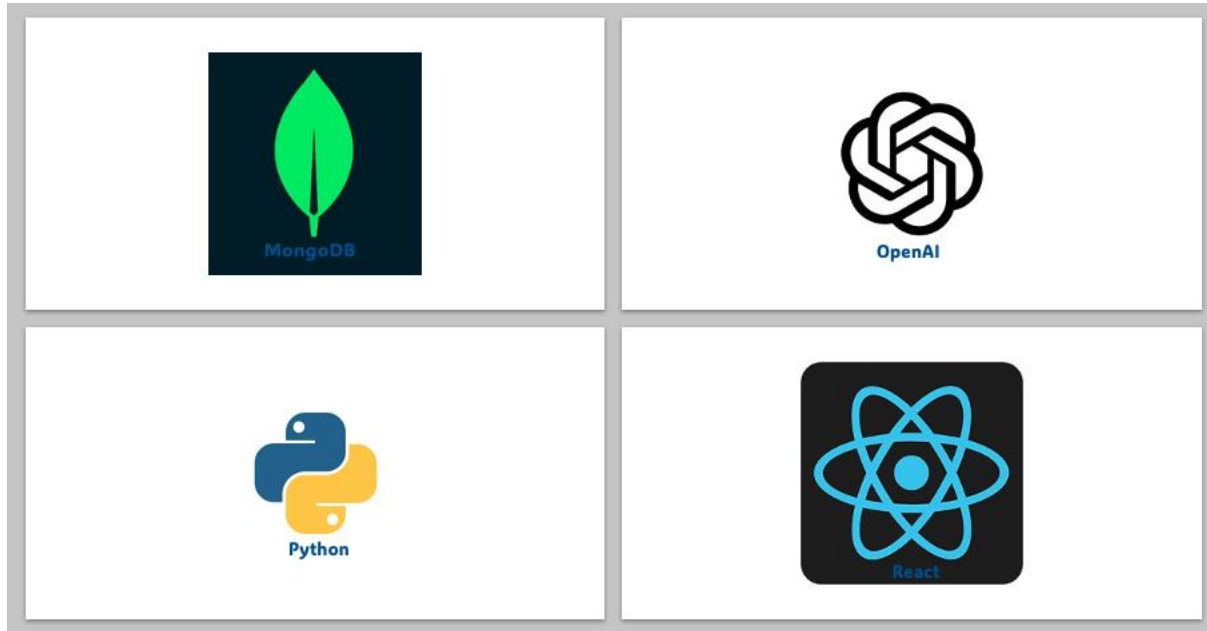
[9] “An Outline for an Interrogative/Prompt Library to help improve output quality from Generative-AI Datasets”, by Adam Svendsen, Bruce Garvey, May 2023.

[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=4495319](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4495319)

### 3. High-Level Design

#### a. System & Technological Alternatives:

The software and tools that we chose during the planning process:



#### 1) Frontend Alternatives

- Flutter -

A UI toolkit created by Google to build natively compiled mobile, web, and desktop applications from a single codebase. Known for its expressive and flexible design.

- Angular -

A widely-used web application framework developed by Google. Allows for building dynamic, single-page web applications.

- Ionic -

Utilizes web technologies (HTML, CSS, JavaScript) for cross-platform mobile applications. A good choice if you're looking to target both web and mobile platforms from a single codebase



## 2) Backend Alternatives

- Java -

Using Java with the Spring Boot framework base, allows the developer to expose endpoints via controllers in a straightforward way as well as the ability to migrate with third-party applications such as MongoDB.

- Express.js -

A lightweight and flexible framework for building web applications and APIs in JavaScript. It can be considered for its simplicity and speed.

- Django (Python) -

High-level Python web framework encouraging rapid development. Comes with built-in features like an admin panel and ORM.

- ASP.NET Core (C#) -

Free, open-source, cross-platform framework for building modern applications. Suitable for performance and scalability.

## 3) Database Alternatives

- PostgreSQL -

Advanced open-source relational database. Suitable for applications requiring complex queries, table joins, or transactions.

- Firebase Firestore -

NoSQL database from Google, offering real-time data synchronization. Ideal for applications needing real-time updates.

- SQLite -

C-language library for a small, fast, self-contained SQL database engine. Suitable for mobile applications requiring a lightweight database solution.

#### 4) Generative AI Alternatives

- Google Cloud Natural Language API -  
Enables sentiment analysis, entity recognition, and syntax analysis, among other language processing features.
- Microsoft Azure Text Analytics API -  
Offers natural language processing capabilities, including sentiment analysis, key phrase extraction, and language detection.
- Hugging Face Transformers API -  
Hosts various pre-trained transformer models, including GPT-2 and GPT-3, for natural language processing tasks.

### b. System Requirements

These system requirements provide a foundation for developing the "Voyage" project, ensuring that the system meets user expectations, performance standards, and legal compliance. Adjustments and additions may be necessary based on specific project considerations and stakeholder feedback during the development process.

#### 1) User Interface (UI) Requirements -

- Responsive Design -  
The UI must be responsive and accessible across various devices, ensuring a seamless user experience on both mobile and desktop platforms.
- Intuitive Navigation -  
Navigation within the application should be straightforward and intuitive, allowing users to easily access different features and sections.

## 2) Functional Requirements –

- Trip-Generation-

The system should generate trip plans based on user preferences, considering factors like travel dates, budget, and activity types.

- Content Publishing -

The system shall provide a platform for business customers to publish their content (restaurants, hotels, attractions) seamlessly into trip options.

- System - generative AI communication -

The system shall integrate with a generative AI that can propose trip plans based on user-specified preferences and prompt queries.

- Data verification -

The system shall perform a data verification of the AI-generated response as well as verify that the response data matches the user's requests.

## 3) Performance Requirements –

- Response Time -

The system should maintain a short response time, particularly for AI-generated trip plans and content retrieval.

- Scalability -

According to future identification of used habits, the system shall be able to scale to accommodate an increased number of users and business clients. For now, the system should build in a layer model that allows easy scalability in the future.

## 4) Reliability Requirements –

- The system shall use reliable data sources and conduct periodic data validation to ensure accuracy.

## 5) Integration Requirements –

- API Integration -

Integrate APIs seamlessly for services like AI trip generation and third-party validation tools.

- Database Integration -

Allow smooth communication between our system and the MongoDB for efficient data retrieval and storage.

## 6) Database Requirements –

- Data Integrity -

The database should maintain data integrity, ensuring accuracy and consistency of information.

- Backup and Recovery -

Implement regular backup procedures and a robust recovery mechanism to prevent data loss.

## 7) Usability Requirements –

- User Guidance -

Provide clear guidance through the UI, helping users to generate ideal queries and navigate the platform effortlessly.

- Configurability -

Allow users to configure various parameters within the itinerary generation process.

*\*\*Our project will start as a POC to meet the represented requirements. In the future, we aim to proceed with the development process and to answer the following ‘future requirements’ as well:*

**1) Authentication -**

- User Authentication -

Implement a secure user authentication system to ensure that user data is protected.

**2) Security Requirements –**

- Data Encryption -

Utilize encryption protocols to secure sensitive data, especially during user authentication and data transmission.

- Access Control -

Implement access control mechanisms to ensure that users and business clients have appropriate permissions.

**3) Maintenance and Support -**

- Logging and Monitoring -

Implement logging and monitoring tools to facilitate debugging and maintenance.

- Customer Support Integration -

Integrate a customer support mechanism to address user queries and issues promptly.

**4) Legal and Compliance Requirements –**

- Data Privacy Compliance -

Ensure compliance with data privacy regulations and implement features for user data management and consent.

**5) Performance Measurement –**

- Analytics Integration -

Integrate analytics tools to measure user engagement, track popular destinations, and gather insights for continuous improvement.

## c. Use Cases

### 1) User Creates a Personalized Trip Itinerary -

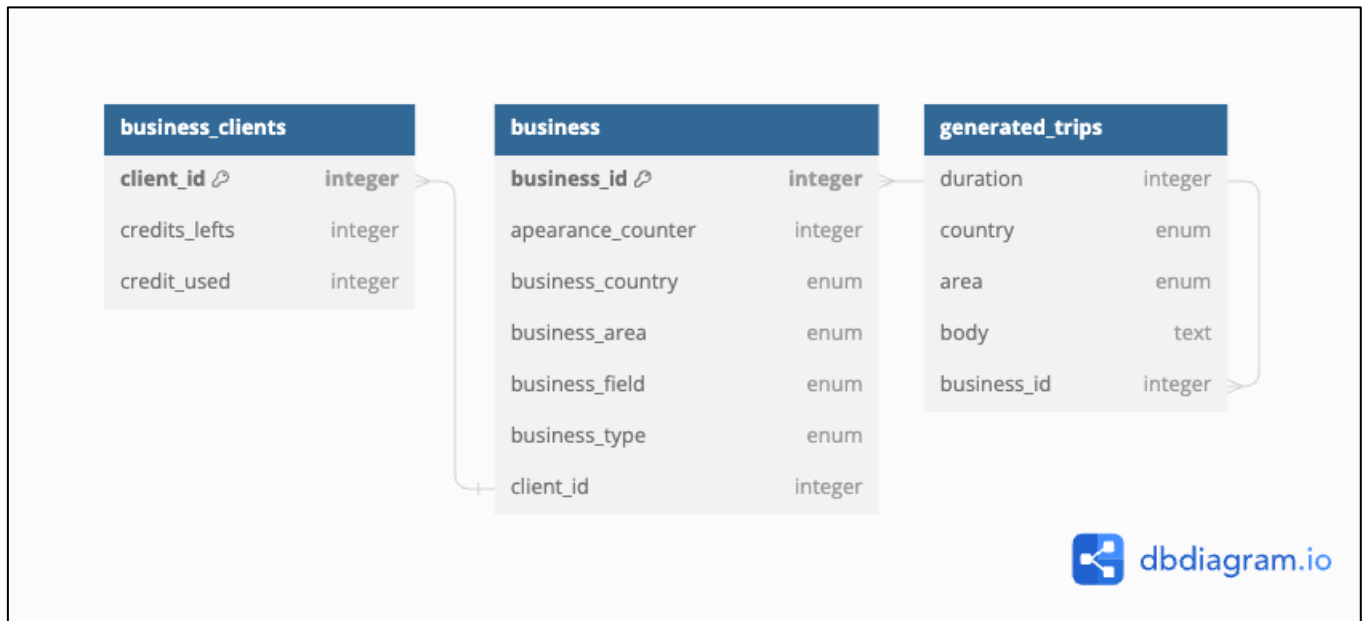
- Actor - User
- Precondition - The user entered the web page and reviewed available features.
- Main Flow -
  - a) The user selects the "Create Trip" option.
  - b) The system prompts the user with a series of guiding questions about preferences (travel dates, budget, interests, etc.).
  - c) The user interacts with the UI to configure various parameters for the desired trip itinerary and submit the query.
  - d) The system builds a prompt to send a query to the generative AI tool, the prompt includes the user preferences and the match business-clients content.
  - e) Generative AI processes the prompt and returns potential trip options.
  - f) The system performs the needed validations on the returned data.
  - g) The system displays the generated itinerary suggestions, including potential accommodations, activities, and transportation options.
- Alternative Flows -
  - a) Users can choose to explore pre-defined trip packages instead of creating a personalized itinerary – according to already saved trips.
  - b) The user encounters errors during input and receives clear error messages with guidance for correction.
- Postcondition - The user has a personalized trip itinerary saved and the system saves the generated trip for future use.

## 2) Business Client Uploads Content -

- Actor - Business Client
- Precondition - The business client has a registered account and appropriate access permissions.
- Main Flow -
  - a) The business client logs in and navigates to the "Content Management" section.
  - b) The system presents an interface for adding new business listings.
  - c) Business client enters details about their offering (restaurant, hotel, attraction), including name, description, location, images, and contact information.
  - d) The client categorizes the listing according to relevant activity types.
  - e) The client selects a payment plan to buy the credits he would like to.
  - f) The client reviews and submits the listing for publication.
  - g) The system validates the information and, upon approval, integrates the listing into the database.
- Alternative Flows –
  - a) Business clients can edit or delete existing listings.
  - b) Business clients can buy additional credits to existing listings.
  - c) Business clients can watch its listing and the generated trips it has been included.
- Postcondition - Business client's content is published and accessible for potential inclusion in user-generated itineraries.

## d. ERD

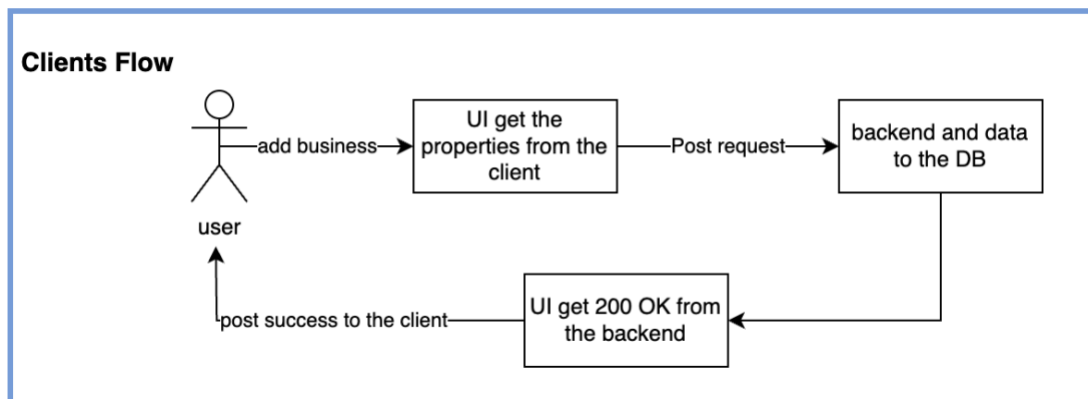
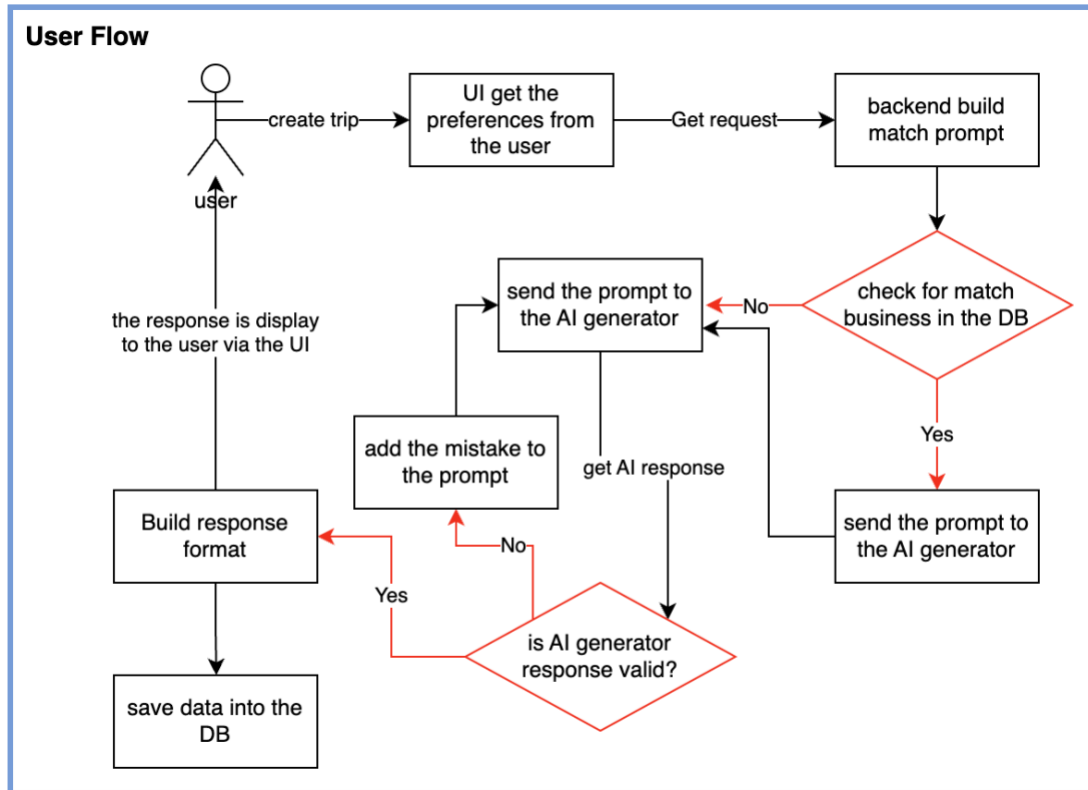
The ERD only represents DB entities and their relationships. In the next step, the LLD will include a class diagram for the entire system architecture.





## e. Flow Diagram

The two main use cases are illustrated in the attached flow diagrams, one for the user and one for the clients. The alternative flows, which required only small modifications, have been omitted to avoid data overloading.



## f. API Definition

### Add New Business Content

**POST** /api/v1/business\_app/add\_business

**Host:** www.VOYAGE

### Request Body

Name	Description	Type	Required	Example
business_name	Business name	String	Yes	"Agriturismo Gli Elfi"
business_type	Type of business		Yes	"Hotel"
business_phone	Business phone		Yes	"+393482264470"
business_email	Business email		Yes	"GliElfi@gmail.com"
business_country	Business country		Yes	Italy
business_contact_person	Name of the Business contact person		Yes	"Isable"
business_contact_person_phone	Phone of the business contact person		Yes	"+393482264470"
credits_bought	Number of credits to advertise the content		Yes	"100"
business_match_interest_points	list of match		Yes	"farm stay, winery"

	interest points, expected a String with commas (',') to separate without withspaces			
business_latitude	The exact location of the Business (can be found using GoogleMaps)		Yes	"45.7511748"
business_longitude			Yes	"7.3541632"
business_description	Short description of the business		No	"A family winery with an hotel"

### Response Body

```
{
  _id:<>
  business_client_id: <>
  business_name: <>
  business_type: <>
  business_phone: <>
  business_email: <>
  business_match_interest_points: <>
  business_country: <>
  appearance_counter: <>
  business_description: <>
  business_latitude: <>
  business_longitude: <>
}
```

StatusCode:

- 200 (for success - saved to the DB)/
- 500 (for failure due to an internal server error)/
- 400 (for failure due to invalid request error)

## Generate New Trip

**POST** /api/v1/users\_app/build\_trip

**Host:** www.VOYAGE

### Request Body:

Name	Description	type	Required	Example
budget	The requested trip budget	String	Yes	"Luxury"
season	The season of the trip		Yes	"Spring"
participants	The composition of the participants		Yes	"couple"
duration	The numbers of days of the requested trip		Yes	"4 days"
country-code	ISO 3166-1 code of the requested country		Yes	"IT"
interest-points	list of interest points, expected a String with commas (',') to separate without withspaces		Yes	"winery, day-trip"
accommodation_type	The type of the requested accommodatino		No	"farm-stay"

### Response Body

```
{
  "trip_itinerary": [
    {
      "day": <n>,
      "morning_activity": [
        {
          "content_description": <>,
          "content_latitude": <>,
          "content_longitude": <>,

```

```

        "content_name": <>,
        "content_type": <>
    },
],
"afternoon_activity": [
    {
        "content_description": <>,
        "content_latitude": <>,
        "content_longitude": <>,
        "content_name": <>,
        "content_type": <>
    }
],
"evening_activity": [
    {
        "content_description": <>,
        "content_latitude": <>,
        "content_longitude": <>,
        "content_name": <>,
        "content_type": <>
    }
],
"accommodation_recommendations": [
    {
        "accommodation_latitude": <>,
        "accommodation_longitude": <>,
        "accommodation_name": <>,
        "accommodation_type": <>
    }
],
"restaurants_recommendations": [
    {
        "restaurant_latitude": <>,
        "restaurant_longitude": <>,
        "restaurant_name": <>,
        "restaurant_type": <>
    }
]
}
]
}

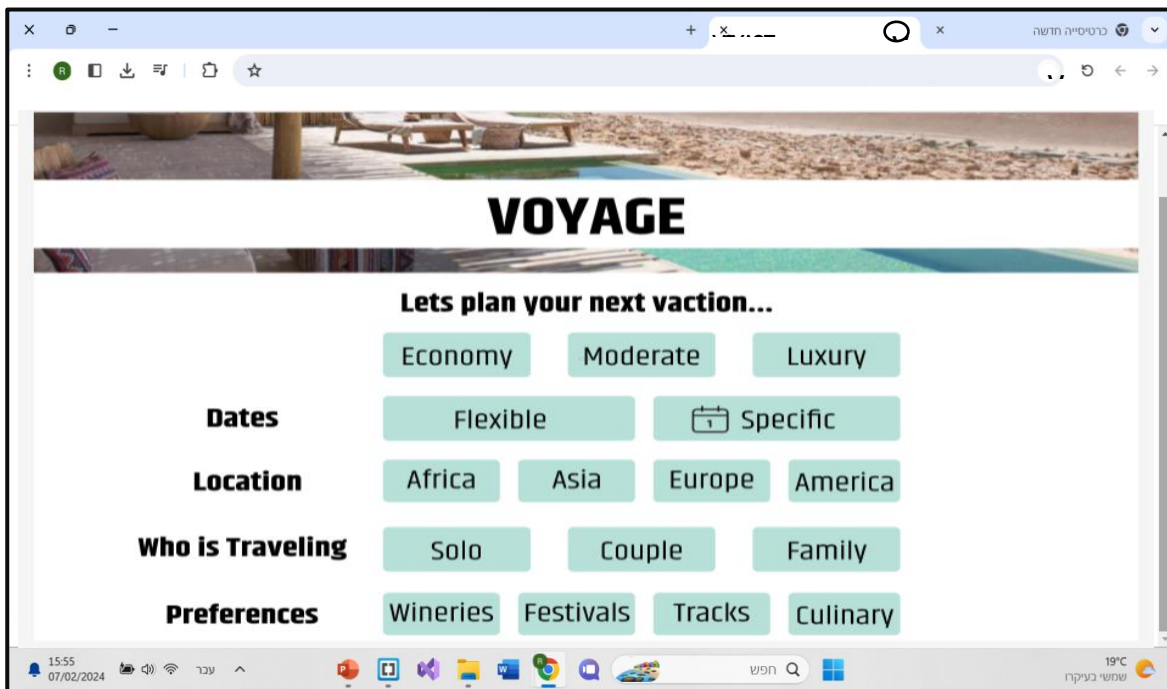
```

**StatusCode:**

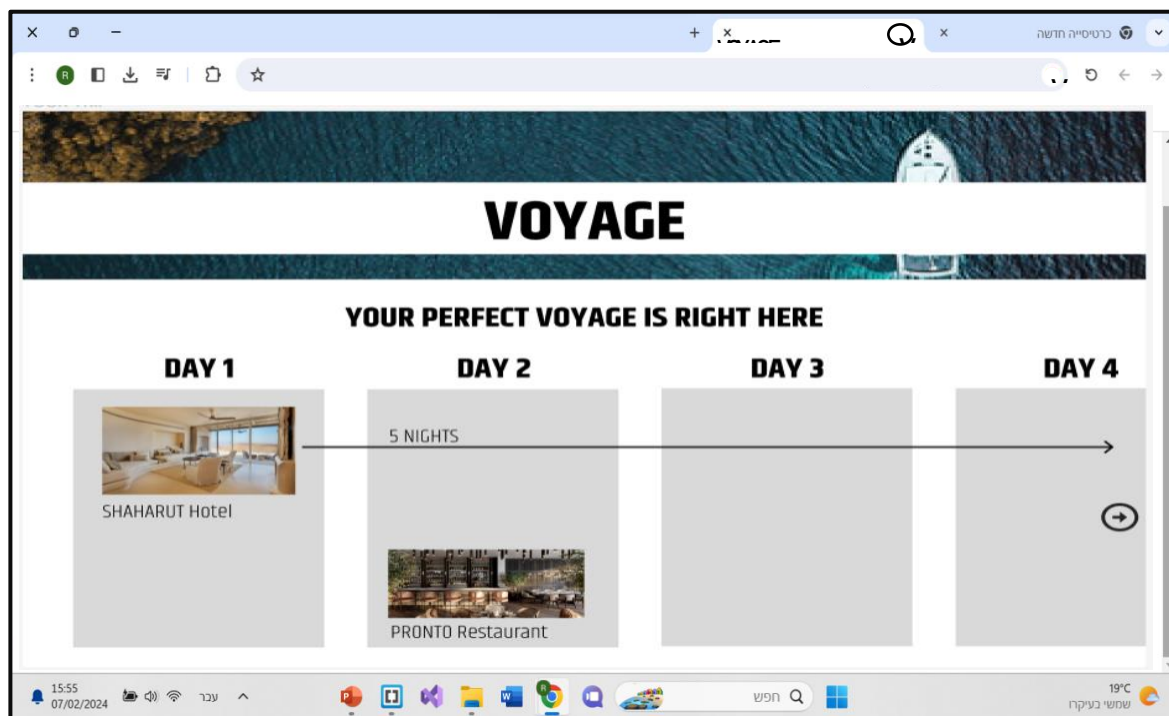
200 (for success - saved to the DB)/  
 500 (for failure due to an internal server error)/  
 400 (for failure due to invalid request error)

## g. Initial Screens design

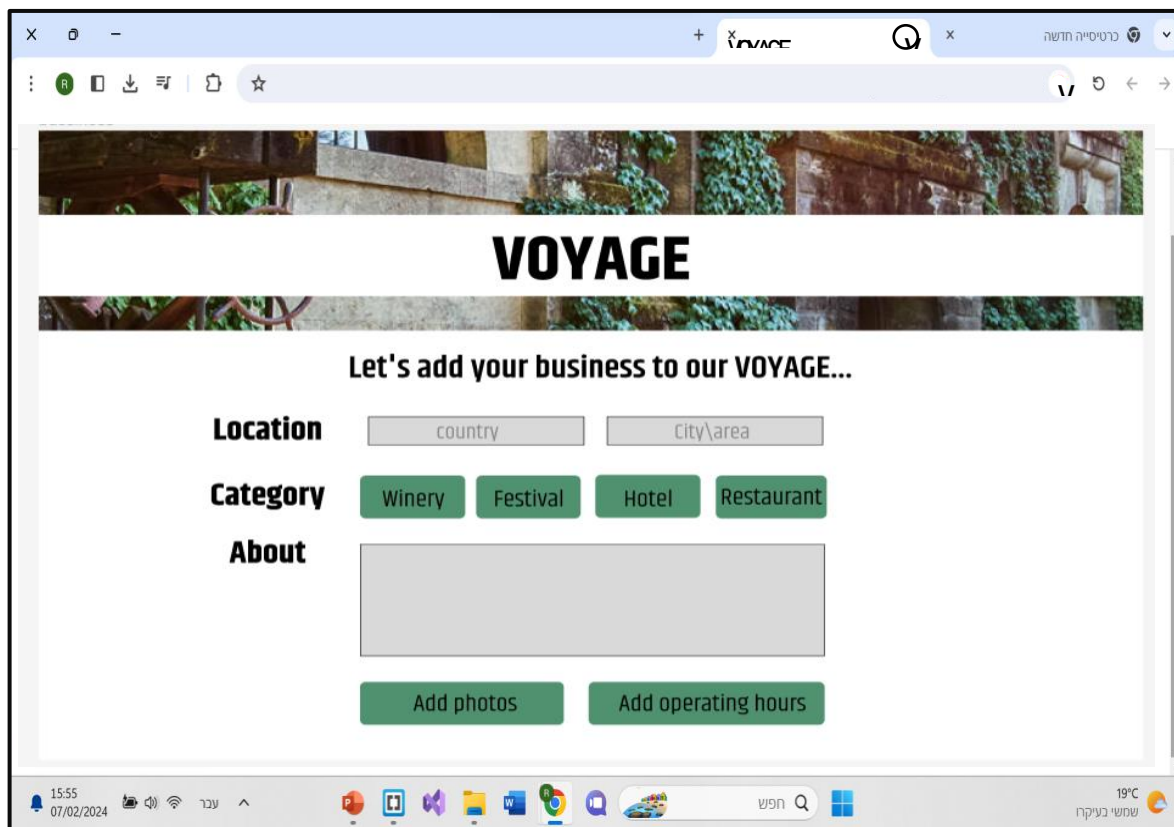
1) Users' web page – build trip screen:



2) Users' web page–generated trip results:



3) Client's web page – add new business:



The screenshot shows a web browser window displaying the VOYAGE website. The page has a header with the VOYAGE logo and a background image of a stone wall. Below the header, there is a section titled "Let's add your business to our VOYAGE...". This section contains three labels: "Location", "Category", and "About". Under "Location", there are two input fields: "country" and "City\area". Under "Category", there are four buttons: "Winery", "Festival", "Hotel", and "Restaurant". Under "About", there is a large text area. At the bottom of the form, there are two buttons: "Add photos" and "Add operating hours". The browser's address bar shows the URL "http://localhost:3000/". The Windows taskbar at the bottom shows the date and time as 15:55 on 07/02/2024, and the temperature as 19°C.

## h. Risk-Management

Type	The Risk	Description	Mitigation
Technical Risks	UI Design and User Experience	Users find the interface confusing or unappealing.	Conduct user testing and feedback sessions. Regularly update the interface based on user feedback.
	AI Output Accuracy	AI might provide inaccurate or irrelevant trip suggestions.	Implement robust testing and validation mechanisms for AI outputs. Regularly update and fine-tune the AI model.
	API Reliability	Unreliable API performance, leading to slow response times or system downtime.	Implement redundancy and failover mechanisms. Regularly monitor and optimize API performance.
	Data Security and Privacy	Potential breaches of user data, causing privacy concerns.	Implement encryption protocols, conduct regular security audits, and comply with data protection regulations.
Project Management Risks	Business Integration	Challenges in integrating business content seamlessly into trip options.	Work closely with business clients, provide comprehensive documentation for integration, and offer technical support. Conduct thorough testing before any new business client integration.
	Scalability	Handling a large number of users simultaneously without compromising response times.	Optimize code for scalability, employ cloud services for resource scaling, and conduct load testing.
	Database Management	Ensuring the database efficiently handles a growing amount of content and user data.	Optimize database queries, implement proper indexing, Regularly clean up and archive old data.
Market and Business Risks	AI Model Limits	AI may struggle with certain complex user queries or may not adapt well to evolving user preferences.	Regularly update and improve the AI model based on user interactions and feedback.
	Continuous Improvement	Continuously evolving the system based on user feedback and technological advancements.	Establish a feedback loop with users, regularly update the application, and stay informed about emerging technologies.



## **i. Engineering Challenges**

### **1) Generating the Ideal Prompt Query -**

- a) Defining and measuring the "best" trip plan is complex due to subjective user preferences and factors beyond the itinerary itself.
- b) Capturing and translating intricate user preferences like "hidden gem restaurants" or "unique cultural experiences" into clear prompts for the AI is challenging.
- c) The system might struggle to handle user preferences for activities or locations with limited data or outside the mainstream.

### **2) User Interface Simplicity and Speed -**

- a) Balancing simplicity with user control and customization is crucial. A simple interface might limit itinerary diversity and personalization.
- b) The system needs to be intuitive for users with varying levels of tech-savviness and comfort with AI-powered tools.
- c) Ensuring consistent and fast response times under high user traffic can be challenging, especially as the platform grows.

### **3) API Reliability and Data Quality -**

- a) Generative AI models are still under development and prone to biases, factual errors, and unexpected outputs. Addressing these issues in the context of trip planning is crucial.
- b) Relying on external data sources introduces the risk of inaccurate or outdated information. Ensuring data quality and addressing inconsistencies is essential.
- c) Integrating publisher content seamlessly can be challenging due to potential inconsistencies in data formats or structures.

### **4) Business Customer Integration -**

- a) Enticing businesses to invest in publishing their content requires a compelling value proposition.
- b) Managing data ownership, privacy, and control for both users and publishers requires robust data security and access controls.

## **j. Limitations and Mitigation Strategies**

For our Voyage project, aiming to deliver a personalized vacation planning platform utilizing generative AI, we acknowledge potential limitations that may impact progress and overall success. Understanding and addressing these limitations will be crucial for achieving project objectives efficiently. Here are some key limitations and corresponding mitigation strategies:

### **1) Lack of Knowledge in Generative AI and User Experience Design -**

#### **a) Description -**

Within our team, there are gaps in technical expertise and domain knowledge, specifically in generative AI algorithms and user experience design.

#### **b) Impact -**

- i. Inaccurate trip suggestions due to insufficient understanding of generative AI.
- ii. Suboptimal user interface leading to poor user engagement.
- iii. Challenges in aligning user preferences with AI-generated itineraries.

#### **c) Our Actions -**

- i. Learning and Skill Enhancement: We will invest in training sessions to upskill our team members in generative AI and user experience design principles.
- ii. Collaboration with Experts: We plan to consult with external experts in AI and UI/UX to provide guidance and mentorship.
- iii. Continuous Learning: We commit to encouraging continuous learning within our team through workshops, webinars, and knowledge-sharing sessions.

### **2) Technical Limitations in API Response Time -**

#### **a) Description -**

Potential technical constraints, such as server response time and API latency, may affect the overall user experience.

#### **b) Impact -**

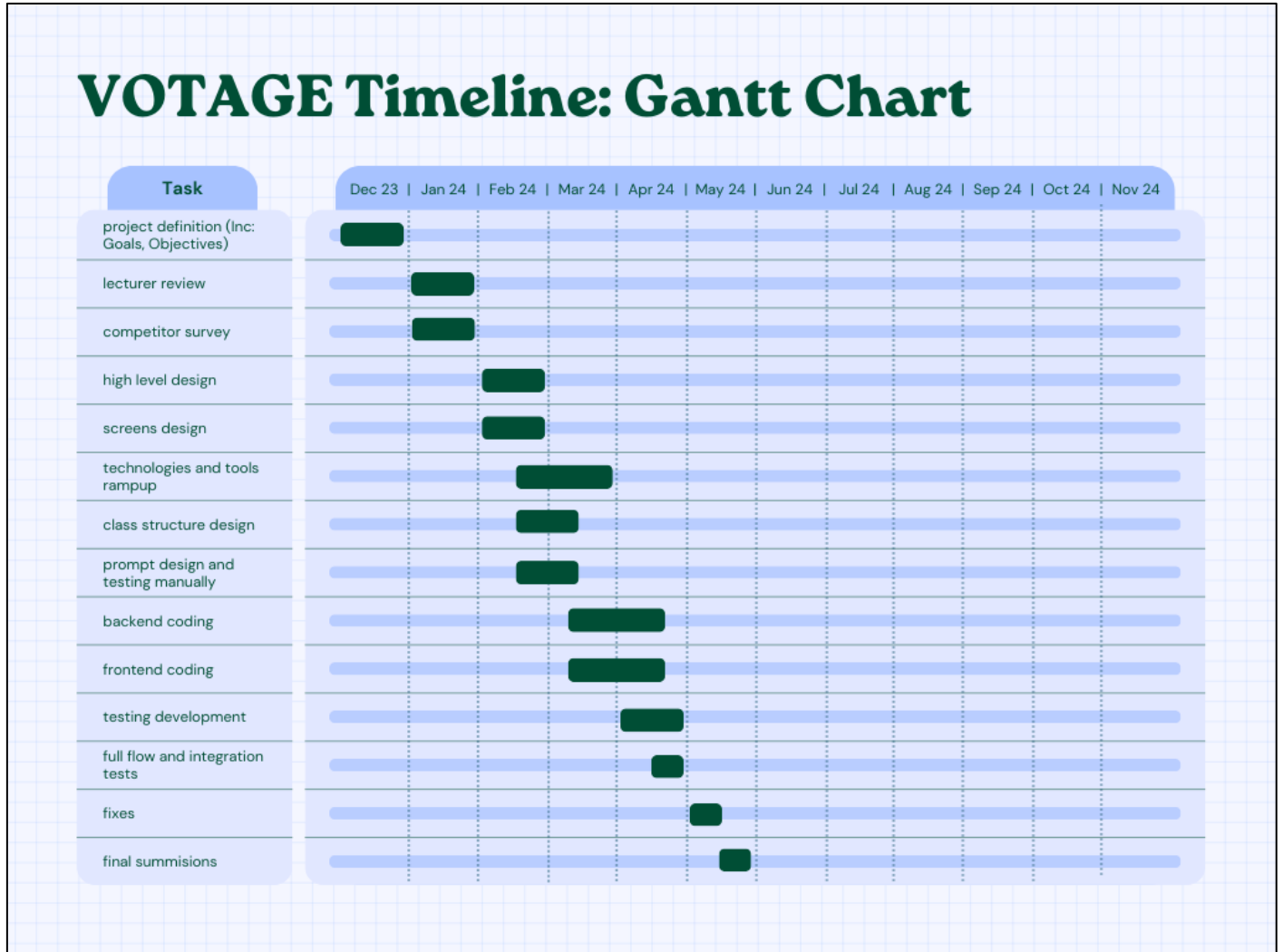
- i. User dissatisfaction due to slow response times.
- ii. Reduced platform reliability, impacting user trust.

- iii. Negative impact on the number of users interacting with the platform.

**c) Our Actions -**

- i. Performance Optimization: We will regularly optimize API performance to maintain response times below the established threshold.
- ii. Scalability Planning: Our plan involves developing a scalable infrastructure to handle increased user loads without compromising response times.
- iii. Monitoring and Maintenance: We are committed to implementing monitoring tools to detect and resolve performance issues proactively.

## k. Gant Charts



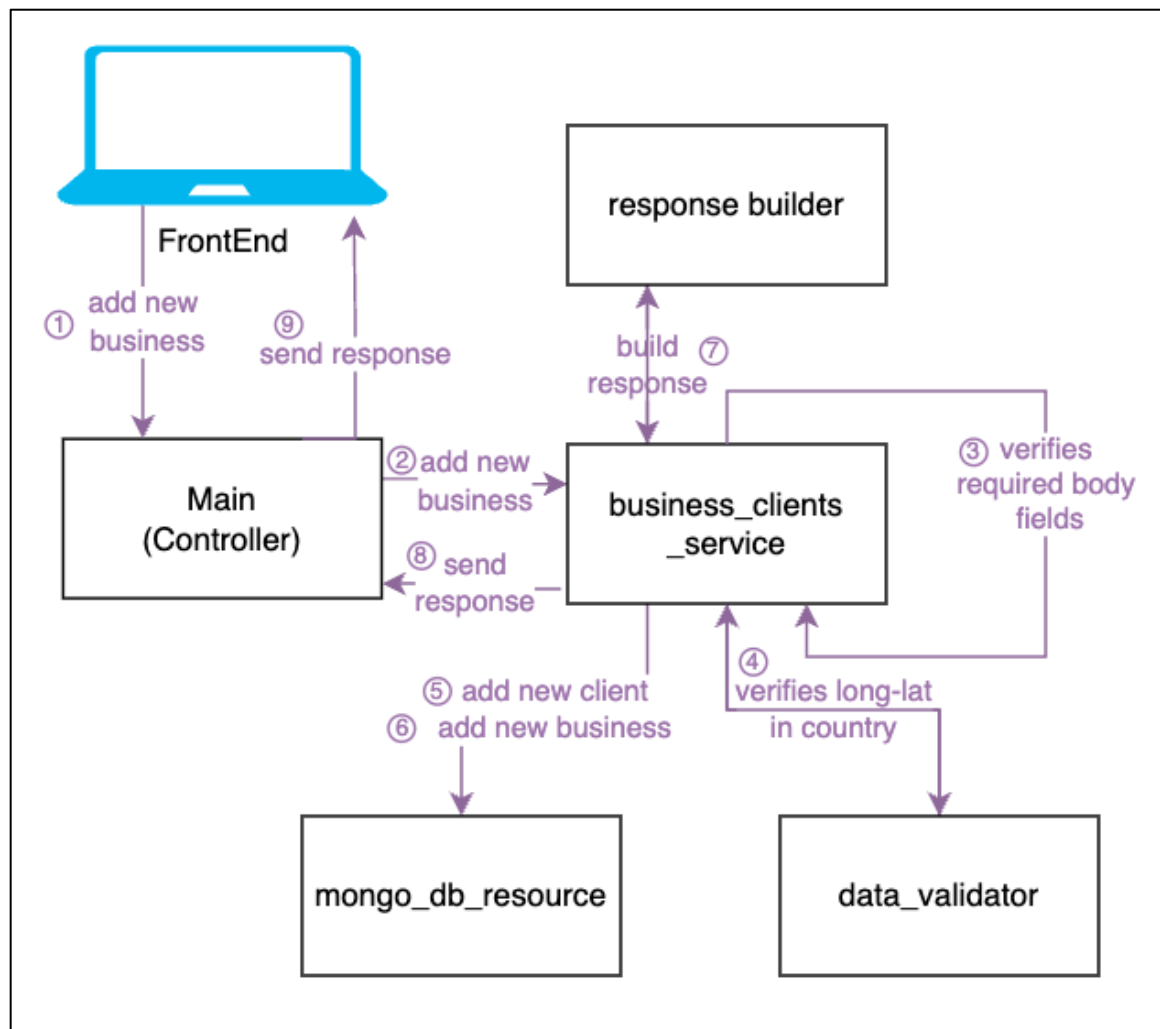
## 4. Products

### a. Git code link:

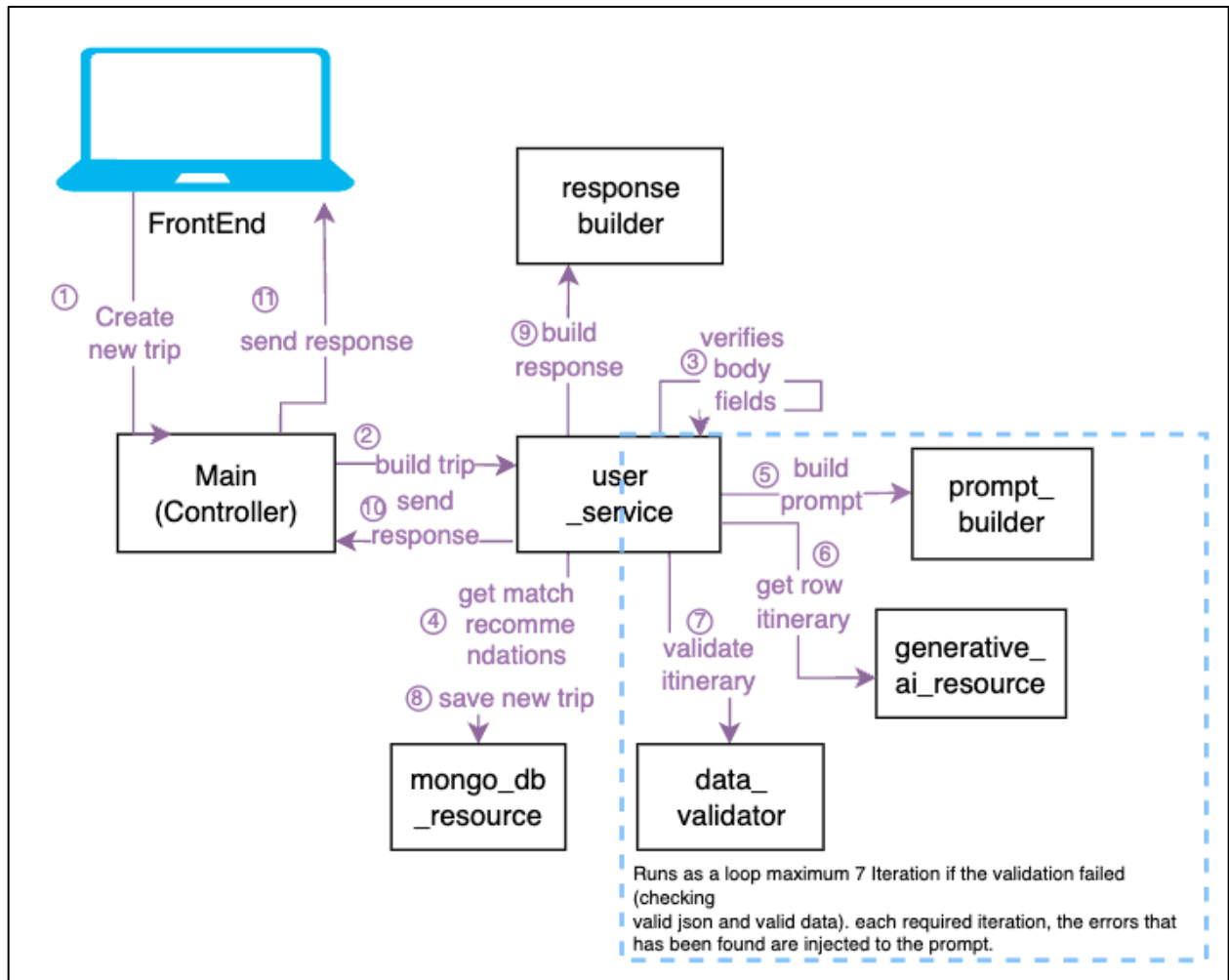
<https://github.com/voyageRN-project/voyage>

### b. Final System Design

#### 1) Business Controller Flow:

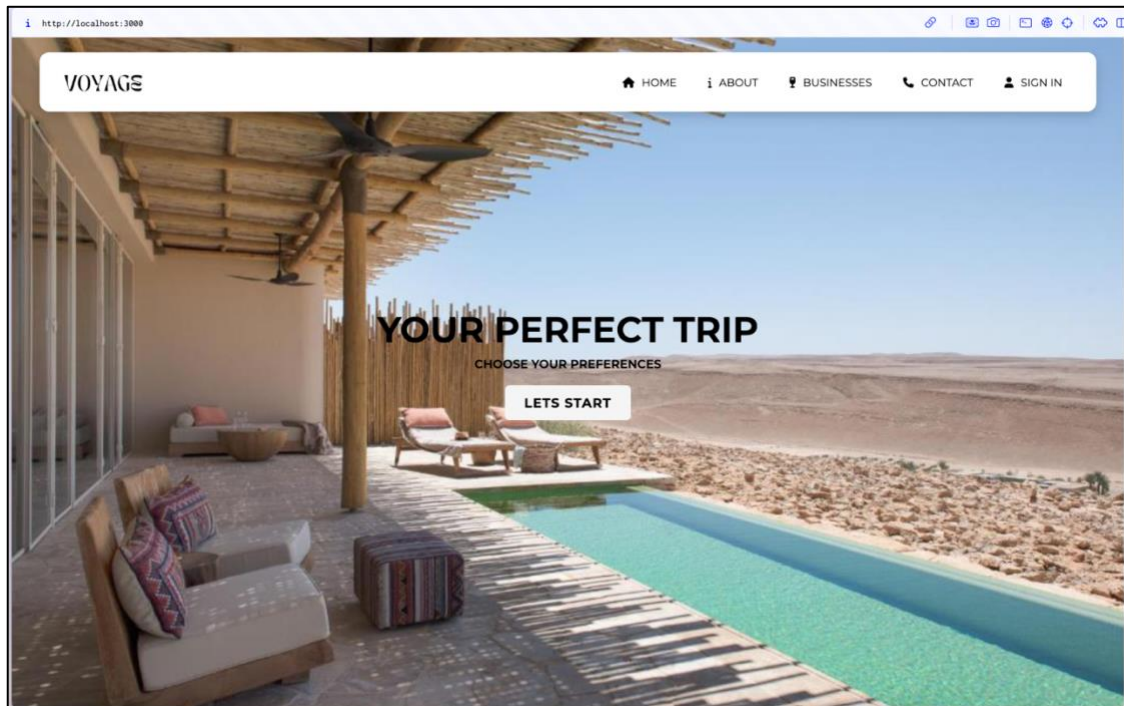


## 2) Users Controller Flow:

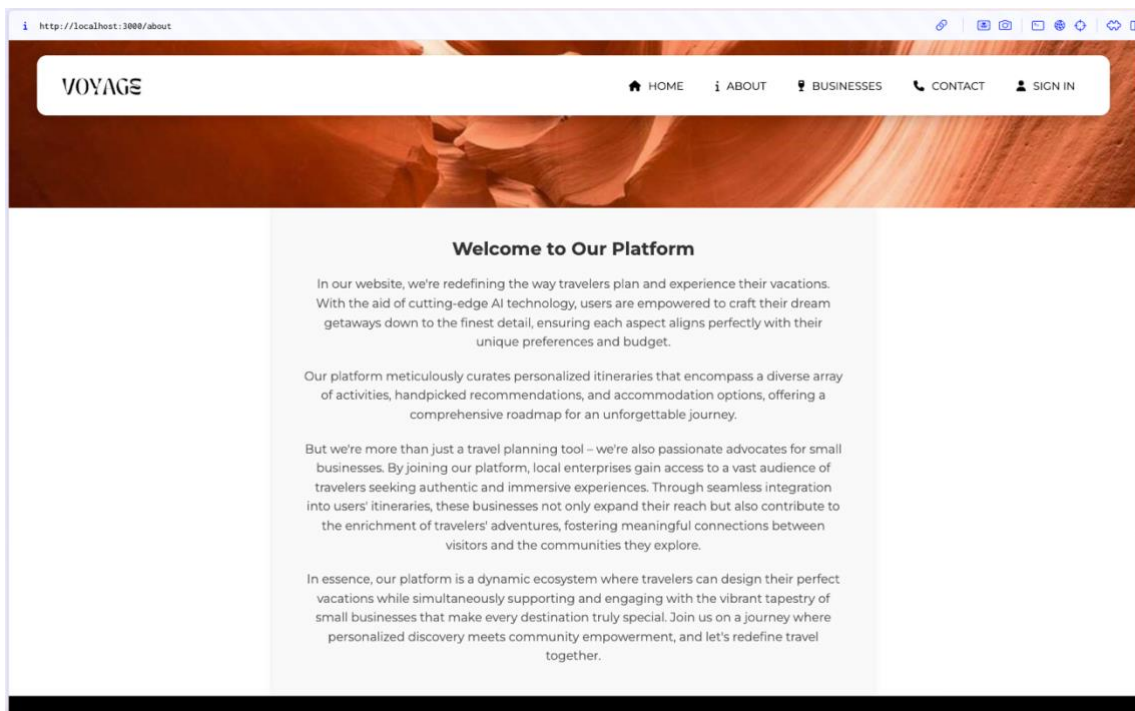


## c. Final Screens Design

### 1) Home page



### 2) About



### 3) Choose the trip preferences

The screenshot shows a web browser window with the URL `http://localhost:3000/tripBuild`. The page features a header with the 'VOYAGE' logo and navigation links: HOME, ABOUT, BUSINESSES, CONTACT, and SIGN IN. The main content area has a large banner image of a tropical beach with the text 'CHOOSE YOUR PREFERENCES' and 'BUILD YOUR PERFECT VOYAGE'. Below the banner is a form with the following fields:

- Budget:** A dropdown menu with 'Medium' selected.
- Season:** A dropdown menu with 'Spring' selected.
- Participants:** A dropdown menu with 'Couple' selected.
- Choose your destination:** A dropdown menu with 'Italy' selected.
- Duration of the trip (days):** A text input field with '3' entered.
- Choose accommodation type:** A dropdown menu with 'Farm house' selected.
- Choose your points of interest:** A row of checkboxes:
  - Art venues: ☐
  - Festivals: ☐
  - Hiking trails: ☒
  - Places to stay: ☐
  - Culinary: ☒

A 'Submit Preferences' button is located at the bottom of the form. The footer of the page is dark and contains the 'VOYAGE' logo, social media icons (Facebook, Instagram, Be, Twitter), and four columns of links: PROJECT (Afeka, Final Project, Neta & Roni), COMMUNITY (GitHub, Twitter, Issues), HELP (Support, Contact Us, Businesses Support), and OTHERS (Terms of Service, Privacy Policy, License).



#### 4) Result generated trip

The screenshot displays a web application interface for a travel agency named 'VOYAGE'. The header features the company logo and navigation links: HOME, ABOUT, BUSINESSES, CONTACT, and SIGN IN. The main banner image shows a scenic view of a coastline with the text 'YOUR VOYAGE' and 'This is your perfect trip... Discover and order'.

The itinerary is organized into three columns, each representing a day of the trip:

- Day 1:**
  - Morning Activity:** **Cinque Terre** - Explore the picturesque villages of Cinque Terre on a breathtaking coastal hike.
  - Afternoon Activity:** **Portovenere** - Visit the enchanting port town of Portovenere, known for its colorful houses and charming atmosphere.
  - Evening Activity:** **Ristorante Belforte** - Indulge in traditional Ligurian cuisine at the renowned Ristorante Belforte, offering panoramic views of the Gulf of Poets.
  - Restaurants Recommendations:** Trattoria dal Billy (Seafood).
  - Accommodation Recommendations:** Agriturismo Le Nacche (Farm House).
- Day 2:**
  - Morning Activity:** **Parco Nazionale delle Cinque Terre** - Embark on a scenic hike through the UNESCO-listed Cinque Terre National Park, offering stunning coastal views.
  - Afternoon Activity:** **Santuario di Nostra Signora di Soviore** - Visit the historic Santuario di Nostra Signora di Soviore, perched high on a hilltop with breathtaking views of the surrounding countryside.
  - Evening Activity:** **Trattoria da Aristide** - Savor authentic Italian flavors at the cozy Trattoria da Aristide, serving traditional dishes and local wines.
  - Restaurants Recommendations:** Ristorante Miky (Seafood).
  - Accommodation Recommendations:** Agriturismo Il Torchio (Farm House).
- Day 3:**
  - Morning Activity:** **Via dell'Amore** - Experience breathtaking coastal views on the Via dell'Amore, a romantic walking path connecting Riomaggiore and Manarola.
  - Afternoon Activity:** **Castello di Riomaggiore** - Discover the historic Castello di Riomaggiore, offering panoramic views of the town and coastline.
  - Evening Activity:** **Ristorante La Cantina** - Enjoy a delightful farewell dinner at Ristorante La Cantina, known for its traditional Ligurian dishes and warm ambiance.
  - Restaurants Recommendations:** Trattoria La Lanterna (Seafood).
  - Accommodation Recommendations:** Affittacamere Le Giare (Farm House).

The footer of the application includes the 'VOYAGE' logo and social media icons for Facebook, Instagram, and Twitter.

5) Add new business:

http://localhost:3000/businesses

**VOYAGE** [HOME](#) [ABOUT](#) [BUSINESSES](#) [CONTACT](#) [SIGN IN](#)

# BUSINESSES

## YOUR BUSINESS | THEIR VOYAGE

[LET'S START](#)

Business Name

Business Number

Business Category

Type

Business Description

Country

Latitude

Longitude

Email

Contact Person Name

Contact Person Number

Voyage Credit

**SEND**

## d. Software Tools

Name	Description
<b>Backend Tools</b>	
<b>Python</b>	High-level program language. Python allows the developer to perform a lot of complex actions in a short code. Very friendly integration with a lot of libraries and third-party tools. Used as the backend app language.
<b>Flask</b>	Flask is a lightweight Python web framework that balances ease of use with scalability for building web applications. Used to implement the backend's web application in our project.
<b>Google generative AI</b>	The Google generative AI Python library, allows you to interact with Google's powerful Gemini generative AI models for tasks like text generation and code completion, all through a convenient Python interface. Originally we planned to work with OpenAI Python's library because the Google tool hasn't been published to the community yet. During the development process and the library testing sessions, google published their tool for community usage and we decided to use it due to great testing results, the fact that it connects to the network (which means that it has live updates), and the fact that the community version is free to use.
<b>Pedantic</b>	Pedantic is a lightweight Python library that enforces stricter function-calling conventions by requiring keyword arguments and type-checking. We used it in our application to create the required JSON format from the generative AI.
<b>Pymongo and MongoDB</b>	MongoDB is a NoSQL database management system. It stores data in flexible JSON-like documents and is known for its scalability and ease of use with various programming languages, including Python (through Pymongo). Pymongo is a Python library that lets you conveniently interact with MongoDB databases.
<b>Geopy</b>	Geopy is a Python library that helps you work with geographical data, including address lookups and distance calculations.

	We used it in our backend to validate the location of the new businesses and the generate-ai-generated trip.
<b>Frontend Tools</b>	
<b>JS</b>	JavaScript, a programming language commonly used for web development, adds interactivity and dynamic behavior to web pages. It allows for manipulation of HTML and CSS, handles user interactions, and fetches data from servers. In our project, JavaScript was pivotal in creating a clear and user-friendly website, facilitating interactivity, dynamic content updates, and user input handling without page reloads.
<b>React</b>	React, a JavaScript library developed by Facebook, is widely used for building single-page applications and reusable UI components. Its component-based architecture and efficient rendering through virtual DOM make it a preferred choice for frontend development. In our project, React was leveraged to structure the user interface into reusable components, ensuring consistency and maintainability while enabling efficient management of UI elements.
<b>Figma</b>	Figma is a web-based design tool for interface design, prototyping, and collaboration. Used to create initial screens, serving as blueprints for the user interface design. From these screens, key components of the site were extracted, streamlining the development process. In our project, Figma was crucial for creating screens as blueprints for the UI design, streamlining development and ensuring alignment between design and implementation.
<b>Deployment Tools</b>	
<b>MongoDB Atlas</b>	A fully managed cloud database service for MongoDB, allowing easy creation, scaling, and management of your MongoDB databases without worrying about infrastructure.
<b>AWS App Runner</b>	A service by Amazon Web Services that simplifies building and deploying containerized applications and APIs. It streamlines the process, allowing you to focus on your application code rather than server management. Used to deploy our backend application
<b>Others</b>	

<b>MongoDB Compass</b>	A graphical user interface (GUI) for MongoDB, provides a user-friendly way to visualize, explore, and manage your MongoDB data through intuitive tools and visualizations.
------------------------	--

## e. Tests Coverage

### 1) Backend Test Coverage:

The unit test has been implemented within the backend itself. Due to time constraints and the fact that we are currently publishing a POC version, we decided to focus on testing the integration with third-party libraries and tools, such as generative AI (including the prompt builder and its use), the Mongo system, and the validators.

### 2) Frontend Test Coverage:

Testing the frontend has been currently performed as manual testing using Postman.

### 3) Backend and Frontend Integration Coverage:

During and after the development process, we manually checked and verified the integration between the frontend and backend applications.

## 5. Summary and Conclusion

### a. Compliance

#### 1) Measure Of Success Compliance:

Goal	Measure of success	result	Grade
Using guiding questions, the UI will assist the end user in generating the ideal prompt query, enabling the generative AI to propose the best trip	Number of configurable parameters: Count the number of user-adjustable variables within the itinerary generation process (travel dates, budget, activity types, etc.).	We allow the user a range of preferences to personalize his itinerary, such as budget, season, participants, destination country, trip duration, accommodation type, and point of interest. This range allows the user to	10

plan that matches the user's preferences		specify his preference in parallel to give the freedom to the generative AI to generate the trip. Additionally, we implemented the infrastructure to add options to ask for specific areas and cities as well as transformation types.	
	Well-designed prompt template: define a prompt template to inject the user's properties to get reasonable and accurate answers from the generative AI.	One main challenge was designing the prompt to bind the generative AI to return an expected JSON prompt to allow us to parse it. using the 'Pedantic' library we succeed in doing it.	10
	Average number of itinerary variations: Track the average number of alternative suggestions offered for each element (accommodation, transportation, landmarks) within a generated itinerary.	Due to the time limits we did not fully implement the variation tracking but we implement a match collection in the data base to allow simple extensions in the future.	7
Create a simple and clear user interface, while providing a short response time	Number of clicks to create an itinerary: Count the average number of clicks required to complete an itinerary from start to finish, indicating efficiency and simplicity.	The required number of clicks to complete an itinerary from start to finish is a Maximum of 13 in our POC version thanks to a clean and simple design use interface.	10
	Number of user interface elements: Minimize the number of buttons, menus, and options to avoid overwhelming users.	Using Figma to design our screen we succeed created a clean design that allows the user a simple understanding of our system.	10
	Average API response time under load: Test API performance under various user loads and aim for a consistent response time below 2 seconds.	The result average API response time is dependent on the itinerary length in days and in the model of the generative AI that we used. caused by our decision to use the free-of-charge	7

		version of the generative AI in our POC version, the response time is significantly higher than the target response time that we mentioned during our planning process. As will be described in detail in the 'future suggestion' section, we built the app in a layer abstract model so it will be easier to implement other generative AI tools in the future, which to our research, is supposed to accelerate significantly the response time.	
The API will provide a clear, reasonable, and reliable result that matches the user's request	Third-party data validation: Conduct periodic audits to ensure data accuracy and adherence to quality standards if relying on external data sources.	We implemented two levels of verification on the data and format which promised us to get a reliable itinerary that matched the user preference as well as to ensure that the business-inserted data is valid.	10
The application allows business-customers to publish their content in a way that integrates into the trip options	Each generated trip will include at least two recommendations according to the data in the DB.	Of course, depends on the data in the DB (match optional recommendations, enough credits to the optional once), our prompt is built to ask the generative AI to attach it list two managed recommendations in his response. According the tests coverage that we implemented, a successfully answered this goal.	10
	Build a payment method that relies on buying credits that the DB manages.	Our business program is based on a very common method that can be called "pay for what you get". each business that signs to our DB	10



		decides the number of credits he would like to buy - each credit allows recommendations in one itinerary. off course, we did not implement a third-party payment system in our POC version but we tested our system and ensured that credits were being counted by the appearances in the resulting itineraries.	
--	--	--	--

## 2) Implementation changes through the development process:

- a) Generative AI – during the development process, Google published their community version for the Python generative-ai client, and we decided to switch our implementation to a based on their library instead of the OpenAI tool for two main factors:
  - The Google generative AI is currently available to the community for free.
  - The model is connected to the network which means it gets live data.
- b) Verification tools – during the development process we discovered that our validation tool that performs the validation by the content name sometimes returns a false result, to deal with it, we decided to implement a second level validation by longitude-latitude using the ‘Geopy’ library.
- c) Frontend Screen – At a very early level of the development process we decided to improve our screen implementation as shown during the planning process. The improvements and the design were performed based on the academic article as described in the lecturer review.



## **b. Discussion**

### **1) Achievements Description:**

- First, we believe that our main achievement is the fact that in a short time and despite a lot of limitations (as described in the previous sections, but mainly our reserve duty and our decision not to spend money on this level of development) we have functional POC version of our planned project, that answered to our main goals.
- Secondly, we are proud to highlight our business program as a significant achievement. We believe that our application effectively meets the needs of small businesses in the travel industry and represents a valuable idea that will enhance our ability to promote and sell our concept.
- Our primary accomplishment in the frontend section is the implementation of a clear and simple user interface that adheres to the basic professional rules for a user-friendly application. Additionally, our frontend has been deployed and is available online for users in both web browser and mobile versions.
- In terms of our achievements in the backend section, we will focus on the two main challenges that we were very concerned about implementing correctly during the planning stage:
  - a) We have successfully created a clear prompt that frames the generative AI tool to generate the raw itinerary in an expected JSON format, allowing us to parse the data as needed.
  - b) We have found two ways to perform data validation that enable us to conduct two levels of verification of the data, ensuring that we will publish reliable data that matches the user preferences.

## 2) Drawing conclusion:

- The Gantt chart is an essential part of the High-Level Design (HLD) planning process. After its publication, we established a framework for each stage of product development. We think that creating a Gantt chart at an earlier stage, after mapping out the project requirements, could improve our process management.
- "During the coding stage, we had a clear division of responsibilities. One of us focused on the frontend, while the other worked on the backend. This division enabled us to concentrate on our respective strengths and work independently without being dependent on each other's development processes."
- During the deployment testing stage, we aimed to publish our POC version of VOYAGE by deploying it to the cloud. We only started learning the deployment process at the end of the coding and testing stages, but we found it to be a much more challenging task than we had anticipated. Additionally, it required us to make some small changes in the code. Therefore, we believe that learning the requirements and the process of deploying our project at an earlier stage could save us time and allow us to complete the deployment stage as we desired successfully.

## c. Future Suggestions

During our planning stage, we have identified some significant requirements that we believe we will be unable to meet within the time constraints of the POC version of VOYAGE. While many improvements can and should be made to the product, in this section, we will only highlight the major ones that we have chosen to focus on:

### 1) Accelerate the API's response time –

as mentioned in the previous sections, we aimed to generate our itinerary response in 2 seconds. As explained before, due to the bottleneck caused by the community-

free version of the Google generative AI, the response time is significantly higher than our expectation, therefore, one of the first required improvements. This acceleration can be achieved by two main improvements:

- We already implemented previous itinerary backup in a MongoDB collection to reduce the approaches to the generative AI tool – this way, before approaching the generative AI API, we can first approach the DB match collection and search for previously published itinerary that matches the user preferences.
- Secondly, as mentioned before, we implemented our backend in a layer's model which enables us to switch easily to other generative AI models. By replacing our model with an official payment version, we could significantly accelerate the generative AI response time and solve this bottleneck.

\*\*\* Due to the current situation the itinerary length is limited to a maximum of 5 days, this limitation will be removed after solving the generative-ai API bottleneck

- 2) Implementing the payment system – before we are ready to publish to the market our first POC version of VOAYGE, we must connect to a payment system to fully implement the business-clients side of the app. This should include integration with a third-party tool that manages the payment itself and all the security protocols that are required for such a system that holds such insensitive information.
- 3) Implementing a diversity tracing system – as mentioned in previous sections – we already implemented the itinerary collection in the DB as an infrastructure to implement this tracking system. This tracking system will allow us to analyze the response data in a better way and to improve our prompt if needed.
- 4) Generate Alternative routes – This could be an easy-to-implement goal that might improve our system significantly – instead of publishing only one version of the optional itinerary, we will publish two options and define the difference between them.

- 5) Publish our login system – our UI has already exposed a log-in window that hasn't been implemented. The log-in system allows the users and the clients to log in by their role – community user, or business-client.
  - The community users will be able to watch their previously generated trips and update them to their updated requirements and preferences.
  - The business-clients will be able to track their business recommendations by checking the remaining number of credits, increasing this number if needed, and track the full itineraries that include their business.

*\*\* It's important to note that these improvements do not represent our final product but rather mark the next stage goals.*

## **d. Conclusion**

**We want to end this stage of development with the following statement: "Thank you for having the courage to try."**

When we initially conceived the project, it felt like an almost unattainable goal, particularly given our limited knowledge and experience. We were on the verge of abandoning the idea due to the challenges that began to emerge.

We decided to give it a try and to see how it goes.

After a longer period of learning and research than we expected, we began our design and coding stage. Throughout the process, we gradually noticed that parts of the system were complete. However, there were many instances where one step forward led to a bug that set us two steps back. Despite this, we were successful in finding creative solutions and putting in the effort to improve the system whenever a bug was found.

Additionally, we would like to highlight the fact that we chose to embark on this 'voyage' as good friends, which can be challenging and daunting when working on a long-term project that is likely to lead to differences and disagreements. However, we feel that we succeeded in remarkably navigating through these challenges! The way we split the responsibilities allowed each of us to

focus and maximize our benefits, while also providing a critical outside perspective on the other part of the system, leading to significant improvements.

Finally, we take pride in our progress and accomplishments. While there is still a long road ahead before 'VOYAGE' becomes everything we envision, we believe that in a short time, despite significant challenges, we have successfully designed an entire system, researched best practices in the project field, learned and utilized new technologies and software tools, implemented and tested our proof of concept (POC) version of 'VOYAGE', and set our sights on the next steps. We are extremely proud of the results and excited to continue our 'VOYAGE'.