



**Department of Software Engineering**

**Capstone Project Phase A, 24-1-D-24**

**Supervisors**

Alexander Keselman

**Authors**

Dor Toizner 314763657 [toiznerd@gmail.com](mailto:toiznerd@gmail.com)

Raz Ben-Aharon 318317542 [raz228822@gmail.com](mailto:raz228822@gmail.com)

Table of Contents

[1 Abstract 3](#_Toc165975147)

[2 Introduction 3](#_Toc165975148)

[3 Background 5](#_Toc165975149)

[3.1 Hotels & Flights API 5](#_Toc165975150)

[3.2 ML (Neural Networks) 5](#_Toc165975151)

[3.3 JinaAI– Embeddings 5](#_Toc165975152)

[3.4 Recommendation Systems 5](#_Toc165975153)

[**3.4.1 Collaborative Filtering** 5](#_Toc165975154)

[**3.4.2 Content-Based Filtering** 6](#_Toc165975155)

[**3.4.3 Hybrid Recommendation Systems** 6](#_Toc165975156)

[3.5 MongoDB 7](#_Toc165975157)

[3.6 ReactJS 7](#_Toc165975158)

[3.7 Next.js 7](#_Toc165975159)

[3.8 Tailwind CSS 7](#_Toc165975160)

[4 Expected Achievements 8](#_Toc165975161)

[4.1 Achievements 8](#_Toc165975162)

[**4.1.1 Unique Software System** 8](#_Toc165975163)

[**4.1.2 Personalized Recommendation System** 8](#_Toc165975164)

[**4.1.3 Integration with External Sources** 8](#_Toc165975165)

[**4.1.4 Improved User Experience** 8](#_Toc165975166)

[4.2 Criteria for Success 8](#_Toc165975167)

[**4.2.1 Enhancement of User Experience** 8](#_Toc165975168)

[**4.2.2 Personalized Recommendation Quality** 8](#_Toc165975169)

[**4.2.3 Efficiency and Accuracy in Search** 9](#_Toc165975170)

[**4.2.4 Quality of External Integration** 9](#_Toc165975171)

[5 Engineering Process 10](#_Toc165975172)

[5.1 Process 10](#_Toc165975173)

[**5.1.1 Methodology** 10](#_Toc165975174)

[**5.1.2 Research** 11](#_Toc165975175)

[**5.1.3 Challenges** 12](#_Toc165975176)

[5.2 Product 12](#_Toc165975177)

[**5.2.1 Requirements** 13](#_Toc165975178)

[**5.2.2 Architecture** 13](#_Toc165975179)

[**5.2.3 Deployment** 15](#_Toc165975180)

[6 Evaluation and Verification Plan 16](#_Toc165975181)

[6.1 Testing 16](#_Toc165975182)

[6.2 Evaluation 17](#_Toc165975183)

[7 References 17](#_Toc165975184)

# 1 Abstract

As of today, the demand for flights is increasing every year due to the decrease in flight prices and the emergence of low-cost carriers that have opened up and lowered market prices. Consequently, more people prefer to travel abroad for vacations rather than within their own country. Additionally, digital advancement is attracting more individuals to book vacations independently through the internet, rather than through a human travel agent.

In our project we will build a web-application that includes ML & AI recommendation systems, that will propose to those individuals who book digitally a variety of options for booking vacations abroad in a way that consolidates both flight, accommodation and even car rental booking at the desired destination.

Our goal is to help the user to find a more suitable vacation for his own preferences in the easiest way.

# 2 Introduction

"Airline passenger traffic reached 94.1% of pre-pandemic levels in 2023, according to the International Air Transport Association (IATA). Traffic in 2023 climbed by +36.9% compared to 2022.", *The Moodie Davitt Report. [3]*

These days, there is a serious confusion in travel search or accommodations as there are a lot of websites dealing with these matters, and it is difficult to determine which ones are reliable and recommended. Each one focuses on a different aspect, whether it is a website for flight searches only or a site for hotel room reservations. The current situation is very confusing and complicates the efficient planning of a vacation.

On the other hand, today there are solutions to this problem that offer vacation packages including both flights and hotels. However, the existing solutions are not entirely efficient because they provide generic vacation packages to customers, rather than presenting personalized offers tailored to the individual customer's vacation preferences.

The existing solutions today are:

* Travel Agencies (Issta, Ofir Tours, Expedia etc.)
* Dedicated websites for booking flights (SkyScanner, Google Flights, etc.)
* Dedicated websites for booking guesthouses (Booking, Air B&B, Aguda, etc.)
* Car rental agencies (Budget, Shlomo Sixt, Hertz, etc.)
* Dedicated websites for booking vacations that include flights and guest houses (Holiday Finder, Ista, Deka 90, etc.)

Our goal is to transform vacation booking into a seamless and enjoyable task, minimizing complications for the general population. The primary focus lies in locating vacations according to the personal preferences of the client ordering the vacation, rather than treating them as just another generic customer seeking a holiday based on filtering. Additionally, serving as a digital hub for booking vacations worldwide, encompassing flights, accommodations, and possibly facilitating car rentals at the vacation destination.

Our stakeholders in the system:

1. End Users: Tourists, families, couples, business travelers, groups.
2. Tourism Service Providers: Guesthouses, hotels, car rental companies, travel agencies, tour companies.
3. Other Stakeholders: investors.

It will help each group as followed below:

1. End Users:

* Easy-to-use interface and convenient search.
* Wide variety of accommodation options and vacation types.
* Detailed and accurate information about each accommodation.
* Ability to easily compare prices and conditions.
* Secure and easy booking system.

1. Tourism Service Providers:

* Efficient platform to showcase their services to a wide audience.
* Convenient and easy-to-use booking management system.
* Ability to increase sales and reach more customers.
* Marketing and promotion tools for their businesses.

1. Investors:

* Investment potential in a growing and developing platform.
* Sustainable business model with high profitability.
* Experienced and professional team with experience in the tourism industry.

# 3 Background

In this section, we will provide an overview of the technologies and concepts we are employing in the development of Vacation Quest.

## **3.1 Hotels & Flights API**

Integrating with hotel and flight APIs gives us a live feed of information about available accommodations, flight schedules, and pricing. We partner with reputable sources, like major airlines and hotel chains, to make sure our users have access to the most recent and reliable travel info. This smooth integration lets users make informed decisions when they're planning their vacations.

## **3.2 ML (Neural Networks)**

Machine Learning like neural networks, is the backbone of our recommendation system. Neural networks enable us to analyze huge amounts of data according to user preferences, travel trends, and accommodation options. Through training models on historical booking data and user interactions, we can generate personalized recommendations matched to each user's preferences.

## **3.3 JinaAI– Embeddings**

We use JinaAI embeddings to enhance the natural language processing capabilities of our system. JinaAI embeddings enable us to understand and process user queries, and preferences in a better way. By using pre-trained language models, we can provide more accurate and relevant responses to user inquiries, and improve the overall user experience.

## **3.4 Recommendation Systems**

Our recommendation systems use advanced algorithms to analyze user behavior, preferences, and past interactions. By employing collaborative filtering, content-based filtering, and hybrid approaches, we can generate personalized vacation recommendations that align with each user's preferences and constraints. These recommendation systems serve as the main of our effort to streamline the vacation planning process and improve the user satisfaction.

### **3.4.1 Collaborative Filtering**

Collaborative filtering recommends items based on the preferences of similar users. It assumes that users who have liked similar items in the past will like similar items in the future. There are two approaches within collaborative filtering:

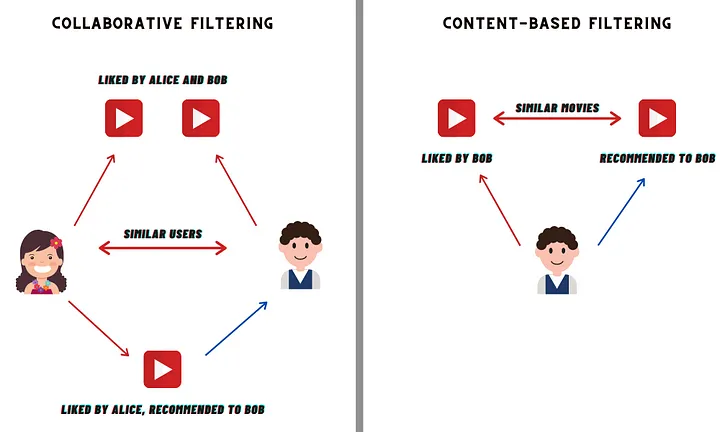
* **User-Based Collaborative Filtering** - Recommends items to a user that similar users have liked. For example, if User A and User B have both liked similar vacation destinations, User A might be recommended a destination that User B has visited and enjoyed.
* **Item-Based Collaborative Filtering** - Recommends items that are similar to items the user has liked in the past. In the context of Vacation Quest, this could mean recommending vacation destinations that are similar to ones the user has previously booked or shown interest in.



*Figure 1: User collaborative filtering by Amazon [1]*

### **3.4.2 Content-Based Filtering**

Content-based filtering recommends items similar to those the user has liked in the past. It analyzes the properties or features of items and recommends items with similar attributes. For example, if a user has booked beach vacations in the past, Vacation Quest might recommend other beach destinations based on attributes such as location and activities available.



*Figure 2: Collaborative filtering alongside Content-based filtering [2]*

### **3.4.3 Hybrid Recommendation Systems**

Hybrid recommendation systems combine collaborative filtering and content-based filtering approaches to provide more accurate and diverse recommendations. By leveraging the strengths of both methods, hybrid systems can overcome limitations and improve recommendation quality. For example, Vacation Quest might combine collaborative filtering with content-based filtering to provide personalized recommendations based on both user preferences and item attributes.

## **3.5 MongoDB**

We use MongoDB as the database for Vacation Quest, it is a flexible and scalable solution for storing and retrieving data. It’s document-oriented architecture enables us to store complex data structures, such as user profiles, booking histories, and accommodation details, in a schema-less format. This flexibility facilitates seamless data management and allows for easy integration with other components of our system.

## **3.6 ReactJS**

ReactJS is our front-end interface for Vacation Quest, it creates a responsive and interactive user experience. The component-based architecture of ReactJS allows us to modularize the user interface, making it easier to maintain and extend. By leveraging ReactJS's virtual DOM and state management capabilities, we can create a fast and intuitive web application that meets the needs of our users.

## **3.7 Next.js**

We use Next.js for performance and scalability of our React-based application. With features such as server-side rendering (SSR) and static site generation (SSG), Next.js enables us to optimize the loading times and SEO-friendliness of Vacation Quest. Additionally, Next.js provides built-in routing and API routes, streamlining the development process and improving code organization, it integrates with TypeScript and ReactJS which make it very convenient to use as a backend framework.

## **3.8 Tailwind CSS**

Tailwind CSS is used for styling Vacation Quest's user interface. Tailwind CSS enables us to rapidly customize the visual appearance of our application. Also Tailwind CSS is building only the necessary CSS styles in the project and it enhance the performance by using less memory for styling. The easy use of Tailwind CSS and the high performance of it makes it a very good choice for us.

By choosing these technologies and concepts, we aim to create a sophisticated and user-friendly vacation discovery system that empowers users to find their ideal holiday destinations with ease and confidence.

# 4 Expected Achievements

Our product aims to provide a comfortable and organized user experience for basic and advanced users.

## **4.1 Achievements**

### **4.1.1 Unique Software System**

We will develop and deploy an advanced software system integrating state-of-the-art Machine Learning algorithms with a sophisticated front-end platform based on ReactJS and Next.js, alongside modular and clean design using Tailwind CSS. The system will provide users with a user-friendly interface allowing them to search, find, and book vacations easily and conveniently.

### **4.1.2 Personalized Recommendation System**

We will implement a recommendation system utilizing advanced Machine Learning algorithms to recommend vacations tailored to users’ personal profiles and preferences.

### **4.1.3 Integration with External Sources**

We will utilize APIs from leading tourism companies such as airlines and hotel chains to provide users with real-time and reliable information regarding flights, accommodations, and additional travel services.

### **4.1.4 Improved User Experience**

We will design a comfortable and easy-to-use user interface providing a personalized and impressive user experience, including easy navigation and advanced search options.

## **4.2 Criteria for Success**

### **4.2.1 Enhancement of User Experience**

We will assess improvements in user experience metrics, including response time, ease of navigation, and user’s deals suggestions satisfaction.

### **4.2.2 Personalized Recommendation Quality**

We will conduct tests and evaluations to ensure that the personalized recommendations provided by the system align optimally with users' profiles.

### **4.2.3 Efficiency and Accuracy in Search**

We will evaluate the accuracy and efficiency of the search and filtering system and make adjustments to improve results.

### **4.2.4 Quality of External Integration**

We will verify that the information provided to users from external sources is updated and reliable.

# 5 Engineering Process

In the development journey of Vacation Quest, we encountered various challenges and anticipate encountering more as we progress. This section outlines the methodologies and techniques we employed during the engineering process of our project.

## **5.1 Process**

The engineering process for Vacation Quest started with a comprehensive exploration of the travel technology realm. This involved delving into existing vacation booking platforms, emerging trends in AI and ML-powered recommendations, and advancements in web development technologies. Given the dynamic nature of the travel industry, we had to stick to the latest innovations and market trends.

Simultaneously, we began on formulating the concept and use cases of Vacation Quest. This required us to brainstorm ideas, assess market needs, and identify potential issues faced by travelers. Developing a prototype played a crucial role in this phase, allowing us to experiment with different features and functionalities, validate our assumptions, and refine our vision.

Throughout this process, we encountered a learning curve as we familiarized ourselves with various technologies essential for building Vacation Quest. This included mastering frameworks like ReactJS and Next.js for front-end development, as well as exploring APIs for integrating with external services. Additionally, gaining insights into the complications of recommendation systems, neural networks, and natural language processing (NLP) was essential for implementing personalized vacation recommendations.

### **5.1.1 Methodology**

Our approach to the engineering process of Vacation Quest was guided by an iterative and hands-on methodology. We chose to build a simple prototype as an extension of fundamental project components. This allowed us to rapidly iterate over ideas, experiment with different features, and gain practical experience with the technologies involved.

By implementing core functionalities incrementally, we were able to validate our design decisions, identify potential issues early on, and correct them as needed. This iterative approach fostered agility and flexibility, enabling us to adapt to evolving requirements.

Moreover, our methodology emphasized continuous learning and skill acquisition. We embraced the opportunity to expand our knowledge base, whether it was mastering new programming languages and frameworks or deepening our understanding of machine learning algorithms and recommendation systems.

Overall, our process methodology was characterized by a balance of exploration and experimentation.

### **5.1.2 Research**

The research phase of our project was integral to understanding the landscape of vacation booking platforms, the existing technologies in the travel industry, and user preferences. This phase encompassed comprehensive exploration and analysis aimed at informing the design and development of the platform.

**Market Analysis**

We conducted a thorough analysis of existing vacation booking platforms, studying their features, user interfaces, and business models. This involved reviewing prominent players in the market, such as Expedia, Booking.com, and Airbnb, to identify trends, strengths, and areas for improvement.

**User Surveys and Interviews**

To gain insights into user preferences and pain points, we conducted surveys and interviews with potential users. This qualitative research helped us understand the needs, expectations, and behaviors of travelers, guiding the design of our application features and functionalities.

**Competitive Analysis**

We analyzed competitors to benchmark Vacation Quest against existing solutions. The analysis involved evaluating competitors' strengths and weaknesses, analyzing their market positioning, and identifying gaps that Vacation Quest could address.

**Prototyping and Validation**

Throughout the research phase, we developed prototypes to validate our ideas. These prototypes allowed us to test different design concepts, and iterate on the platform's features based on user input.

### **5.1.3 Challenges**

Throughout the development of our project, we encountered several challenges in different aspects of the project, including technical complexities, resource constraints, and unforeseen obstacles. Despite these, we remained committed to overcoming them and delivering a high-quality product.

At first, we weren’t sure if Next.JS is the best framework for working with ML models since we were used to implement them in python. later we changed direction of our project to be a middleman in searching a vacation by decreasing the financial transactions from our back and redirecting the client to the source we found for the price he wants to pay.

We also expect to encounter even more challenges in phase 2 of the project, when we will fully implement the project and begin to design the front-end.

## **5.2 Product**

In today's fast-paced world, planning vacations can be overwhelming. From choosing the destination to booking flights, accommodations, and other amenities, travelers often find themselves full of choices. Our project aims to streamline this process by developing a comprehensive web application that leverages the power of machine learning and artificial intelligence recommendation systems.

Our web application will employ state-of-the-art AI recommendation algorithms, powered by JinaAI, to suggest personalized vacation options to users based on their preferences and past booking history. These recommendation systems will analyze user data, including travel habits, budget constraints, preferred destinations, and activities of interest, to generate tailored suggestions. By adopting the potential of AI, we aim to provide users with a selection of vacation packages that align with their unique preferences and requirements.

To support the recommendation systems and store user data securely, we will use MongoDB. MongoDB offers flexibility, scalability, and robust document storage capabilities, making it an ideal choice for managing the diverse data sets associated with vacation bookings. By leveraging MongoDB's schema-less architecture and indexing capability, we can adapt to evolving user preferences and efficiently store and retrieve information relevant to each user's profile and booking history.

In planning the user experience of our web application, we prioritize simplicity and intuitiveness. We have chosen ReactJS, a popular JavaScript library for building user interfaces, to create a responsive and interactive front-end experience. The UX design will feature a clean interface, with intuitive navigation. Our key features of the UX design will be user-friendly interface, personalized recommendations, interactive booking process and real-time updates.

### **5.2.1 Requirements**

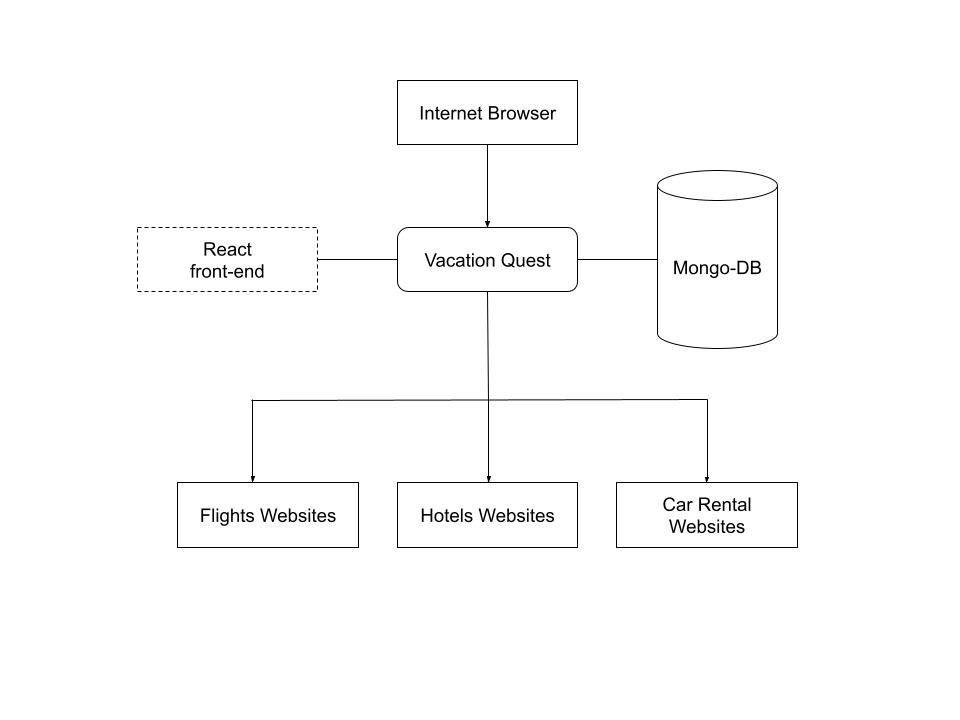
**Functional:**

* The system will allow to create a new user.
* The system will allow to book a new order.
* The system will search for deals for accommodation using external websites.
* The system will search for deals for flights using external websites.
* The system will search for deals for car rentals using external websites.
* The system will collect personal data for each user.
* The system will allow to contact by email.
* The system will allow to make changes on profile.

**Non-functional:**

* The system will collect data about the users’ preferences.
* The system will store booking history.
* The system will be used anywhere with internet access.
* The system will have a user-friendly interface.
* The system will be based on NextJs framework.
* The system will be connected to Mongo database.
* The system will be based on JinaAI embeddings for recommendation system.

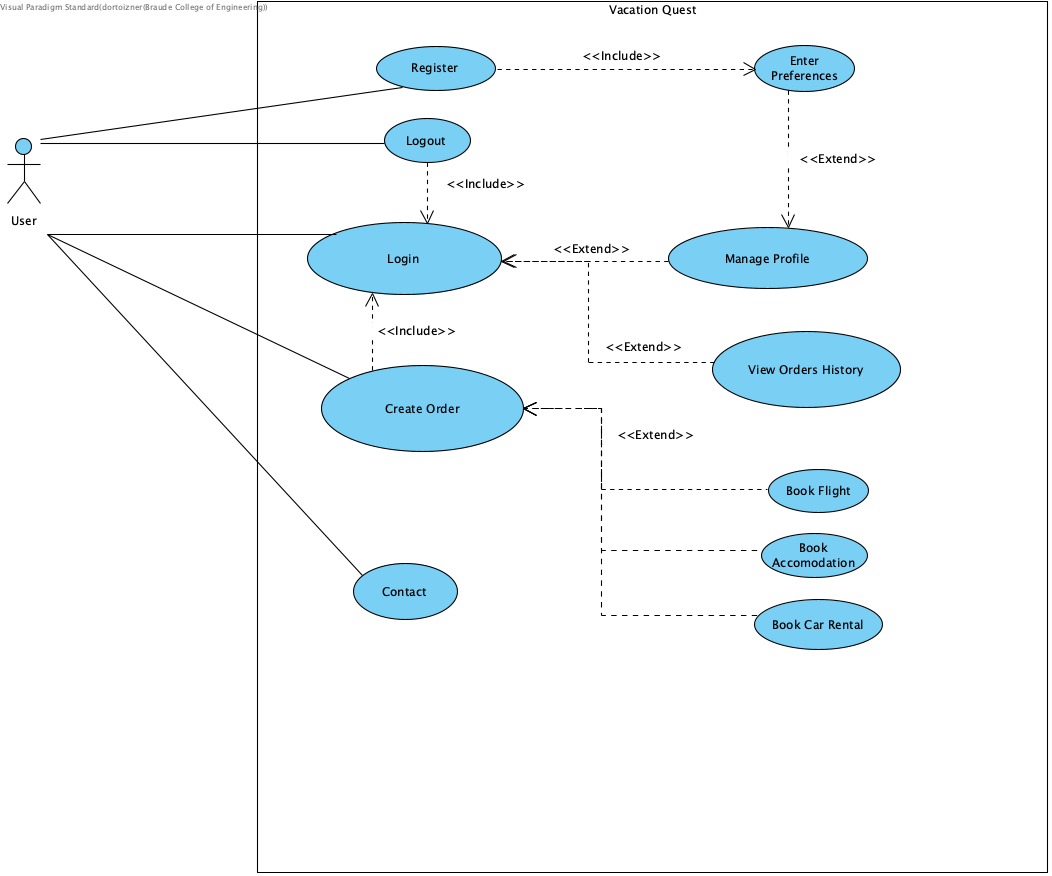
### **5.2.2 Architecture**



*Figure 3: An overview of the project architecture as a flowchart*

**Use case diagram**

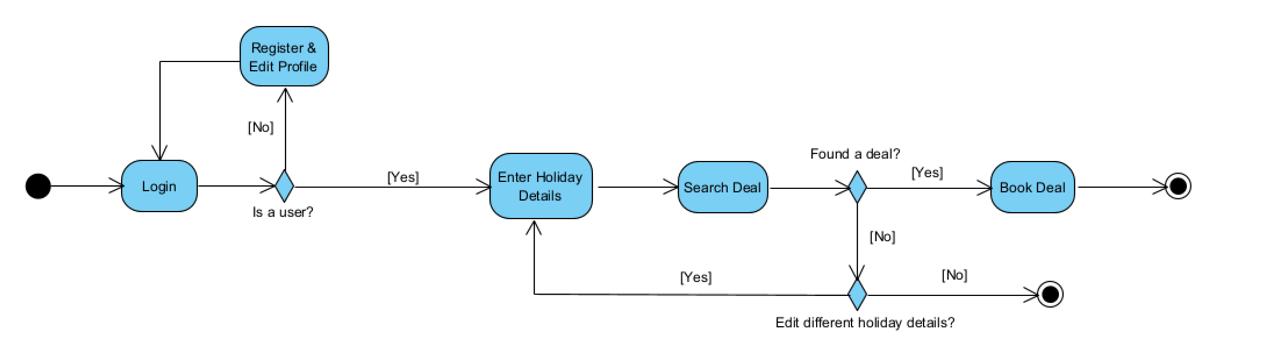
This diagram represents the main functionality a user can do.



*Figure 4: Vacation Quest Use-Case Diagram*

**Activity Diagram**

This diagram represents the user activity for booking a vacation.

****

### **5.2.3 Deployment**

# To deploy the project follow the instructions:

1. Open Terminal (Better in vs code)
2. Run “clone <https://github.com/ToiznerD/vacation-quest>”
3. Run command “cd vacation-quest”
4. Run command “npm install” (Make sure npm is installed in your computer)
5. Run “npm run build”
6. Run “npm run start”

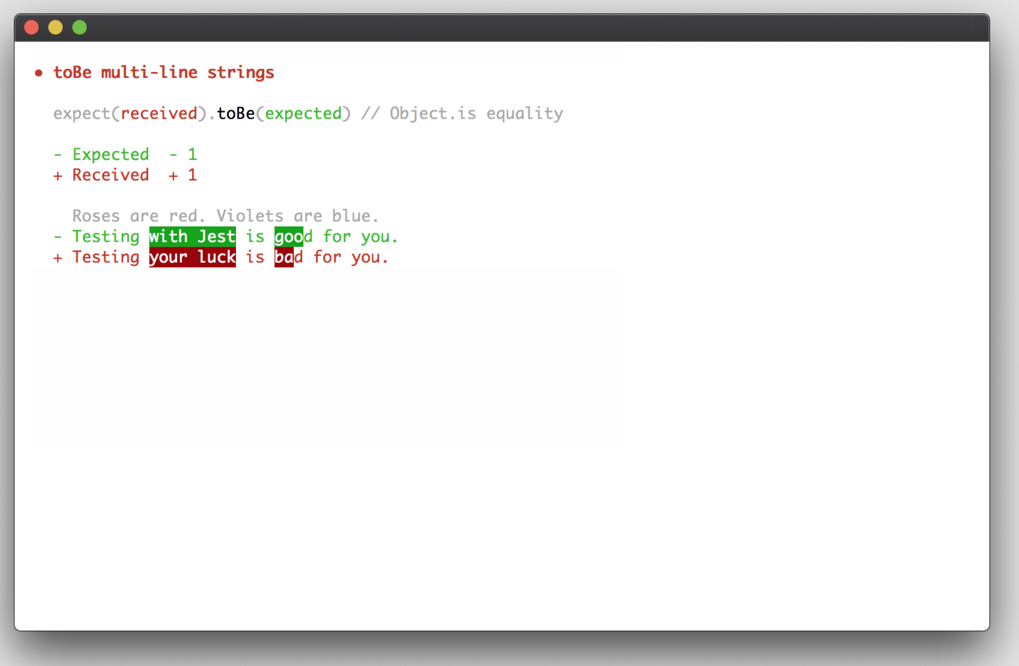
# 6 Evaluation and Verification Plan

## **6.1 Testing**

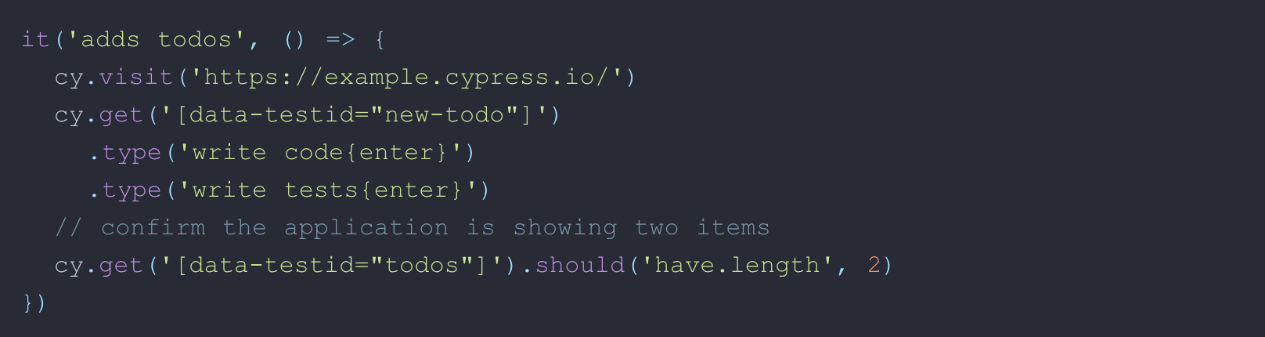
Our testing strategy encompasses both front-end and back-end components, ensuring the reliability and functionality of our vacation booking web application. For front-end testing, we will use Cypress, an end-to-end testing framework, to simulate user interactions and validate front-end elements. Cypress allows us to test the application from the user's perspective, ensuring smooth user experiences across various scenarios.

On the backend, we use Jest, a testing framework, for unit and integration testing of backend logic, APIs, and services. With Jest, we can test individual components and their interactions, ensuring the reliability at the backend level.

By combining Cypress and Jest, we create a comprehensive testing suite that covers both front-end and back-end aspects of our application, guaranteeing high quality and performance throughout.



*Figure 5: Example of the Jest CLI tool.*



*Figure 6: E2E example of the Cypress tool.*

## **6.2 Evaluation**

To ensure that our project meets the needs and expectations of our users, we will conduct surveys, questionnaires, and feedback forms to gather valuable insights from our users. Through this feedback, we aim to get user feedback on various aspects of the application, including usability, functionality, user experience, and overall satisfaction.

By asking targeted questions about their experiences, preferences, and pain points, we can gain a deeper understanding of what users value most in a vacation booking platform and identify areas for improvement. Additionally, integrating feedback forms directly into the application will allow users to provide real-time feedback and report any issues or suggestions for enhancement as they navigate the platform. Through this iterative feedback loop, we can continuously refine and optimize the application to better serve the needs of our users and provide a seamless and enjoyable booking experience.

# 7 References

[1] <http://amazon.com/>

[2] [User-User Collaborative Filtering For Jokes Recommendation](https://towardsdatascience.com/user-user-collaborative-filtering-for-jokes-recommendation-b6b1e4ec8642)

[3] [Global air travel demand reaches 94% of pre-pandemic levels, says IATA](https://moodiedavittreport.com/global-air-travel-demand-reaches-94-of-pre-pandemic-levels-says-iata/)

[4] <https://jestjs.io/>

[5] <https://www.cypress.io/>

[6] <https://nextjs.org/>

[7] <https://react.dev/>

[8] <https://tailwindcss.com/>

[9] <https://www.mongodb.com/>