TripOTrail

Your Adventure awaits!

REDEFINING THE TRAVEL EXPERIENCE!

By Ashwani Balakrishnan Neminimadathil

OVERVIEW

- Introduction
- Limitations of Existing Systems
- User Perspective
- Motivation
- Proposed Solution
- ER Diagram
- Security Measures
- Features implementation
- Demo
- Conclusion
- Future Work
- Q&A

INTRODUCTION

TripOTrail is an all-in-one travel platform that empowers users to:

- Plan personalized itineraries
- Optimize travel routes
- Collaborate with co-travelers
- Split and manage expenses
- Upload images, reels, and documents
- Modify routes on the go with real-time timeline updates

Smart, **flexible**, and **stress-free travel**—from planning to return.

LIMITATIONS OF TRADITIONAL APPLICATIONS

Fragmented Tools

Separate apps for booking, planning, and navigation lead to a disjointed experience.

Limited Group planning Features

Limited support for group planning or sharing of itineraries.

Poor Expense Management

No built-in tools to track and split costs among travelers.

Manual Planning Effort

Lack of automation makes planning tedious and error-prone.

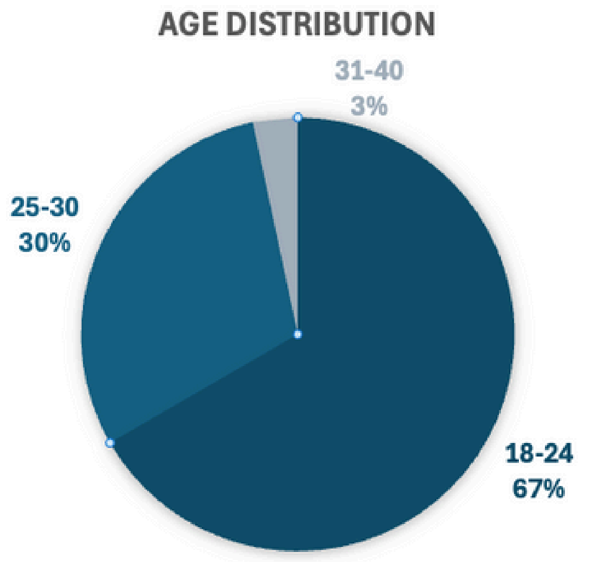
USER PERSPECTIVE

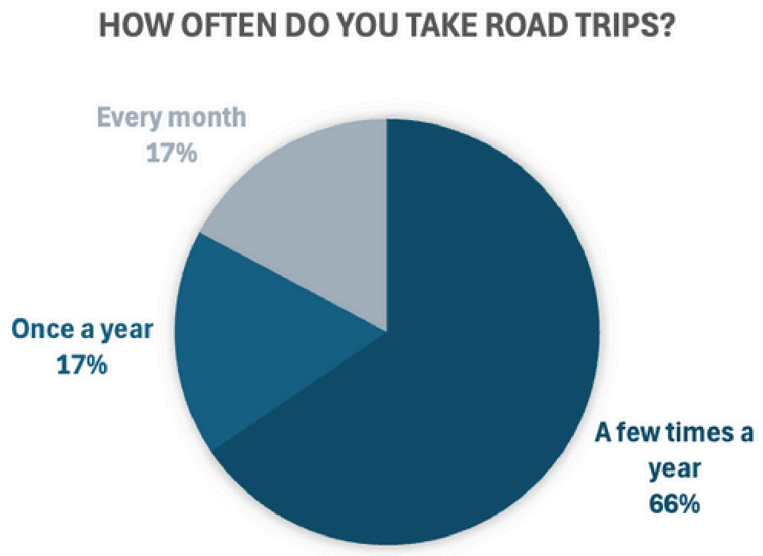
To better understand real user needs, a survey was conducted targeting individuals aged 18–40, the most active demographic for road trips and travel planning. It focused on current planning habits, key challenges, and interest in an all-in-one customizable travel tool, helping shape TripOTrail’s core features.

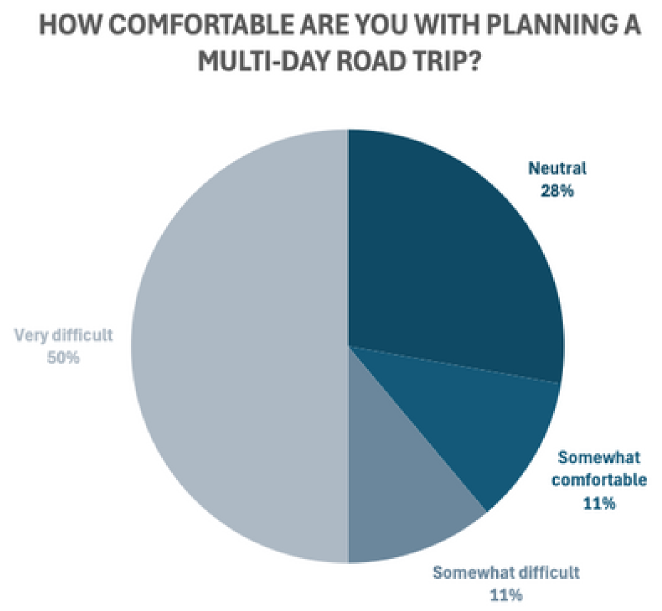
Key Areas Covered:

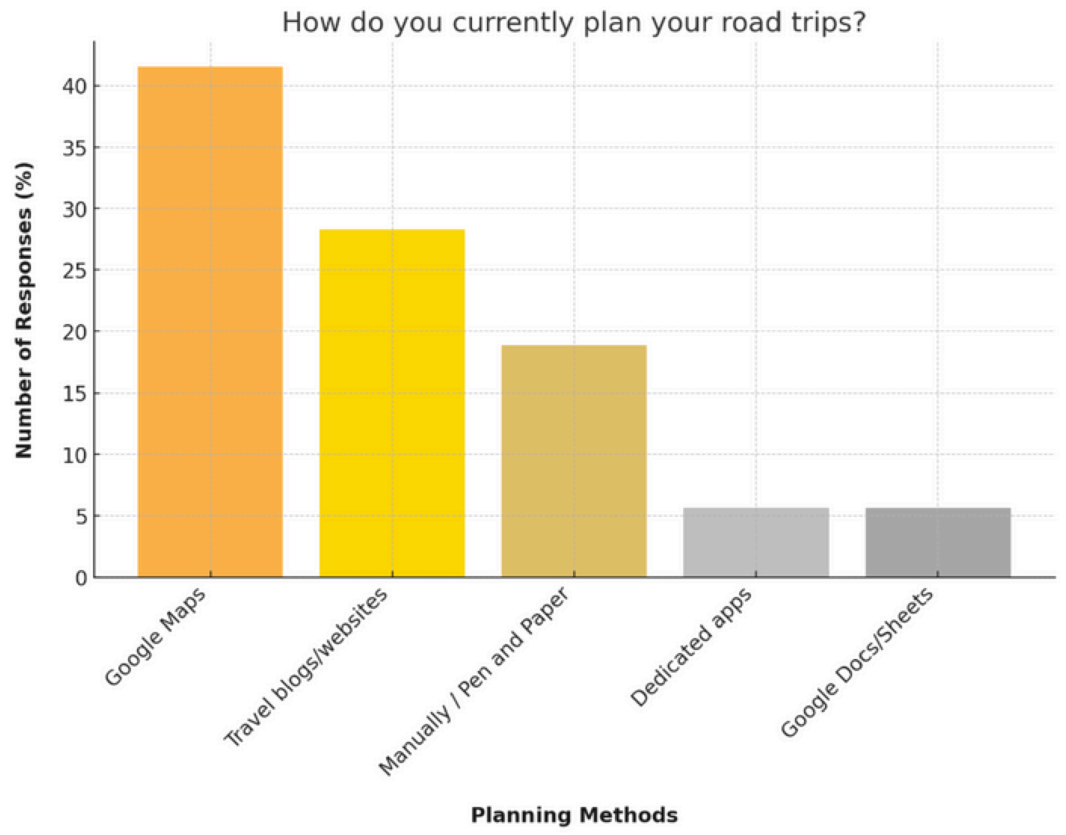
- Travel frequency and group size
- Planning methods and comfort level
- Common planning pain points
- Existing apps used and their limitations
- Feature interest:
  - Stop search & addition
  - Day-wise timeline builder
  - Expense tracking & bill-splitting
  - Collaborative planning
  - Itinerary updates during trip
  - Post-trip summaries

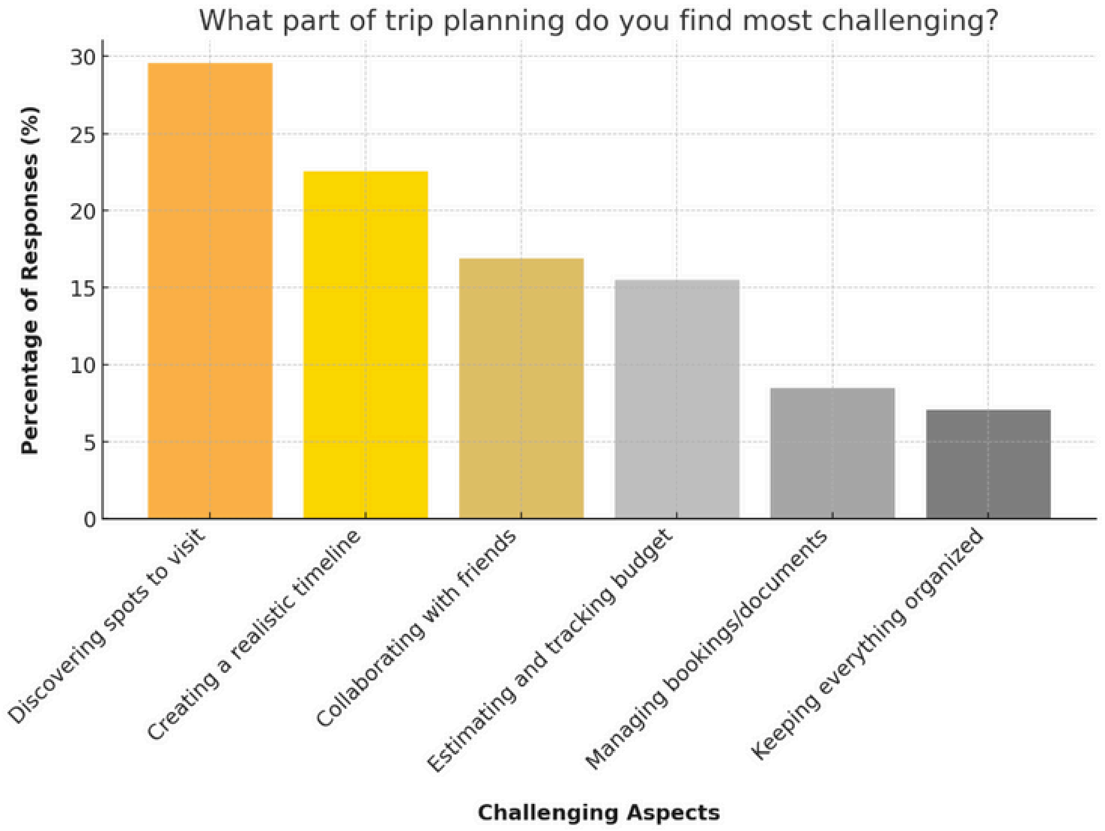
SURVEY INSIGHTS

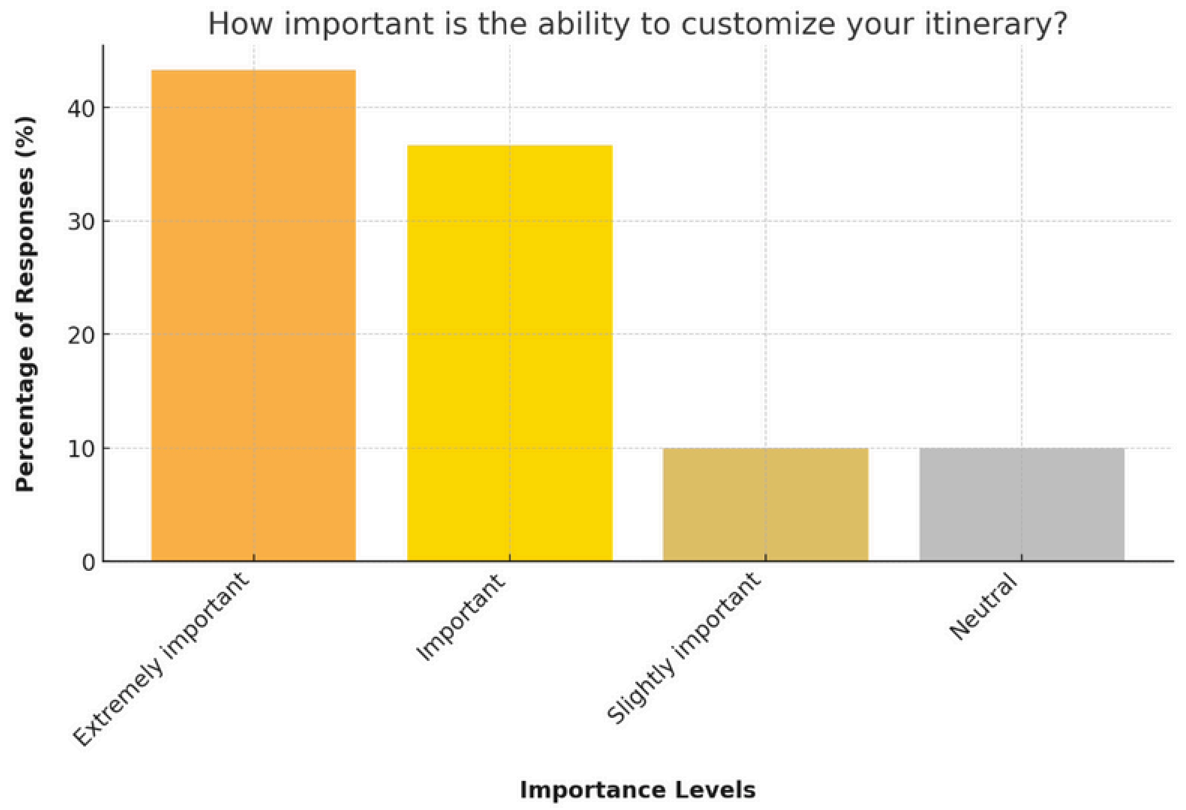




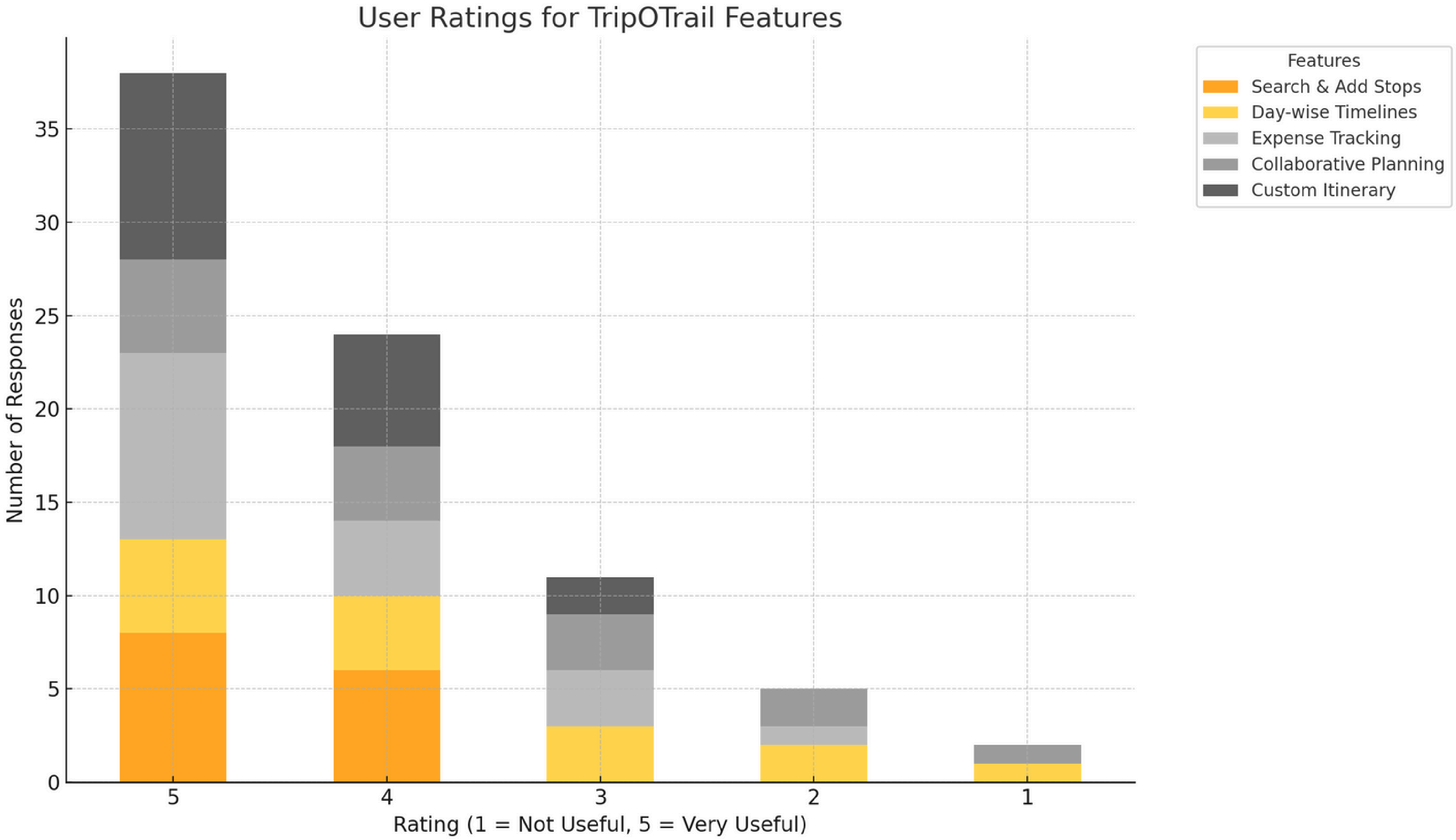






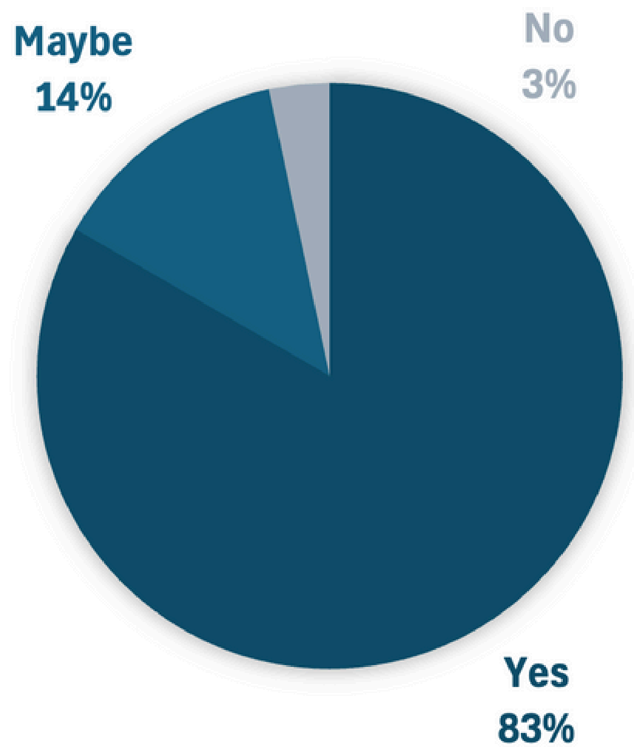


SURVEY INSIGHTS (CONT’D.)



User Ratings for TripOTrail Features

WOULD YOU BE WILLING TO TRY A NEW ALL-IN-ONE TRIP PLANNING APP?



MOTIVATION

01 Eliminate the hassle of complex trip planning

02 Minimize miscommunication in group travel

03 Streamline route planning and expense tracking

04 Enhance overall travel experience

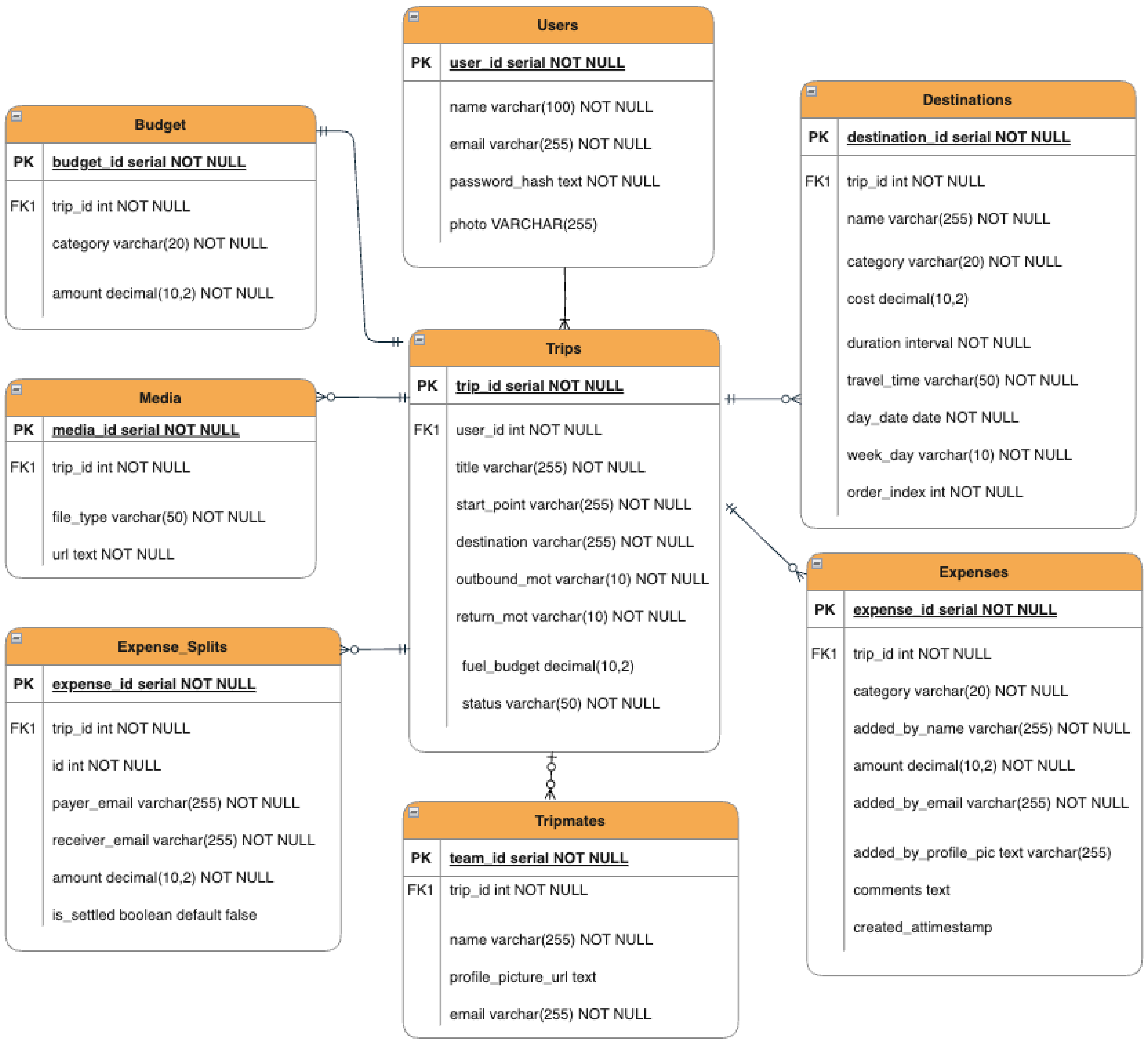
PROPOSED SOLUTION

- **All-in-one platform** for complete trip planning and management
- Customizable, timeline-based itinerary with intuitive drag-and-drop support.
- Seamless integration of route planning, document uploads, and expense tracking.
- Real-time trip updates: easily add stops, bills, and photos during the journey.
- Collaborative features like itinerary sharing and group expense splitting.
- Deliver a cohesive, stress-free, and memorable travel experience.

SYSTEM DESIGN

- **Frontend**: React.js
- **Backend**: Node.js with Express
- **Database**: PostgreSQL
- **UI/UX Design**: Figma
- **API Documentation**: Swagger

Entity Relationship Diagram



Budget

PK	budget_id serial NOT NULL
FK1	trip_id int NOT NULL
	category varchar(20) NOT NULL
	amount decimal(10,2) NOT NULL

Media

PK	media_id serial NOT NULL
FK1	trip_id int NOT NULL
	file_type varchar(50) NOT NULL
	url text NOT NULL

Expense_Splits

PK	expense_id serial NOT NULL
FK1	trip_id int NOT NULL
	id int NOT NULL
	payer_email varchar(255) NOT NULL
	receiver_email varchar(255) NOT NULL
	amount decimal(10,2) NOT NULL
	is_settled boolean default false

Users

PK	user_id serial NOT NULL
	name varchar(100) NOT NULL
	email varchar(255) NOT NULL
	password_hash text NOT NULL
	photo VARCHAR(255)

Trips

PK	trip_id serial NOT NULL
FK1	user_id int NOT NULL
	title varchar(255) NOT NULL
	start_point varchar(255) NOT NULL
	destination varchar(255) NOT NULL
	outbound_mot varchar(10) NOT NULL
	return_mot varchar(10) NOT NULL
	fuel_budget decimal(10,2)
	status varchar(50) NOT NULL

Tripmates

PK	team_id serial NOT NULL
FK1	trip_id int NOT NULL
	name varchar(255) NOT NULL
	profile_picture_url text
	email varchar(255) NOT NULL

Destinations

PK	destination_id serial NOT NULL
FK1	trip_id int NOT NULL
	name varchar(255) NOT NULL
	category varchar(20) NOT NULL
	cost decimal(10,2)
	duration interval NOT NULL
	travel_time varchar(50) NOT NULL
	day_date date NOT NULL
	week_day varchar(10) NOT NULL
	order_index int NOT NULL

Expenses

PK	expense_id serial NOT NULL
FK1	trip_id int NOT NULL
	category varchar(20) NOT NULL
	added_by_name varchar(255) NOT NULL
	amount decimal(10,2) NOT NULL
	added_by_email varchar(255) NOT NULL
	added_by_profile_pic text varchar(255)
	comments text
	created_attimestamp

SECURE USER AUTHENTICATION (JWT)

- **Stateless, Token-Based Authentication**:

  Secure access without maintaining server-side sessions.

- **Sessionless Login Management**:

  Enhances performance and simplifies authentication flow.

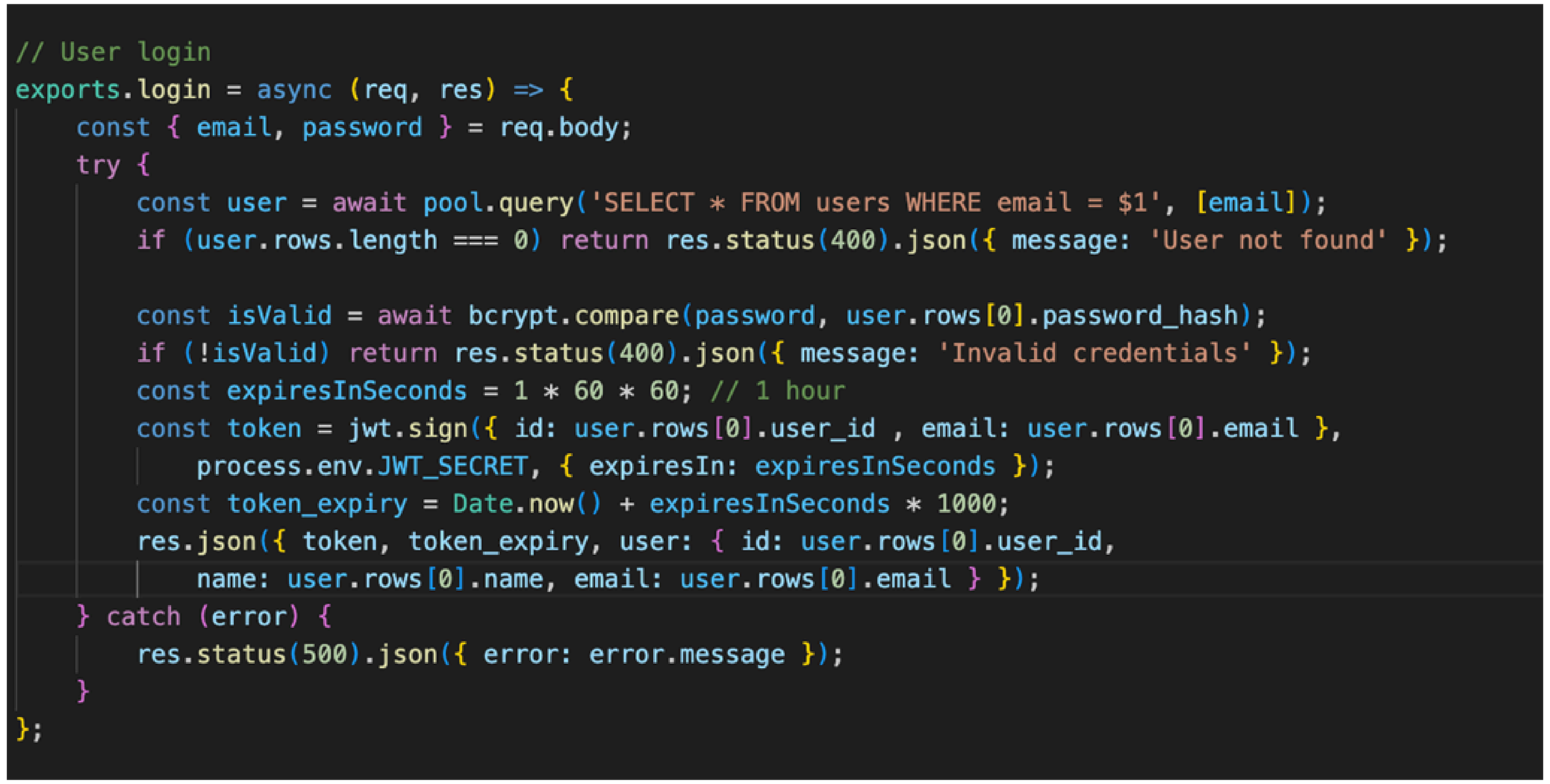
- **Data Protection**:

  Shields personal trip plans and expenses from unauthorized access.

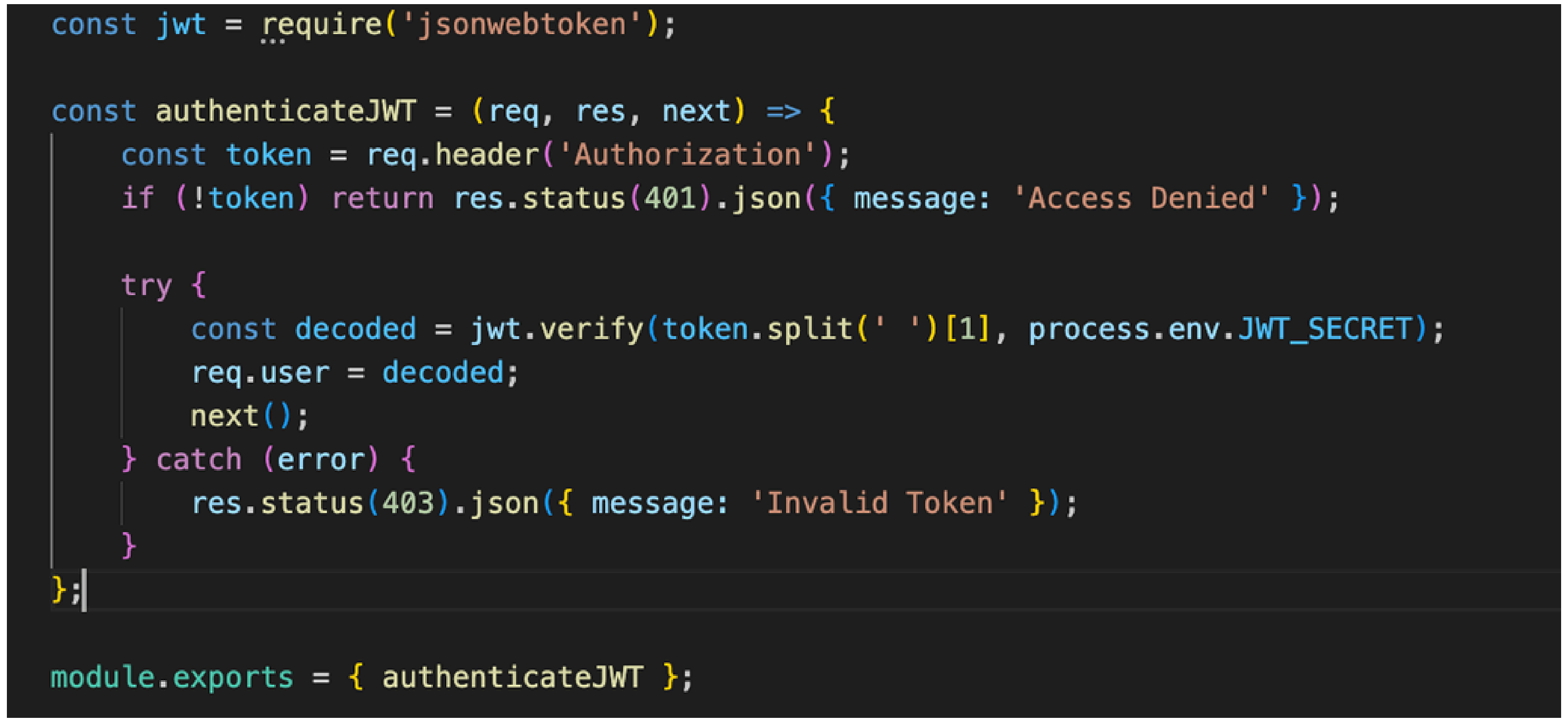
- **Enhanced User Experience**:

  Enables secure, seamless interactions across the platform.

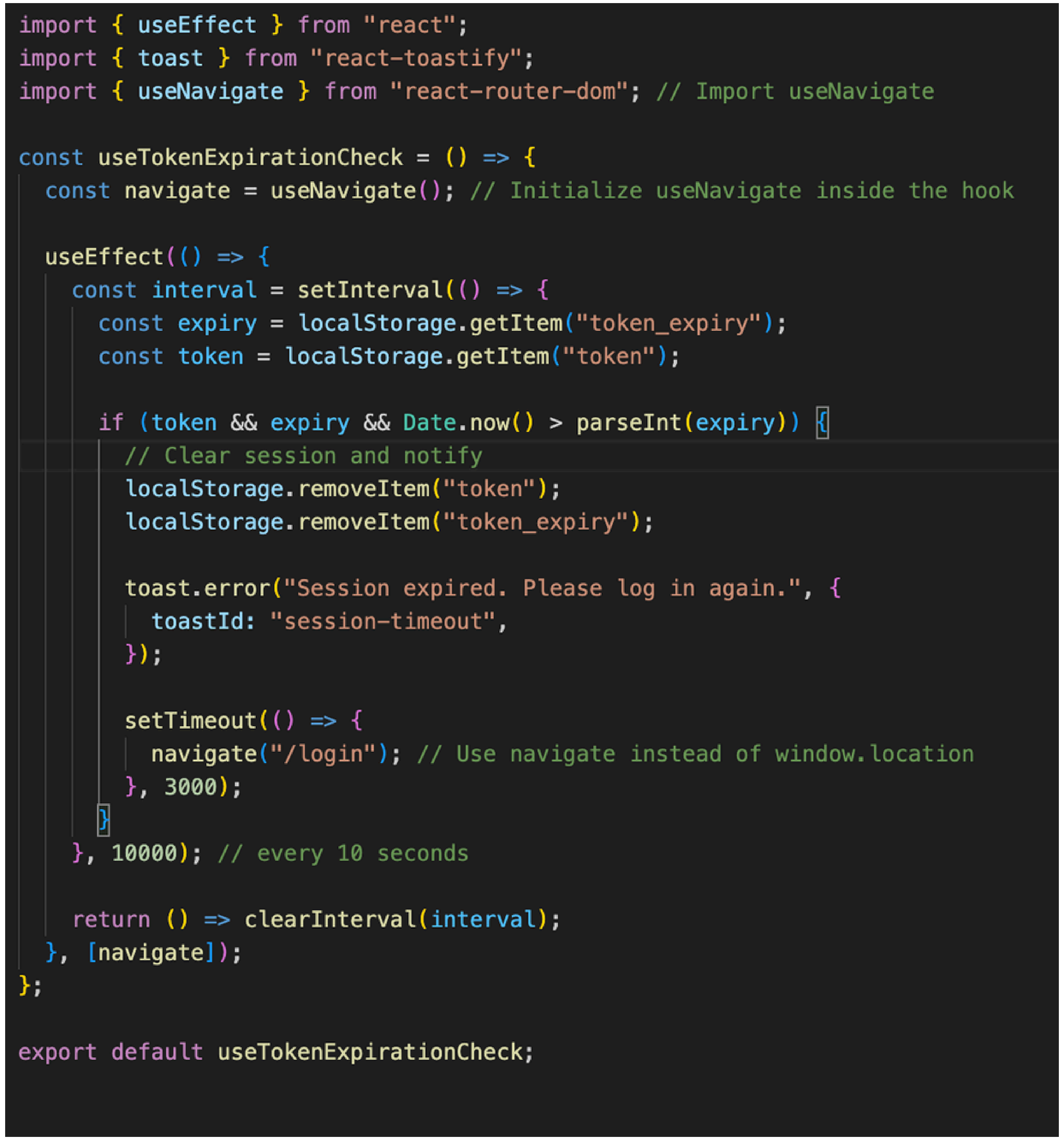
SECURE USER AUTHENTICATION (JWT)



```
// User login
exports.login = async (req, res) => {
    const { email, password } = req.body;
    try {
        const user = await pool.query('SELECT * FROM users WHERE email = $1', [email]);
        if (user.rows.length === 0) return res.status(400).json({ message: 'User not found' });

        const isValid = await bcrypt.compare(password, user.rows[0].password_hash);
        if (!isValid) return res.status(400).json({ message: 'Invalid credentials' });
        const expiresInSeconds = 1 * 60 * 60; // 1 hour
        const token = jwt.sign({ id: user.rows[0].user_id , email: user.rows[0].email },
            process.env.JWT_SECRET, { expiresIn: expiresInSeconds });
        const token_expiry = Date.now() + expiresInSeconds * 1000;
        res.json({ token, token_expiry, user: { id: user.rows[0].user_id,
            name: user.rows[0].name, email: user.rows[0].email } });
    } catch (error) {
        res.status(500).json({ error: error.message });
    }
};
```



```
const jwt = require('jsonwebtoken');

const authenticateJWT = (req, res, next) => {
    const token = req.header('Authorization');
    if (!token) return res.status(401).json({ message: 'Access Denied' });

    try {
        const decoded = jwt.verify(token.split(' ')[1], process.env.JWT_SECRET);
        req.user = decoded;
        next();
    } catch (error) {
        res.status(403).json({ message: 'Invalid Token' });
    }
};

module.exports = { authenticateJWT };
```

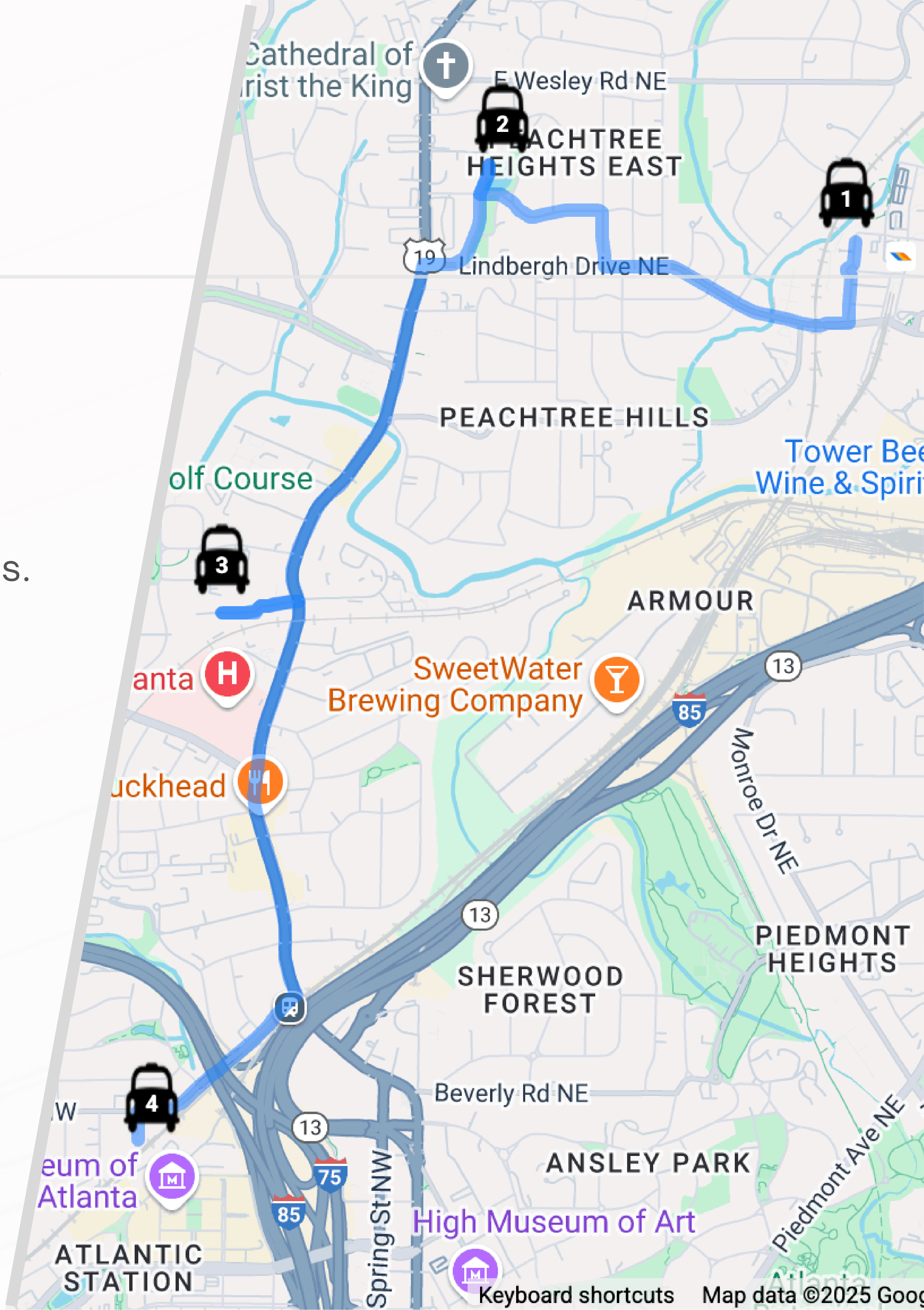


```
import { useEffect } from "react";
import { toast } from "react-toastify";
import { useNavigate } from "react-router-dom"; // Import useNavigate

const useTokenExpirationCheck = () => {
  const navigate = useNavigate(); // Initialize useNavigate inside the hook

  useEffect(() => {
    const interval = setInterval(() => {
      const expiry = localStorage.getItem("token_expiry");
      const token = localStorage.getItem("token");

      if (token && expiry && Date.now() > parseInt(expiry)) {
        // Clear session and notify
        localStorage.removeItem("token");
        localStorage.removeItem("token_expiry");

        toast.error("Session expired. Please log in again.", {
          toastId: "session-timeout",
        });

        setTimeout(() => {
          navigate("/login"); // Use navigate instead of window.location
        }, 3000);
      }
    }, 10000); // every 10 seconds

    return () => clearInterval(interval);
  }, [navigate]);
};

export default useTokenExpirationCheck;
```

GOOGLE MAPS API

To enhance trip planning efficiency and user experience, the following Google Maps APIs were integrated:

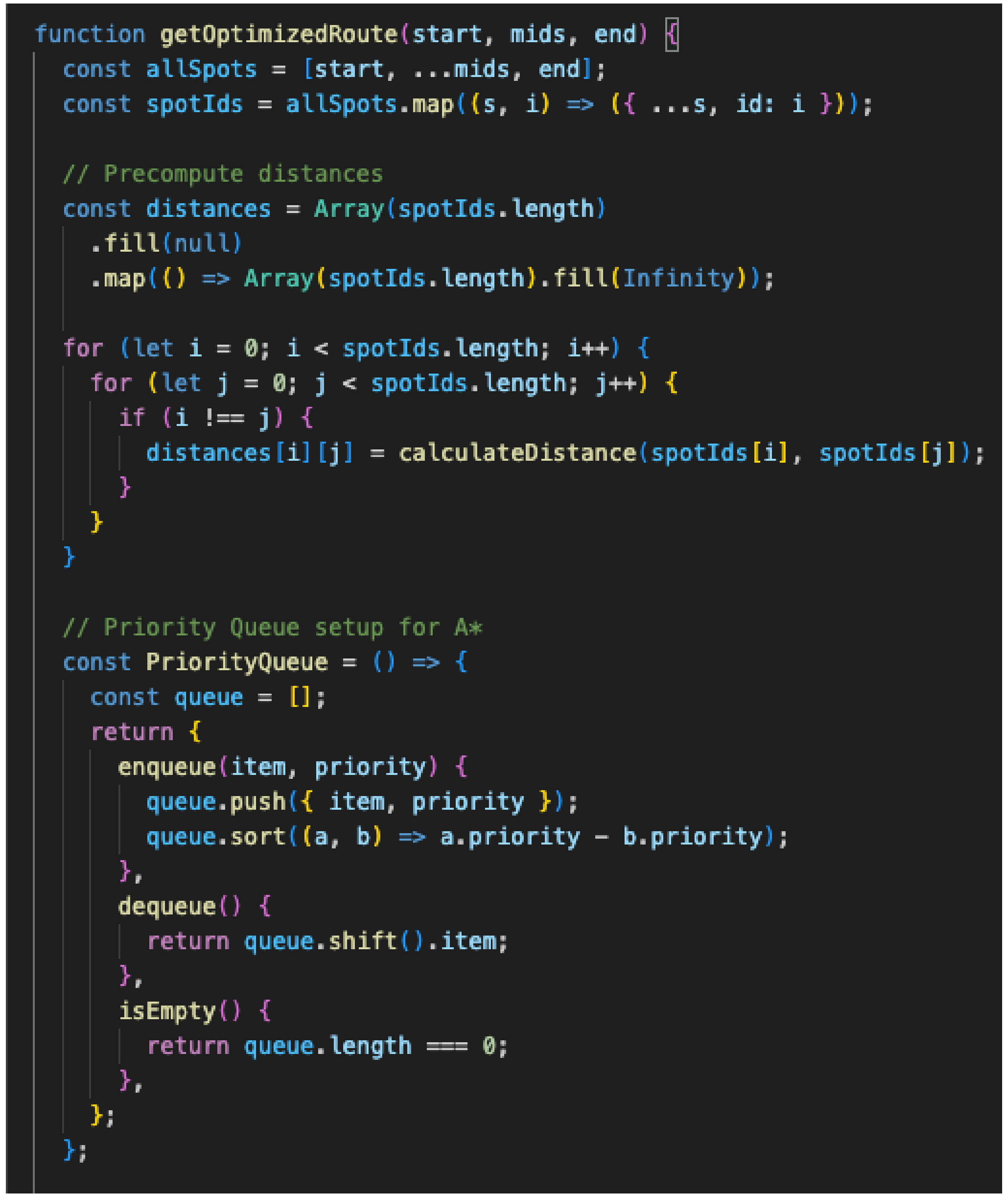
- **Places API** – Location autocomplete & discovery of stops with ratings.
- **Maps JavaScript API** – Dynamic map previews & route visualization.
- **Distance Matrix API** – Real-time travel durations between stops.
- **Geocoding API** – Address to coordinate conversion for mapping.
- **Directions API** – Optimized routes across multiple locations.



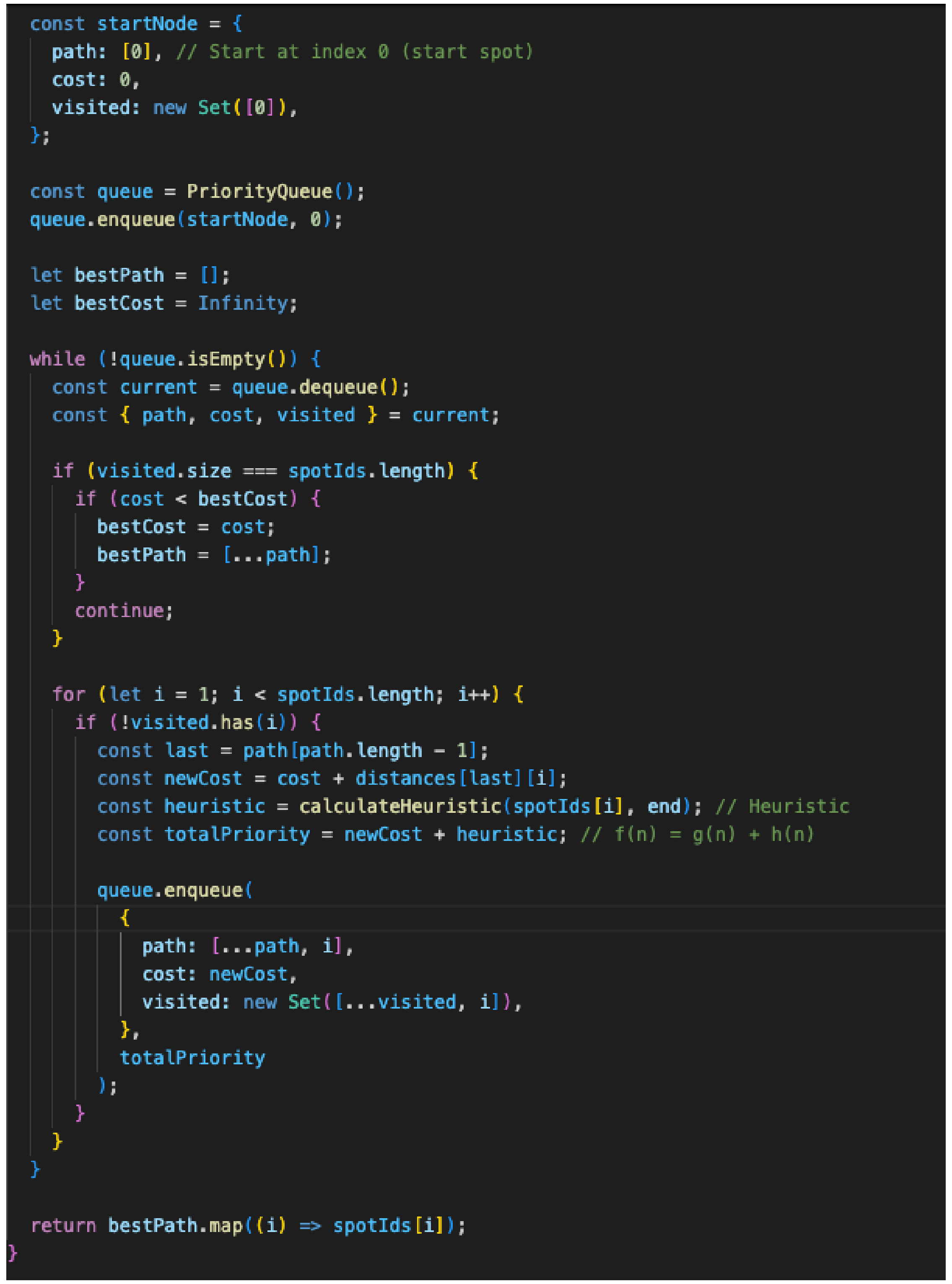
SMART PLANNING & ROUTING

- Utilized **A* pathfinding** to ensure optimal routing between stops.
- Optimizes travel time and enhances trip efficiency.
- Drag-and-drop itinerary with real-time updates and on-the-go edits.
- Supports dynamic changes to accommodate evolving travel plans.

SMART PLANNING & ROUTING



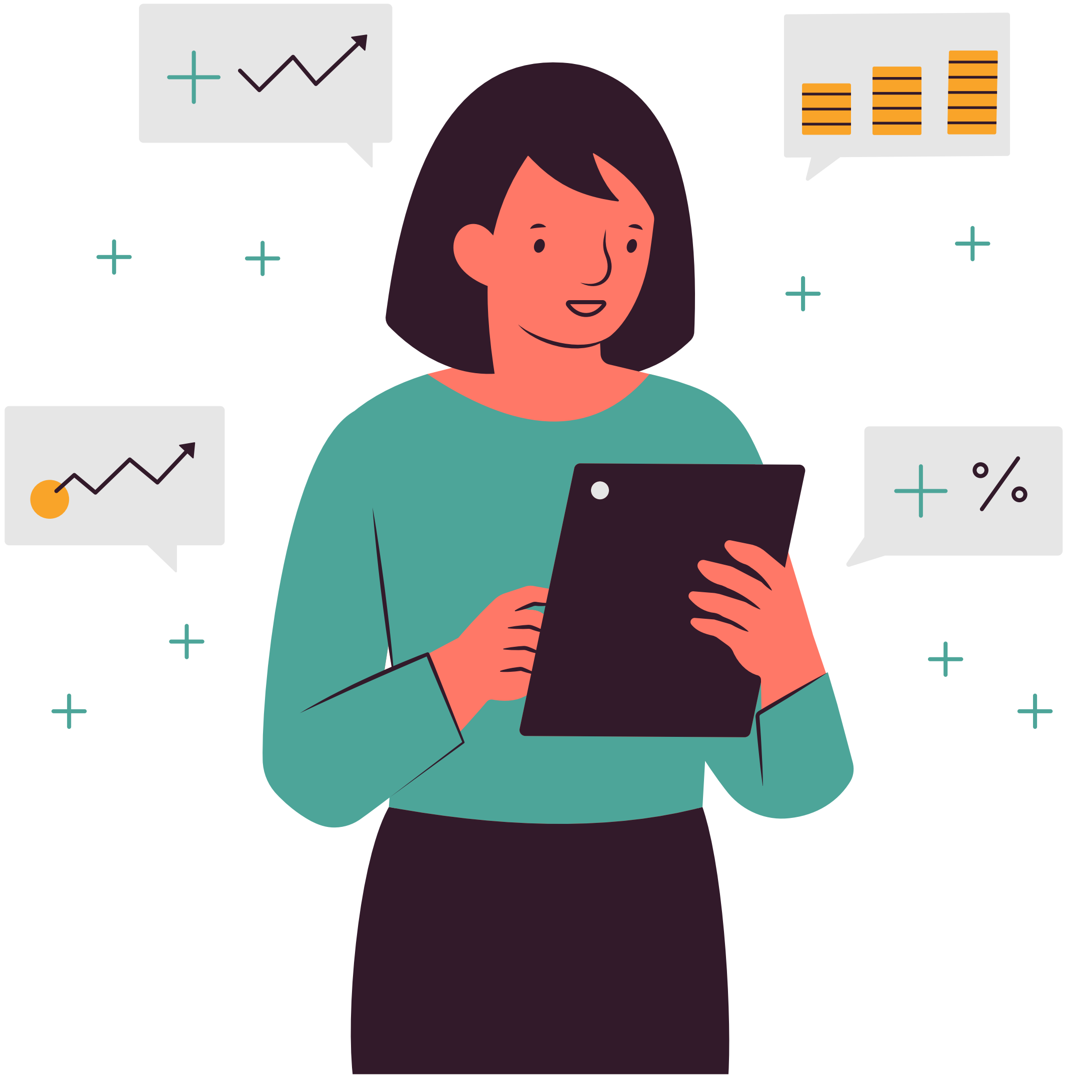
```javascript
function getOptimizedRoute(start, mids, end) {
  const allSpots = [start, ...mids, end];
  const spotIds = allSpots.map((s, i) => ({ ...s, id: i }));

  // Precompute distances
  const distances = Array(spotIds.length)
    .fill(null)
    .map(() => Array(spotIds.length).fill(Infinity));

  for (let i = 0; i < spotIds.length; i++) {
    for (let j = 0; j < spotIds.length; j++) {
      if (i !== j) {
        distances[i][j] = calculateDistance(spotIds[i], spotIds[j]);
      }
    }
  }

  // Priority Queue setup for A*
  const PriorityQueue = () => {
    const queue = [];
    return {
      enqueue(item, priority) {
        queue.push({ item, priority });
        queue.sort((a, b) => a.priority - b.priority);
      },
      dequeue() {
        return queue.shift().item;
      },
      isEmpty() {
        return queue.length === 0;
      },
    };
  };
```



```javascript
  const startNode = {
    path: [0], // Start at index 0 (start spot)
    cost: 0,
    visited: new Set([0]),
  };

  const queue = PriorityQueue();
  queue.enqueue(startNode, 0);

  let bestPath = [];
  let bestCost = Infinity;

  while (!queue.isEmpty()) {
    const current = queue.dequeue();
    const { path, cost, visited } = current;

    if (visited.size === spotIds.length) {
      if (cost < bestCost) {
        bestCost = cost;
        bestPath = [...path];
      }
      continue;
    }

    for (let i = 1; i < spotIds.length; i++) {
      if (!visited.has(i)) {
        const last = path[path.length - 1];
        const newCost = cost + distances[last][i];
        const heuristic = calculateHeuristic(spotIds[i], end); // Heuristic
        const totalPriority = newCost + heuristic; // f(n) = g(n) + h(n)

        queue.enqueue(
          {
            path: [...path, i],
            cost: newCost,
            visited: new Set([...visited, i]),
          },
          totalPriority
        );
      }
    }
  }

  return bestPath.map((i) => spotIds[i]);
}
```

SMART EXPENSE TRACKING

- Effortlessly log and categorize trip expenses
- Built-in Splitwise-style bill splitting
- Auto-calculates shares and balances
- “**Who Owes Whom**” view for clear tracking
- Real-time sync across collaborators
- Monitors settled and pending payments



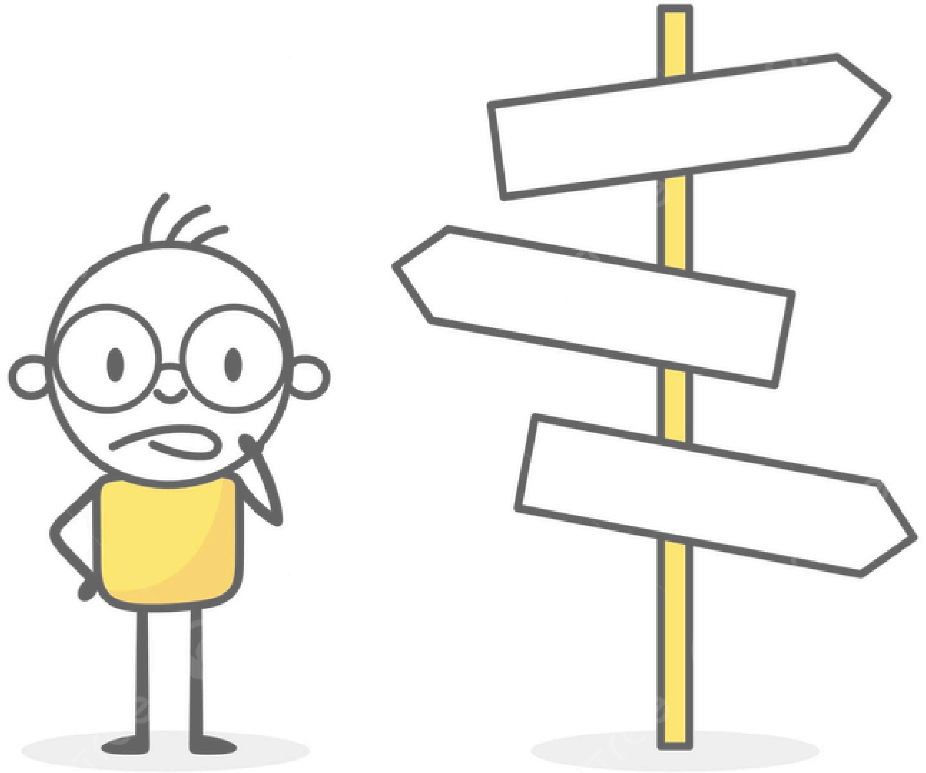
LIVE DEMONSTRATION

A walkthrough showcasing TripOTrail's key features in real time.

TripOTrail

FUTURE WORK

- Real-Time Weather Integration
- Smart Travel Assistant(Instant Messaging)
- Reminders/Alerts for significant information
- AI-powered recommendations based on traveler preferences
- Native mobile application with on-the-go trip edits
- Integration of Payment Gateway



“ And just like that, we are on our way to everywhere... ”

THANK YOU!