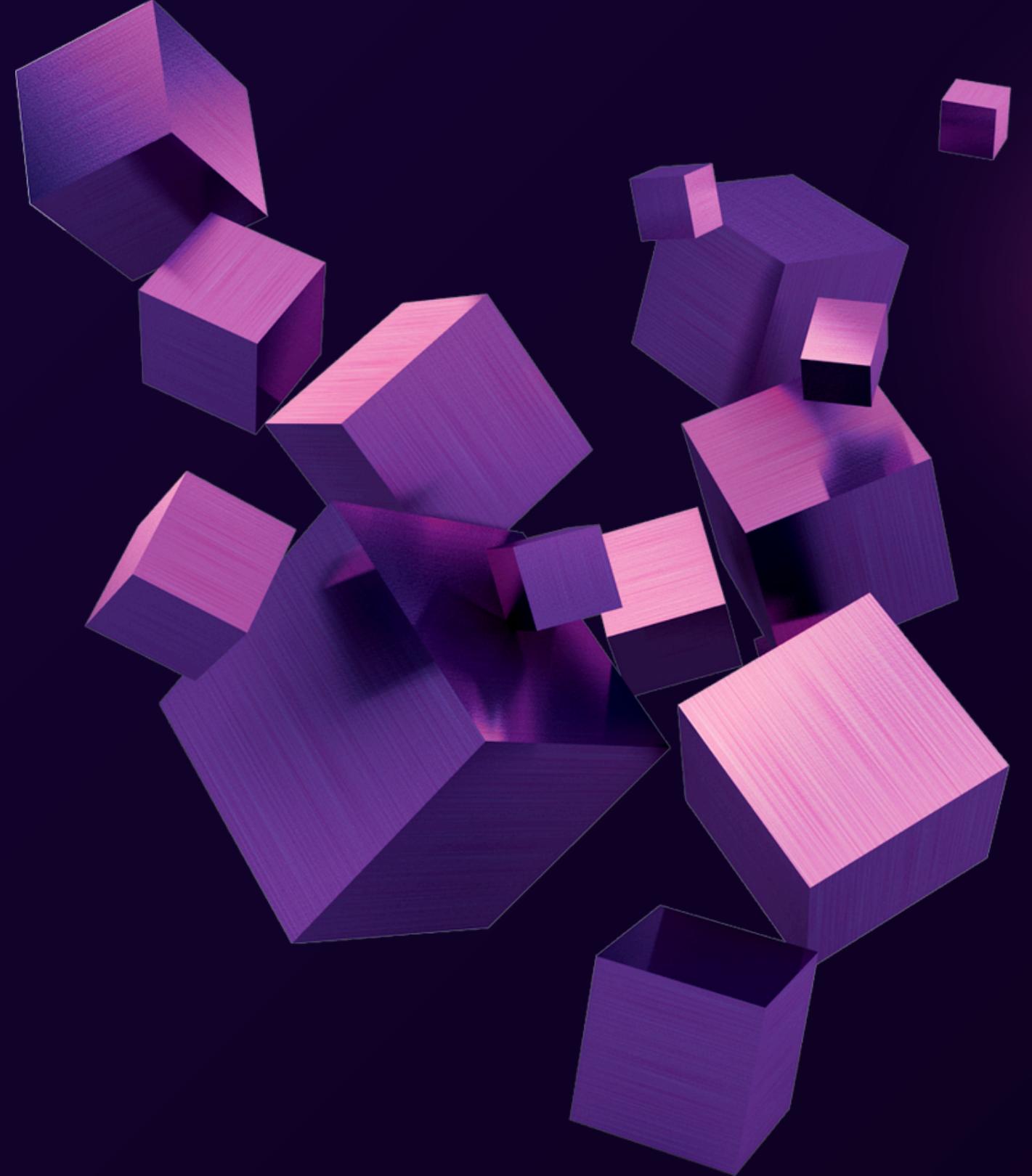


# TRAVEL ASSISTANT

LEISURE





# ABOUT US

12214760

Javokhir Yuldashev (Team Leader)

12225252

Mironshoh Muhammedov

12214762

Firdavs Bakhtiyorov

12225247

Mansur Ilhomov



# About Leisure

Leisure is an interactive web application that helps users discover and plan activities based on their available time and preferences. The application uses geolocation to find nearby attractions and provides detailed routing information to help users navigate to their chosen destinations.



# OUR PROJECT

**01**

## Personalized Time & Distance-Based Exploration:

Users simply input how much free time they have and how far they're willing to travel. Our web app filters real-world options based on these constraints, making it easy to discover activities that fit into any schedule — from quick coffee breaks to half-day adventures.

**02**

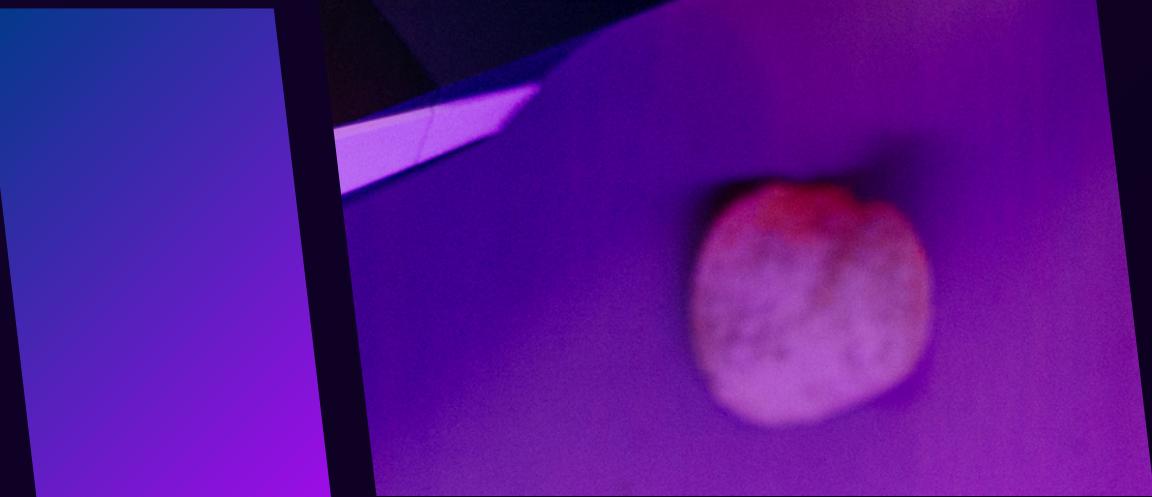
## Powered by Gemini AI for Smarter Recommendations:

We integrated the Google Gemini model to generate intelligent and context-aware suggestions. Gemini analyzes user inputs, nearby locations, weather, and user intent to provide meaningful, natural language activity suggestions tailored to the moment.

**03**

## Location-Aware and User-Friendly Design:

Using geolocation services, the platform automatically detects the user's current position to serve hyperlocal results. The clean, intuitive UI ensures a smooth experience, allowing users to explore without planning stress — just tap, discover, and go.





## OUR VISION

01

### Empower Everyday Exploration:

We envision a world where even a small break in your day becomes an opportunity for discovery, joy, and connection with your surroundings.

02

### Make Smart, Spontaneous Travel Effortless:

WanderTime aims to remove the planning burden and make short, meaningful getaways available to everyone — powered by intelligent, real-time suggestions.

# ✨ FEATURES

01

- ⌚ Personalized suggestions based on free time & distance

02

- 📍 Real-time geolocation via browser

03

- 🧭 Route and geocode handling with GraphHopper API

04

- 🧠 Natural language suggestions via Gemini 2.5 Pro

05

- 📱 Minimal, responsive UI (React + Tailwind)

06

- gMaps Map integration for suggestion visualization



# USED TECHNOLOGIES



## Frontend

01

- **React.js:** For building a dynamic and responsive user interface.
- **Tailwind CSS:** For clean, customizable styling.
- **Geolocation API:** To detect the user's real-time location.

## Backend

02

- **FastAPI:** A modern, fast (high-performance) web framework for building APIs with Python.
- **GraphHopper Geocode API:** Converts user input into precise geographic coordinates.
- **GraphHopper Route API:** Calculates travel distances and durations based on real-world routes.

## AI Integration

03

- **Google Gemini API:** Powers natural language recommendations based on context.

## AI Integration

04

- **Vercel:** For deploying and hosting the web app.
- **Git & GitHub:** For version control and collaboration.

# DOCUMENTATION

```
team.software-engineering/
├── public/                      # Static files
├── src/                         # Frontend components and logic
└── backend/                     # API logic and Gemini integration
├── assets/                       # Images and icons
├── .env.example                 # Environment variable sample
├── vercel.json                  # Deployment config
└── README.md                    # Project documentation
```

```
# Clone the repo
git clone https://github.com/inhacollab/team.software-engineering.git
cd team.software-engineering

# Install frontend dependencies
cd frontend
npm install

# Setup backend (FastAPI or Node.js)
cd ../backend
pip install -r requirements.txt # or npm install

# Configure .env files with your GraphHopper & Gemini keys

# Run frontend and backend
npm run dev   # React frontend
uvicorn main:app --reload  # FastAPI backend
```

# GRAPHHOPPER

## WHY GRAPHHOPPER?

- Fast and reliable for real-time use cases
- Based on open-source data (OpenStreetMap)
- Easy to integrate with our Python backend
- Lightweight and cost-effective compared to alternatives like Google Directions API



## GEOCODING API

- Converts user-friendly place names or addresses into precise geographic coordinates.
- Enables users to type in any location — e.g., “Seoul Station” — and get exact map points for routing.



## ROUTE API

- Calculates realistic travel distances and estimated durations.
- Supports different travel modes (e.g., walking, driving, cycling).
- Helps filter out suggestions that are too far or time-consuming.



# GEMINI INTEGRATION

## (GEMINI-2.5-PRO-EXP-03-25)

01

We chose gemini-2.5-pro-exp-03-25, a powerful language model from Google, for its cutting-edge performance and up-to-date training.

01



## Why This Model?

- **Latest Training Data:**

Trained on a dataset updated until January 25, 2025, allowing it to generate more relevant and context-aware suggestions.

- **Improved Context Handling:**

Supports few-shot learning, enabling the model to understand specific user preferences from minimal examples.

- **Natural Language Generation:**

Delivers highly personalized and human-like activity suggestions tailored to time, location, and user intent.



# UI/UX & LOCATION FUNCTIONALITY

## 💡 Thoughtful UI/UX Design:

01

- Clean, minimalistic layout designed for ease of use
- Mobile-first interface optimized for quick on-the-go decisions
- Focused on simplicity: just enter time & distance — and explore
- Smooth transitions and feedback to ensure an intuitive user journey

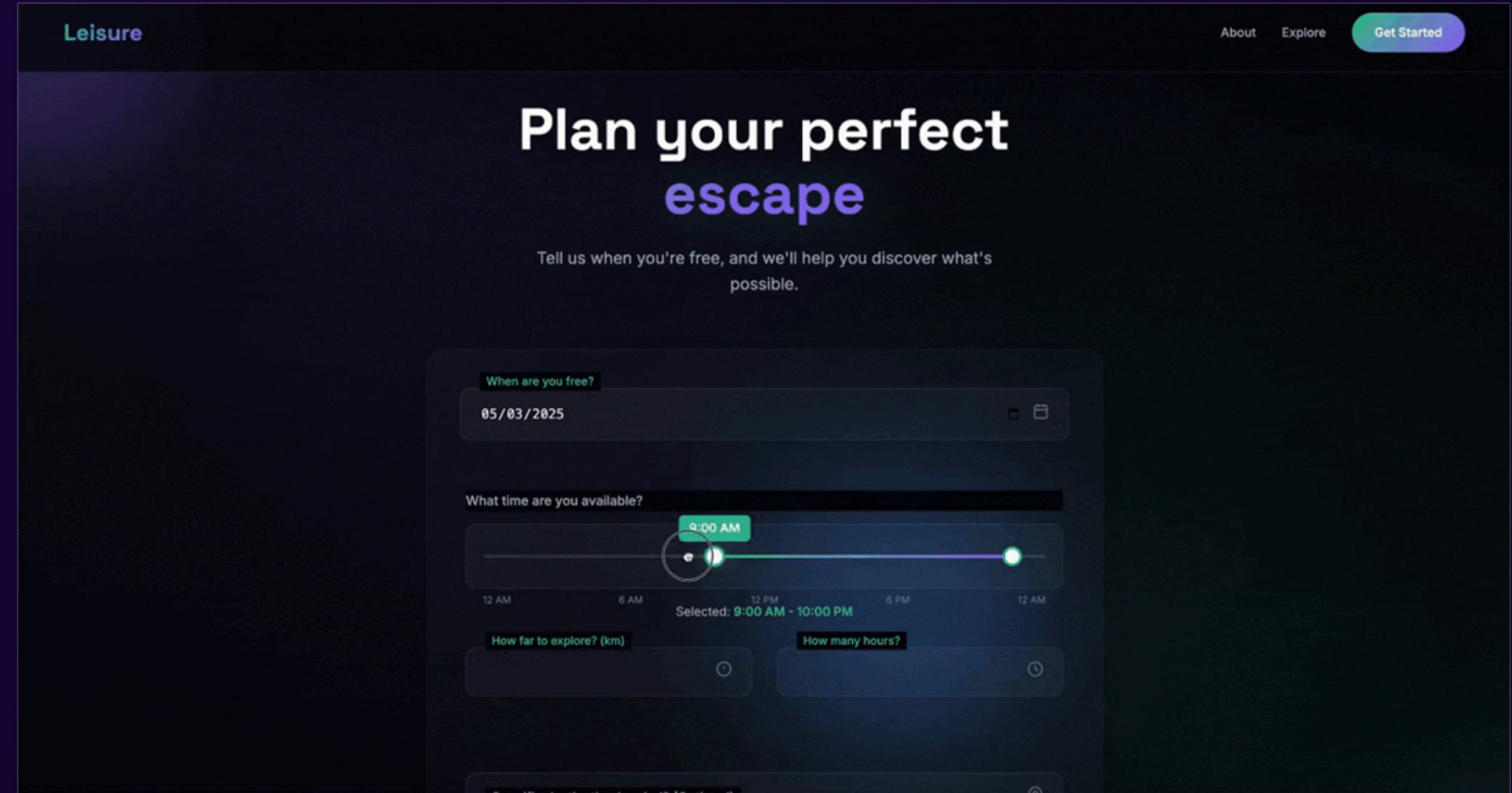
## 📍 Real-Time Location Extraction

02

- The webpage includes a built-in function to automatically extract the user's current latitude and longitude using the Geolocation API
- This function enables hyper-personalized suggestions by identifying nearby activities within the user's travel radius



# TEST RESULTS



The screenshot shows a dark-themed mobile application interface. At the top, there's a navigation bar with tabs for "Leisure", "About", "Explore", and a prominent "Get Started" button. The main heading "Plan your perfect escape" is displayed in large, bold, white font. Below it, a sub-instruction reads "Tell us when you're free, and we'll help you discover what's possible." A modal window is open, prompting the user to "When are you free?" with a date input set to "05/03/2025". Inside the modal, a timeline asks "What time are you available?" with a slider currently set between 9:00 AM and 10:00 PM, indicated by green dots. The timeline scale includes 12 AM, 6 AM, 12 PM, 6 PM, and 12 AM. At the bottom of the modal, there are two input fields: "How far to explore? (km)" with a location pin icon, and "How many hours?" with a clock icon.

Video



# FACED ISSUES

**01**

## Serving in HTTPS

To get users real time location we need to serve webpage in secure server. It was running smoothly in localhost but in time of deployment we faced little issue in security because of http server. But we solved this issue by deploying to Vercel server.

**02**

## Hiding credentials

We had two API keys to keep private. While .gitignore handled this locally, we initially thought we had to expose them in the GitHub repo for deployment with Vercel. However, after some research, we discovered Vercel's built-in feature for managing environment variables.



# THANK YOU