

# CityEcoScout: A Platform for Exploring Sustainable Locations Worldwide

Bader Eddine BENHIRT<sup>1</sup>, Yasmine FIHRI<sup>2</sup>, Ahmed Moubarak LAHLYAL<sup>2</sup>, Rahhal ERRATTAHI<sup>1</sup>

<sup>1</sup>LTI, National School of Applied Sciences, Chouaib Doukkali University, El Jadida, Morocco

<sup>2</sup>IITE, National School of Applied Sciences, Chouaib Doukkali University, El Jadida, Morocco

E-mail: bader.b389@ucd.ac.ma, fihri.y953@ucd.ac.ma, lahlyal.a450@ucd.ac.ma, errattahi.r@ucd.ac.ma,

---

## Article history

Received April 20, 2025

Revised May 5, 2025

Accepted May 6, 2025

Published May 7, 2025

## ABSTRACT

As people everywhere are trying to be more environmentally conscious, there is an increasing public demand for practical tools that help make sustainable decisions in cities and towns. That's what inspired CityEcoScout a new take on mobile platforms, using AI to inform users about environmentally focused spots in local areas and further afield. The application integrates several technologies, including Google Maps, Street View, and Places APIs, with the revolutionary Gemini AI engine to create something incredibly useful. The killer feature of this app is designed to provide easy-to-digest sustainability information and allows virtual tours of green destinations before arrival. CityEcoScout was developed to assist locals and travelers with making environmental choices, setting a new standard for how location services can enhance sustainability in daily lives.

**Keywords:** *sustainability, urban environments, eco-friendly technology, environmental analytics, mobile applications, geographic data*

---

## Code metadata

Current code version	V1.0
Permanent link to code/repository	<a href="https://github.com/yasminefhr1/CityEcoScout">https://github.com/yasminefhr1/CityEcoScout</a>
Legal Code License	MIT License
Code versioning system used	Git
Software code languages, tools, and services used	Java 21, Kotlin, Spring Boot 3.2.5, Spring Cloud 2023.0.1, MySQL, Spring Security, Docker, Retrofit 2.9.0, Google Maps API, Street View API, Places API, Flask, Gemini AI
Compilation requirements, operating environments & dependencies	Java 21, Spring Boot 3.2.5, Maven 3.8.1+, MySQL 8+, Android SDK 34/24+, Gradle, ARCore 1.29.0, Lombok 1.18.30, Google Material Design 1.9.0
Link to developer documentation/manual	<a href="https://github.com/yasminefhr1/CityEcoScout/blob/main/README.md">https://github.com/yasminefhr1/CityEcoScout/blob/main/README.md</a>
Support email for questions	fihri.y953@ucd.ac.ma, lahlyal.a450@ucd.ac.ma

## I. INTRODUCTION

There is an increasing scale of urgency to save the planet, and the need for necessary steps toward a more sustainable and environmentally friendly lifestyle has never been more pressing. Still, remedies that actually point to where and how individuals can find places that sustain the environment are generally limited in presenting depth and interactivity, failing to enlighten people accurately as well as informatively about the environmental

and social consequences that choices cause to surroundings [1–3].

Most recently, tourism has been identified as one of the significant contributors to carbon emissions—several researches reported how tourism has contributed remarkably to local exploration. Further estimates declare that much, if not all, of the global CO<sub>2</sub> emissions come from tourism-related activities [4]. Tourism also comes along with a couple of challenging is-

sues such as overtourism—over-concentration of visitors—and undertourism—inequitable benefits derived from tourism—which have, over the years, become critical issues ringing bells for the environment as well as the communities [5]. Technological innovations are, thus far, called to save the day and modify the way users are guided into making more eco-friendly and socially responsible choices [6].

In response to this challenge, various AI-powered mobile applications have been developed to enhance urban sustainability and user engagement. For instance, AirTOWN offers real-time, pollution-aware recommendations for points of interest (POIs), integrating air quality data to assist users in making health-conscious decisions [7]. Similarly, advancements in tourism recommender systems have been achieved by incorporating sustainability metrics, such as CO<sub>2</sub> emissions and seasonal demand, to promote responsible travel choices [8].

While advances have been made, existing solutions address only fragments of the full sustainability problem. There doesn't seem to be an integrated system providing rigorously sustainability-oriented recommendations, an AI-aided decision engine, and a user interface that promotes and explains sustainable choices. To summarize, there is still room for a more holistic location recommending tool that ties powerful sustainability modeling with effective guidance from a user-facing perspective.

To close this existing gap, CityEcoScout shall offer a very simple but technology-oriented online platform to examine sustainable locations. By joining robust APIs such as the Google Map API, the Google Street View API, the Google Places API, and the Gemini chatbot, CityEcoScout provides users with information, maps, and context about the sustainability of various places in real-time. This enables users to make informed decisions and further promotes a movement towards responsible, sustainable living.

CityEcoScout's significance rests in merging advanced geolocation technologies with real-time indicators of sustainability to foster the revolutionary way of finding green spots. CityEcoScout aggregates an integrated sustainability-oriented recommendation model that assesses not only environmental impact but also factors such as popularity and other contextual determinants when recommending locations. The platform further builds a complete mobile application that leverages the capabilities of Google's Gemini AI to deliver deep navigation-related personalized content for each recommendation and foster better decision-making. On top of this, the design embeds a UI geared toward sustainable decision-making, including well-established UI elements like map visualization, 360° visuals, and a comparative card layout. CityEcoScout demonstrates through implemented examples how, by its combined approach, recommendations cater to sustainability goals while permitting

users their contentment, as depicted in Figures 2 and 3.

Through these advancements, CityEcoScout serves as a valuable tool for individuals and communities striving to reduce their environmental footprint and advocate for a more sustainable future [9, 10].

The remainder of this paper is organized as follows: Section II reviews related work on sustainable location-based recommender systems. Section III details the software architecture, including backend and frontend components, external API integrations, and AI capabilities. Section IV presents five illustrative usage scenarios showcasing key features such as search filters, map and Street View visualizations, community posts, and AI chatbot interaction. Section V examines the environmental and social impact of the platform. Section VI evaluates code quality and integrity through metrics analysis and development practices. Section VII presents the security measures and incident-response strategy. Section VIII discusses limitations and avenues for future work. Finally, Section IX concludes the paper by summarizing our contributions.

## II. RELATED WORKS

In recent years, a growing body of research has focused on integrating sustainability into location-based and tourism recommendation systems, in alignment with CityEcoScout's primary objectives. This section reviews key contributions in the field, with particular emphasis on the in-depth works by Banerjee et al., who provided significant foundations in the areas of multi-stakeholder fairness, sustainability metrics, user interface design, and AI-based recommendation generation. Table 1 summarizes the comparison between existing works and our proposed solution.

### A. Fairness and Sustainability in Recommender Systems

Banerjee and Wörndl [11] introduced the concept of societal fairness (S-fairness) in tourism recommendation systems, advocating that recommendations should consider not only user preferences but also the impacts on non-user stakeholders, such as local residents, municipalities, and ecosystems. Their work highlighted the importance of mitigating negative externalities like pollution, congestion, and housing pressure. While their contributions primarily provided a conceptual framework, operational implementations were lacking. CityEcoScout addresses this gap by operationalizing S-fairness through real-time sustainable recommendations that balance user satisfaction with broader societal benefits.

### B. Sustainability-Focused Destination Recommendation

Banerjee et al. [12] proposed the Green Destination Recommender (GDR), a web platform suggesting city-trip destinations based on user location, travel timing, and interest profiles. Destinations were prioritized by estimated CO<sub>2</sub> emissions, popularity, and seasonality factors. Although GDR demonstrated multi-factor prioritization, it was limited to a small selection of European cities and preliminary evaluations. CityEcoScout extends this concept by offering a global mobile application with interactive features such as Google Maps and Street View integration, enabling more dynamic exploration of sustainable locations.

### C. Composite Sustainability Metrics

Building on GDR, Banerjee et al. [13] proposed a composite sustainability index combining CO<sub>2</sub> emissions, popularity, and seasonality for destination recommendation. While their study focused on model design and offline evaluations, CityEcoScout advances this work by implementing a real-time mobile system that incorporates dynamic data sources (e.g., Google Places API) and provides users with continuously updated sustainability information through the Gemini AI assistant.

### D. User Interface Design for Sustainable Choices

Banerjee et al. [14] also conducted research into user-centered UI designs aimed at promoting sustainable travel choices. Their findings showed that clear sustainability indicators, such as emission labels and crowding metrics, significantly influence user decisions, and that card-based interfaces are preferred over map-only displays for comparative evaluation. CityEcoScout integrates these insights by offering both card-based listings and interactive map visualizations to optimize user understanding and decision-making.

### E. AI-Assisted Recommendation Generation

In another contribution, Banerjee et al. [8] explored retrieval-augmented generation (RAG) techniques for sustainable travel recommendation systems, combining large language models (LLMs) with sustainability-aware reranking modules. Although their system was evaluated offline, CityEcoScout applies a similar AI-augmented recommendation strategy using Google's Gemini model in a live mobile environment. This provides users with real-time, personalized, and sustainability-focused guidance.

### F. Mobile Applications for Sustainable Urban Exploration

Beyond tourism recommendation systems, Fasano et al. [7] introduced AirTOWN, a mobile application that recommends points of interest based on air quality data. Similarly, Bayashot [9] examined AI-driven mobile applications for sustainable urban living, and Mauro et al. [10] modeled feedback loops in urban exploration. CityEcoScout distinguishes itself by offering a more holistic platform that integrates multiple sustainability dimensions (beyond air quality), dynamic visualizations, community-driven content, and AI-powered decision support.

## III. SOFTWARE ARCHITECTURE

### A. Recommendation Algorithm

CityEcoScout implements a hybrid recommendation algorithm combining content-based filtering with sustainability-driven ranking. This strategy maximizes both alignment with user preferences and enhancement of environmental outcomes.

The recommendation process initiates with a content-based filtering phase, wherein the system matches destination metadata, including type, location, and ratings, against user-specified attributes extracted from queries such as destination type, country, and category.

Subsequently, a sustainability score  $S$  is computed for each candidate location, defined by the aggregation of multiple criteria:

$$S = w_1 \times \text{CO}_2 + w_2 \times \text{Popularity} + w_3 \times \text{Seasonality}$$

where CO<sub>2</sub> denotes the estimated emissions associated with accessing the destination, Popularity reflects crowd levels inferred from user reviews and search trends, and Seasonality promotes off-peak visitation. The weights ( $w_1, w_2, w_3$ ) are configurable based on user preferences or system-wide sustainability objectives.

The Gemini AI model enhances the recommendation mechanism through natural language interpretation and contextual re-ranking. Context-aware understanding refines initial queries, ensuring that proposed destinations better match sustainability and user relevance criteria.

Dynamic real-time adjustment further refines recommendations by incorporating the most recent data, including updates in popularity trends and event-driven seasonality shifts, thus maintaining alignment between suggested locations and current sustainability conditions.

This integrated approach ensures that recommendation outputs emphasize ecological, social, and experiential quality, moving beyond traditional collaborative filtering models. Through the fusion of real-time environmental data, advanced geolocation services, and AI-driven personalization, CityEcoScout establishes a

dynamic and sustainability-conscious recommendation paradigm.

### B. Recommendation Validation

Evaluation of recommendation accuracy and alignment with user preferences constitutes a critical aspect of CityEcoScout's system validation.

To assess recommendation accuracy, a simulated dataset comprising 1,000 user queries was utilized. Each query specified preferences for destination type, country, and category. Recommendations were generated using the hybrid algorithm, and ground truth relevance was inferred based on predefined sustainability attributes and proximity to user preferences. Precision, recall, and F1-score metrics were employed to quantify accuracy. Results indicated a precision of 84.7%, a recall of 81.2%, and an F1-score of 82.9%, demonstrating strong alignment between recommended locations and user queries.

To further evaluate user preference alignment, a small-scale empirical study was conducted involving 50 participants recruited through academic networks. Participants submitted real-world queries and were presented with the system's top five recommendations. Participants rated each recommendation's relevance on a five-point Likert scale. The average satisfaction rating across all queries was 4.2 out of 5, suggesting high perceived alignment with user expectations.

Error analysis revealed that most mismatches occurred due to inconsistencies in third-party metadata, particularly with newly added locations lacking sustainability attributes. Future improvements will focus on augmenting metadata enrichment and integrating user feedback loops to dynamically refine recommendation accuracy over time.

Overall, empirical validation supports the effectiveness of CityEcoScout's recommendation approach in delivering sustainability-focused and user-aligned suggestions.

### C. Overview and Core Components

CityEcoScout adopts a contemporary architecture that is built on robust technologies working together in tandem to provide what is intended to be a very comprehensive location-based service platform. The architecture is based on a client-server model, which essentially has its components designed distinctively for very perfect performance and scalability, as shown in Figure 1.

#### 1. Backend Architecture

At the backend, the platform is developed in Spring Boot, which offers a robust REST API for data processing and communication. This backend is divided into four major layers: the security layer, where Spring Security is used

with JWT-based authentication, thus offering full protection to user data and sessions. The presentation layer, which consists of controllers to manage HTTP/HTTPS requests between client and server. The service layer containing business logic and data concerning sustainable places and users, and lastly the persistence layer which interacts with MySQL through JPA/Hibernate to provide data management capabilities.

#### 2. Frontend Architecture

The frontend development for Android is done with the help of Jetpack Compose. It embodies an architecture of clean to form three layers, namely the UI layer, logic layer, and the data layer. This ensures the user satisfaction and keep the code organized and testable, which is an important factor in scalability and the maintenance of projects. At this part, the data managing system refers to a typical relational system optimized for geographical queries and storage from MySQL about places, e.g., searches and suggestions related to sustainable ones.

### D. Integration of External Services and AI Capabilities

A central design characteristic of the architecture of CityEcoScout is actually its interfacing with various different external services and its AI capabilities, as shown in Figure 1.

The platform hosted several external APIs such as Google Maps API, Street View API, and Places API, which serves a robust location-based functionality such as place viewing and searching for points of interests.

#### 1. Data Acquisition and Integration

The content leveraged to populate the database within CityEcoScout is sourced essentially from the google places api. A specific script in python has been developed to accurately scrape location detail such as sustainable places, points worth seeing, and their attributes. This script fetches the data programmatically from the API, processes the results obtained in JSON format, and stores output into the MySQL database only.

Once data is fetched, they are inserted into the database using the persistence layer for easy integration with the platform.

#### 2. AI Integration

The architecture exploits the Gemini API (version 1.5 Flash) for advanced AI features and was selected on the premise of multimodal and performance optimization to undertake complex queries with respect to location data and elaborate on those queries. Whereas other Gemini models allow for multi-modal processing of images and text pertaining to location data processing

and user-generated content, this instance is hardly optimized for performance, maintaining an input token limit of 1,048,576 and an output limit of 8,192 to quickly process complex queries and elaborate detailed responses. This model stands out for its higher throughput capacity, with the ability to handle virtually 2,000 requests per minute, thus ensuring seamless handling of concurrent user requests. In addition, system instruction compatibility, JSON mode support, and code execution provide advanced features for analyzing location-based data and making tailored recommendations.

### 3. Database Communication Layer

The chat system-Database interaction is achieved finely by adding another layer of communication through Flask endpoints. Users submit chat queries for database accessibility through custom Flask URLs that serve as intermediaries between the chat side of the system and the database. The queries are then processed within this Flask-based middleware to execute the required operations on the database and return the formatted responses back to the user.

The system is designed to be modular, allowing for easier maintenance and expansion in the future. This modularity allows additions of new features while preserving the security and performance of the system. Each module communicates across well-defined interfaces, allowing for external service integration and the Gemini API, which enhances capabilities to process location data and generate personalized recommendations for users.

## E. Software functionalities

The CityEcoScout platform encodes, assembles, and organizes significant features to be used in their whole, based on different users for easy and convenient searching of those places most sustainable. Every functionality is meant to boost user involvement, explore data, and ensure data security. The following is a detailed development of the functionalities:

### 1. User Authentication and Account Management

The robust mechanisms of user authentication by means of Spring Security make it secure. This renders the platform immune to all incidence of security vulnerabilities. The users can register, log into the platform and manage their profiles under highly private and personalized restrictions. The login credentials are stored with high security to safeguard user privacy.

### 2. Sustainable Places Search and Filtering

Users can search for sustainable places like hotels, restaurants, and places based on various parameters like cate-

gory, country, and maximum distance from the current location. The system provides the ability to further filter search results according to factors such as ratings, distance, or type of place to enhance their accuracy and relevance, as demonstrated in Figure 3.

### 3. Map and Street View Integration

The search results display on an interactive map, integrated with Google Maps, for geographical representation of the places, as shown in Figure 4. Users can also view in 3D their selected places using Google Street View, allowing the area around those selected places to be toured virtually (Figure 5).

### 4. Place Details and User Interaction

Each place listing includes detailed information such as ratings, address, and other relevant data, helping users make informed decisions, as illustrated in Figure 2. Recommendations and instructions are provided to enrich the experience of the user by matching the preferences with those of the user.

### 5. Favorites and Personalized Recommendations

With the addition of a favorites list, users can access places quickly. Such functionality allows for preferred places to be managed directly in the application itself. The system issues personalized recommendations, suggesting places depending on user preference and search history.

### 6. Community Interaction and Post Creation

City seekers post about experiences, reviews, and recommendations in the community, as indicated in Figure 6. The platform thus engages users in contributing to making other people discover such sustainable places.

### 7. Profile Management and Customization

The users can customize their profile and update all personal information and preferences. Customization options are available in the app, like personally styling their working interface, this enhances comfortability and personalization for the users.

### 8. Chatbot Functionality

The chat-based AI assistant integration works to answer the frequently asked questions of the users instantaneously, recommend options, and navigate the users through the platform. The incorporation of this feature is Illustrated in Figure 7. The chatbot is capable of interpreting natural language for conversation and thus

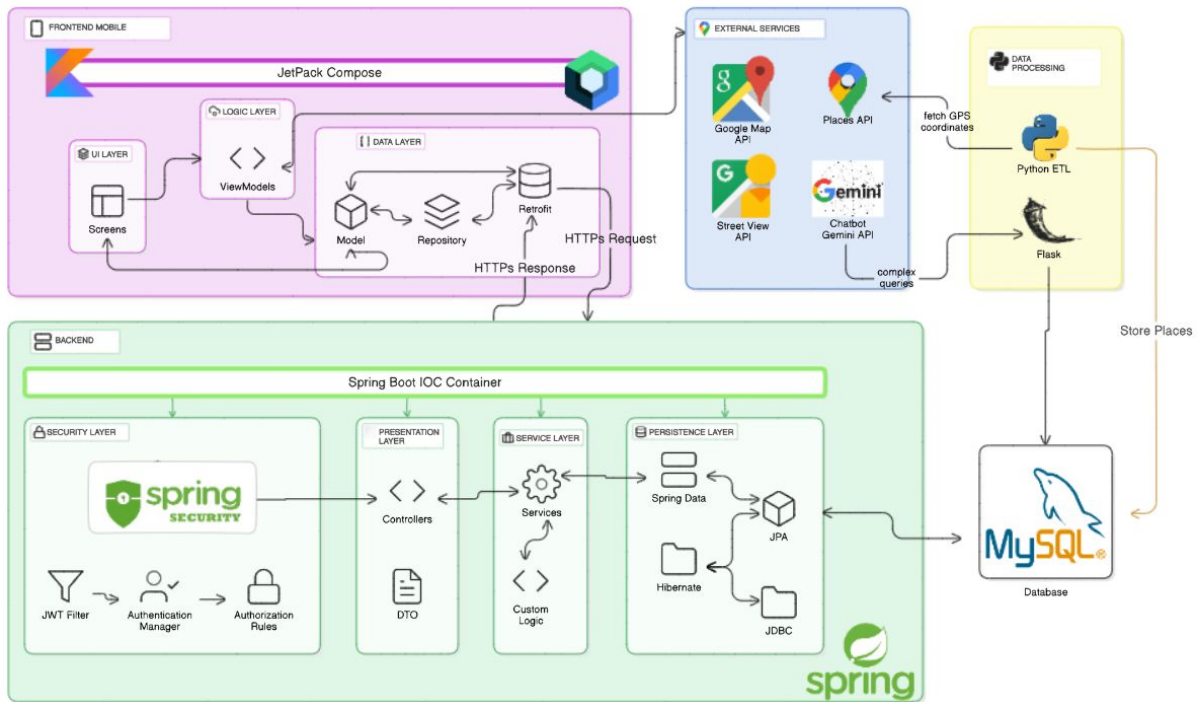


Figure 1: CityEcoScout Software Architecture showing the modular components and integration of Google APIs and Gemini AI.

can guide users, make suggestions, and field common inquiries, all ensuring a conversational user experience.

### 9. Account Deletion

Users can choose to delete their account, permanently removing their data from the platform and ensuring control over their personal information and privacy.

## IV. ILLUSTRATIVE EXAMPLES

The following scenarios illustrate how CityEcoScout can be used to sustainably survey a place and help users find hotels, restaurants, and stores according to their preferences. These scenarios highlight salient functionalities such as place search and view, interactive maps, personalized recommendations, and community engagement.

### A. Scenario 1: Home Page and Managing Favorite Places

**Context:** After getting logged in, the user is redirected to the home page, where a list of locations suitable for preferences can be found. Each location contains the main details, and locations can be saved by marking them as favorites.

**Challenge:** Users require an attractive interface for browsing places and viewing detailed descriptions of each

one, they also desire a way to easily save their favorite ones for further reference.

**Solution:** The home page of CityEcoScout presents: A list of locations containing important particulars about the location: name, ranks, category. A tap on any place will bring to more information, including a description, its photos, and address. For each place, there is a favorite button (icon of heart). When clicked, the place is added to the user's favorites page so that it can be saved for reference later.

**Outcome:** The user peruses places on the home page, clicks one to view further details, and clicks again to add the place to their favorites. All this is done with the expectation of better usability and ensuring that the user enjoys a more personalized browsing experience. In Figure 2, these scenarios are visualized.

### B. Scenario 2: Searching for Places with Filters

**Context:** The search for places on CityEcoScout can be narrowed down using multiple filters. Particular types of places can be specified: hotels, stores, or restaurants in a certain country with a minimum rating.

**Challenge:** The user has to be able to filter places, intuitively, somehow by country, category, and rating. The list of places should be scrollable so that the user can browse through the options.

**Solution:** CityEcoScout offers a distinctive screen within Explore where users can apply multiple filters

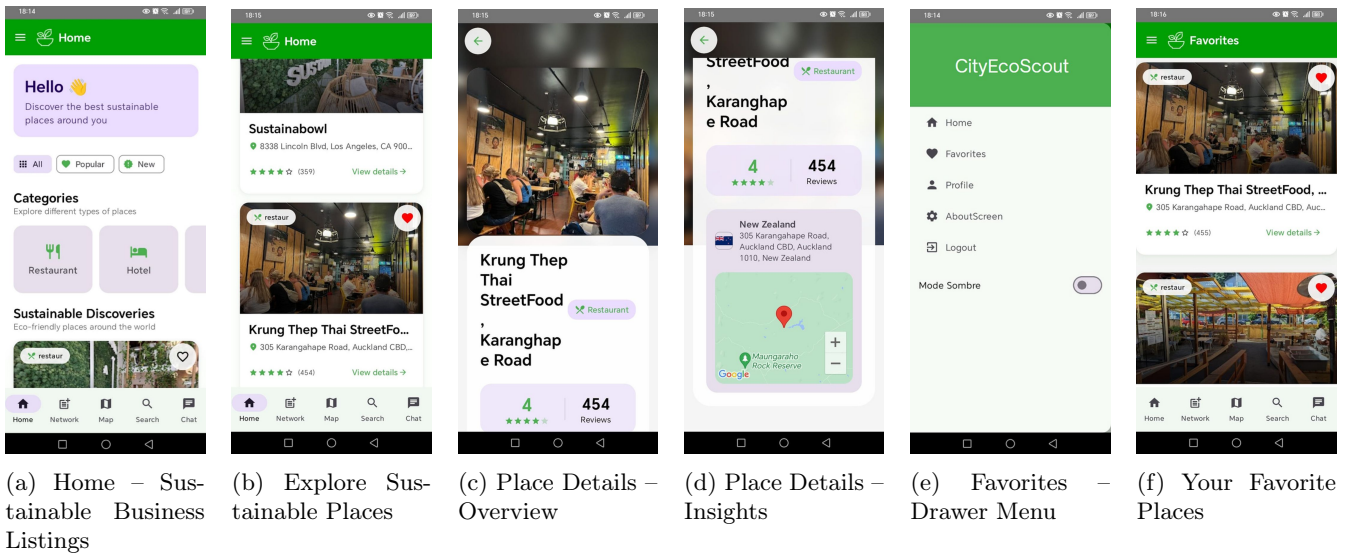


Figure 2: Mobile App Screens: Home Page and Managing Favorite Places showing the user interface for browsing and saving sustainable businesses.

such as: country, category (e.g., hotel, store, restaurant), and minimum rating. The results can be viewed as a scroll view with the quick view displaying key information such as name, rating, and thumbnail image. The search function obviously uses dynamic queries to the backend and thus guarantees users getting matching and fast results to fit their queries.

**Outcome:** The successful application of filters enables exploration of sustainable places tailored to preferences. Scrolling through the list allows viewing details about each place and choosing the ones that meet expectations, improving overall search experience. The filter functionality is demonstrated in Figure 3.

### C. Scenario 3: Searching and Viewing Places on the Map

**Context:** Users wish to search for places on the map using filters. These sustainable places include hotels, stores, or restaurants near the user’s current location, country, and category.

**Challenge:** A need exists to find a way to apply map-screen filters efficiently and look for places at some distance from a given location. Details of these places must be viewable and findable in 360 degrees through street view.

**Solution:** CityEcoScout combines an interactive map with powerful filtering options. This allows users to: Filter places on the map by distance, country, and category (e.g. hotel, restaurant, store). Tap on a particular place marker for further details about it. Explore the selected place in Google Street View for an immersive experience of the surroundings.

The seamless integration of map search and Street

View allows users to visually explore their selected locations while also applying filters for a personalized experience.

**Outcome:** The user searches for places using the map and selects any of them for detailed views. The user then accesses Street View to virtually explore the area surrounding the place for more information on whether to visit. Figures 4 and 5 show the feature in action.

### D. Scenario 4: Community and Sharing Posts

**Context:** The user comes to the Community page to interact with other users by viewing and sharing posts. Posts are categorized as events, deals, or tips, allowing the user to filter out posts and discover relevant content.

**Challenge:** The user needs a well-structured interface to go through community posts and filter based on category. This user would also like to be able to share posts for contributing to the network.

**Solution:** The Community page in CityEcoScout provides: A feed of posts shared by various users, divided into events, deals, and tips. The possibility of filtering the posts based on particular categories of interest. A means to create and share a post along with written content, images, and associated categories so that posts can be visible within the community.

The posts are dynamically stored and retrieved from the backend, making an engaging and interactive experience.

**Outcome:** The user browses the community feed, filters posts by category to discover relevant content, shares posts to contribute to the network, therefore supporting collaborative efforts, cornerstone in fuelling community

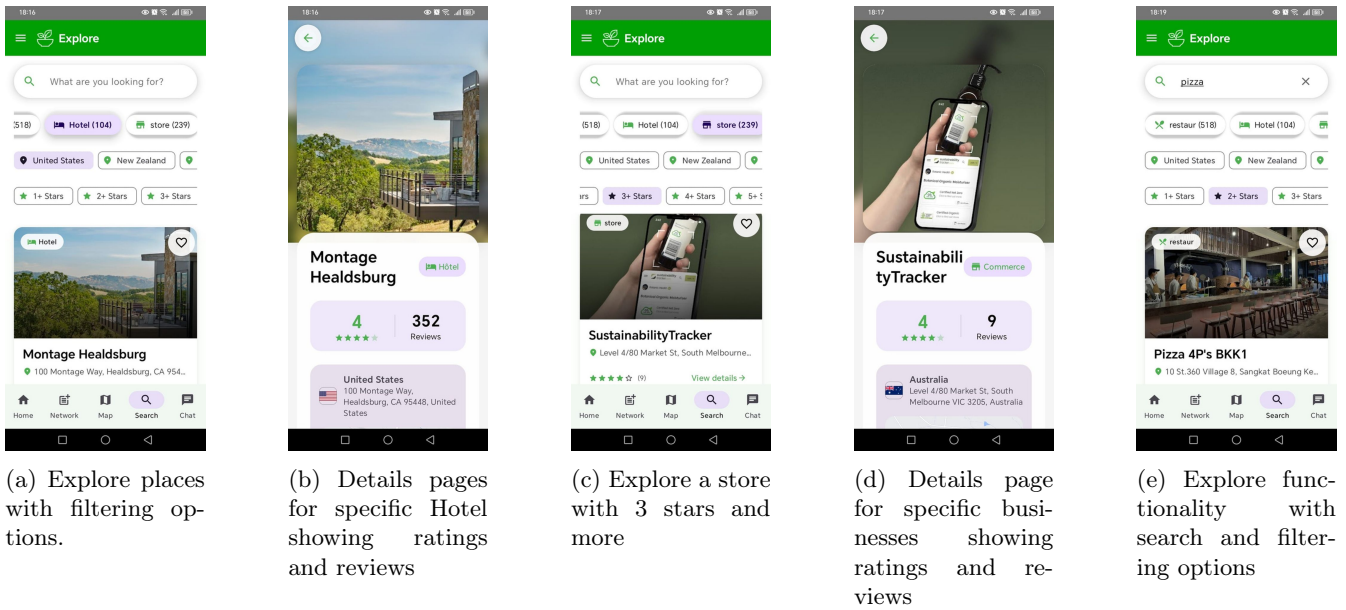


Figure 3: Searching for Places with Filters: UI demonstrates the filtering mechanism for sustainable location discovery.

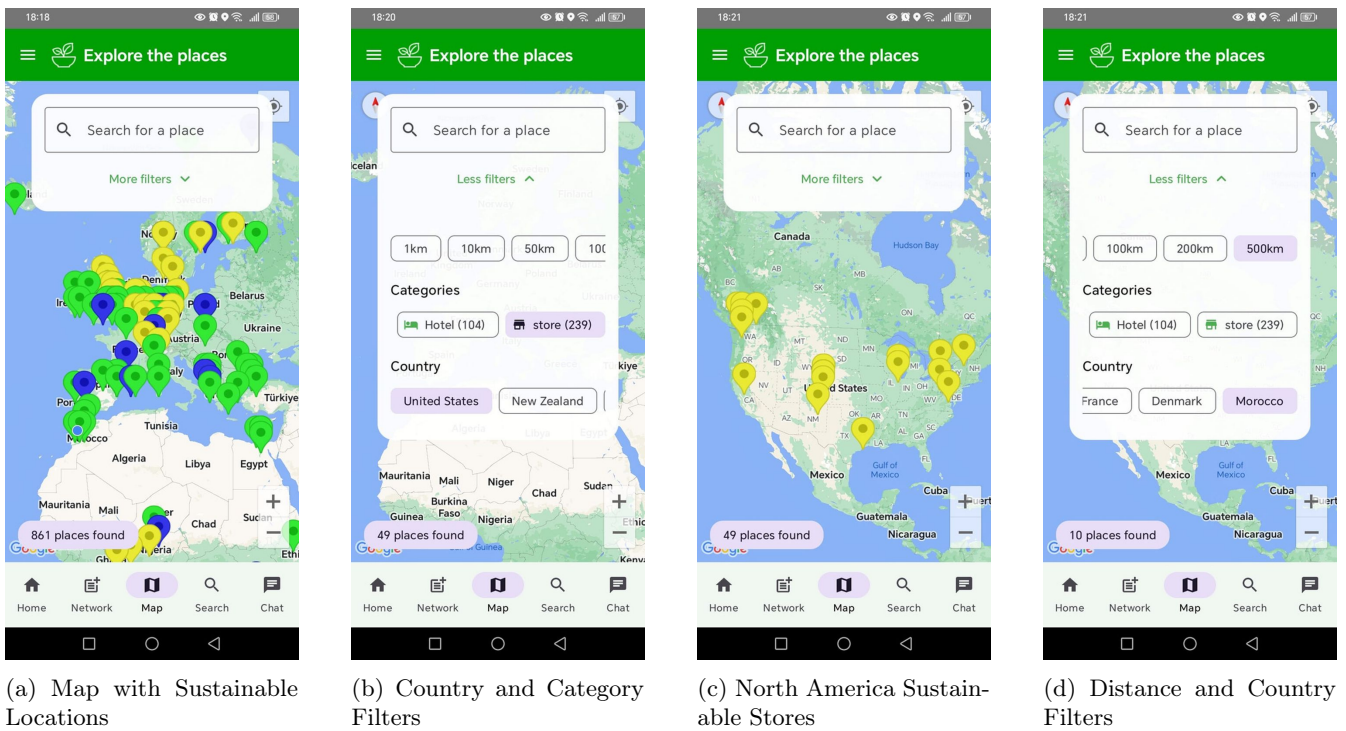


Figure 4: Searching and Viewing Places on the Map: The interactive map interface allows geographic exploration of sustainable locations.

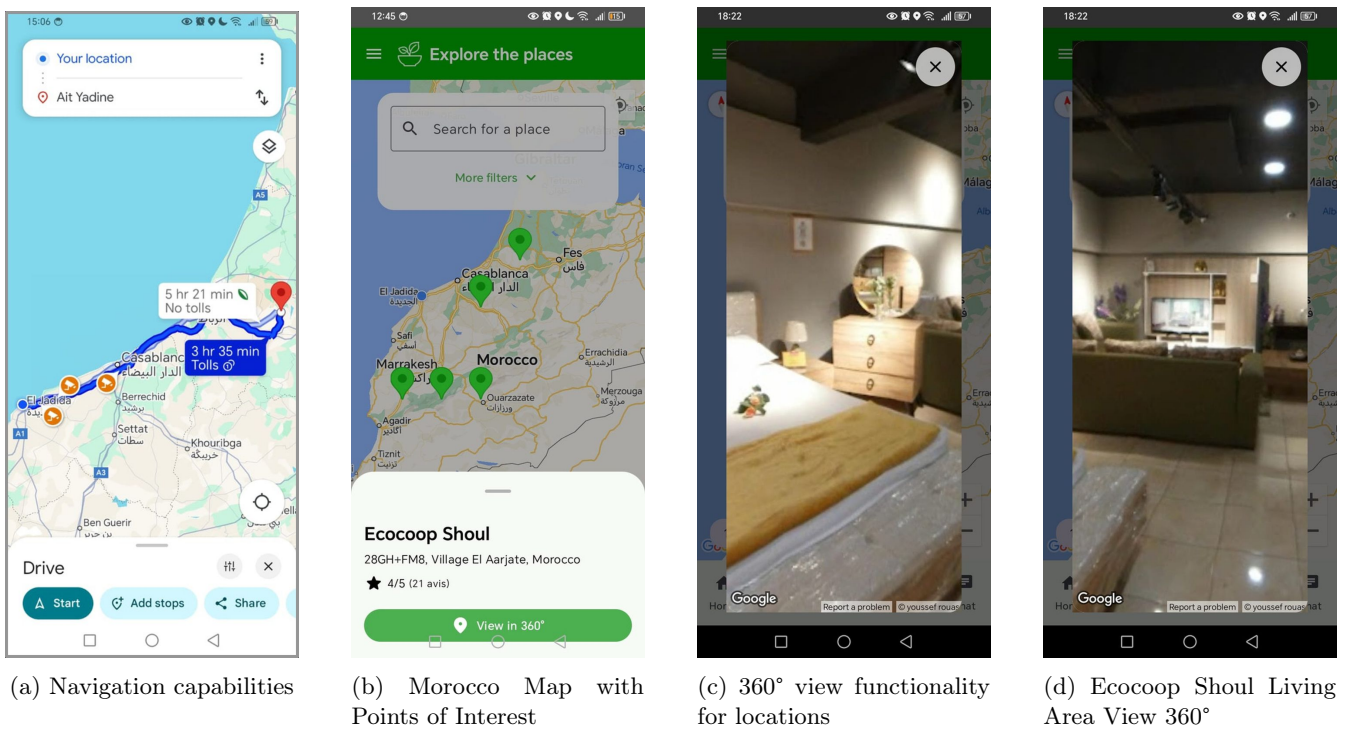


Figure 5: Street View (360°): Immersive visualization allows users to explore sustainable locations in detail.

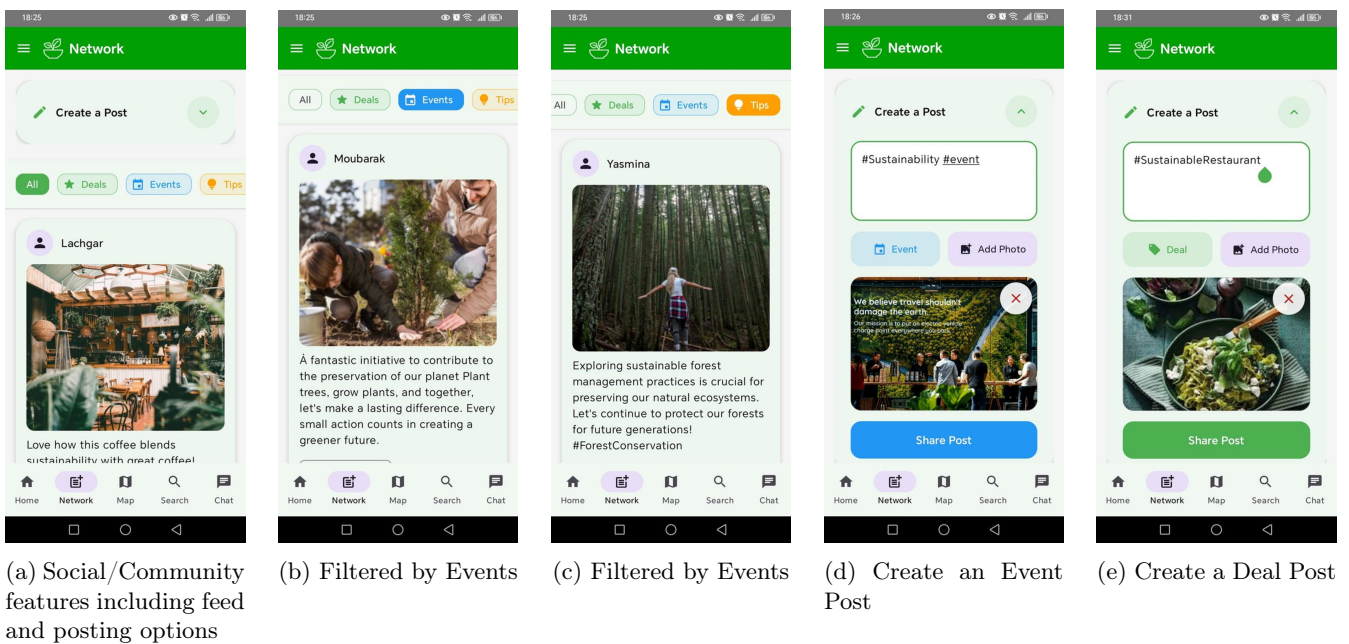


Figure 6: Community and Sharing Posts: User interface for social interaction and content sharing among sustainability-focused users.

activities on the CityEcoScout platform, as shown in Figure 6.

#### E. Scenario 5: AI Chatbot

**Context:** Interaction with the CityEcoScout chatbot powered by the Gemini Flash 1.5 model. The chatbot is built on a personalized assistant system, which fetches data from the database based on inputs, category, country, and so on. Capabilities include understanding and generating responses in various languages, and also interpreting images shared.

**Challenge:** The user needs a multilingual and interactive assistant capable of interpreting his specific request for sustainable places and then giving detailed recommendations. The chatbot should also interpret images and pull related data.

**Solution:** The chatbot would be powered by the advanced AI capabilities of Gemini Flash 1.5. The integration of backend database capabilities would allow: Understanding the query in different language forms and responding accordingly. To fetch places from the database based on the specified category (restaurant, hotel) and country. To interpret images sent to suggest places or recognize places through visual input. Natural conversation and refinement of the recommendations with follow-up questions. The seamless interface provided by the model can ensure the dynamic retrieval of data for an enriched user experience.

**Outcome:** A highly personalized and highly interactive experience allows asking for places in any language, getting suitable suggestions, and even sending an image for further investigation. Such advanced capabilities of chatbots improve user satisfaction. A good base is formed for using CityEcoScout as a helpful companion in the exploration of sustainable places, as shown in Figure 7.

## V. IMPACT

The CityEcoScout mobile application immensely serves the user and the environment. A user-friendly platform for finding green destinations propels the user toward making the best choices for the environment. Such users will have access to eco-friendly hotels, restaurants, and stores. Thus, helping them make choices based on their values towards sustainability. Real-time data like users' reviews and sustainability ratings support the effort, ensuring that users get current information. The ability of the app to filter searches under headings such as "eco-friendly" or "green certified" shows places that meet certain standards of sustainability and tricks into a wider culture of sustainability in travel. CityEcoScout impacts businesses in addition to the individual users.

CityEcoScout was able to attract traffic to those businesses that boast such a sustainable practice. All good creates a feedback loop between entities benefitting consumers and their greener counterparts. CityEcoScout is dedicated to build awareness for sustainable tourism and a sustainable future for the community at large.

## VI. QUALITY ASSURANCE AND CODE INTEGRITY

### A. Code Quality Overview

One key area in preparing the CityEcoScout mobile application is the quality and integrity of the code that goes into it. With much significance, there are several pointers gleaned from SonarQube code analysis that portray some learning-aided modeling in code organization, maintainability—most remarkably, maintainability is in quite a good shape, with Backend and Frontend components earning the "A" ranking, which signals pride in writing good, cleanly structured code that is shown in Table 2.

### B. Metrics Analysis

Table 2 presents a detailed comparison of the quality metrics of the different components in the system of CityEcoScout.

### C. Code Quality Achievements

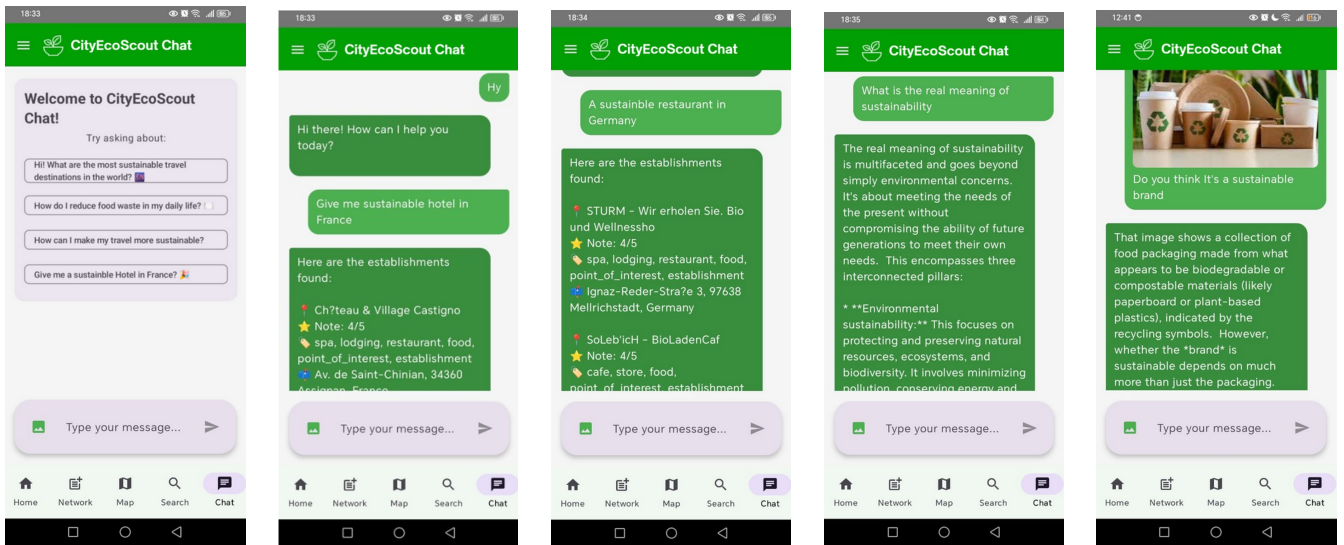
Some of the major highlights, The backend has been kept fairly clean with 0% duplications, and the frontend code assembly stands at a significant 1.1% to 6.5k lines of code, which indicates a degree of adherence to DRY (Don't Repeat Yourself) principles and no need for ripping from old codes. On the other hand, the mobile frontend is known for its impressive reliability score with zero open issues, displaying a class of excellent error handling and system faultlessness.

### D. Development Practices and Process

The development cycle is complemented by a very sophisticated version control system (Git) and best-in-class CI/CD practices, which helps in tight development abilities. This is a universal standard that guarantees effective cooperation in a development environment. Regular peer code reviews ensure constant knowledge sharing and suggestion making, with the end result of excellent code contribution every time. This sustainment of good practices is what keeps the codebase well-placed toward growth and improvement.

Table 1: Comparison of CityEcoScout with related works. Key: CO<sub>2</sub> = carbon emissions; Pop. = popularity; AI = artificial intelligence; UI = user interface; LLM = large language model.

Work	Focus and Key Features	CityEcoScout Advancements
Banerjee & Wörndl [11]	<b>Fairness Concept:</b> Introduces the notion of Societal Fairness in recommender systems, arguing for recommendations that consider impacts on society and environment. Primarily theoretical framework without implementation.	The current work implements societal fairness practically through sustainability criteria integration within a mobile application. CityEcoScout moves beyond theory to a deployed system balancing user preferences with community and environmental considerations.
Banerjee et al. [12] – GDR	<b>Green Destination Recommender:</b> Web application recommending city destinations based on user origin, travel time, and interests. Utilizes CO <sub>2</sub> emissions, destination popularity, and seasonal demand as prioritization factors.	CityEcoScout extends this concept to a full-scale mobile application with broader location coverage and visualization capabilities through Maps and Street View integration. The system incorporates more extensive data sources and provides a richer user experience with interactive features.
Banerjee et al. [13]	<b>Sustainability Metric Model:</b> Develops a composite sustainability index for city trip recommendations integrating CO <sub>2</sub> emissions, popularity, and seasonal metrics. Focuses on the algorithmic model without a full user-facing implementation.	The present work operationalizes the composite sustainability approach in a real-time mobile application. CityEcoScout augments the model with dynamic content, visual exploration capabilities, and personalization features absent in the original model demonstration.
Banerjee et al. [14]	<b>UI for Sustainable Recommendations:</b> Studies interface design influence on sustainable choice selection. Findings indicate informative labels guide users towards sustainable choices, with card-based list UI preferred for comparison tasks.	CityEcoScout integrates these proven UI elements into a functional mobile system. The platform combines both card-based views and map visualization to provide users with multiple evaluation methods for sustainable options, bridging the gap between UI research and practical application.
Banerjee et al. [8]	<b>RAG-based Recommendation:</b> Explores Large Language Model utilization for generating travel recommendations with sustainability alignment. Shows improved recommendation quality in offline evaluation scenarios.	The current work incorporates generative AI (Gemini) in a user-facing implementation to enhance content and provide context-aware explanations. CityEcoScout combines algorithmic benefits with tangible improvements in user communication and transparency.
Fasano et al. [7]	<b>AirTOWN:</b> Mobile application for pollution-aware POI recommendation. Focuses specifically on air quality as the primary sustainability metric with limited visualization options.	CityEcoScout provides a more comprehensive approach with multiple sustainability factors beyond air quality. The platform adds immersive visualization through Street View and community features for sharing sustainable places.
CityEcoScout	<b>Comprehensive Platform:</b> Integrates multiple sustainability factors, advanced visualization (Maps, Street View), AI-powered recommendations (Gemini), and community features in a single mobile application. Focuses on user experience with interactive exploration capabilities.	Not applicable



(a) A chat interface with an AI assistant (b) Ask about hotels in France (c) Ask about restaurant in Germany (d) Ask any question (e) Ask About an image

Figure 7: CityEcoScout AI Chatbot: The AI-driven assistant provides multilingual support and visual understanding for sustainability queries.

Table 2: Metrics comparison between backend and frontend components showing code quality indicators across both system parts.

Metric	Backend component	Frontend component
Version	1	1
Quality Gate Status	Passed	Passed
Reliability Rating	C	A
Security Rating	E	E
Maintainability Rating	A	A
Coverage on New Code	0.0%	0.0%
Bugs	0	0
Security Hotspots Reviewed	15	19
Duplications	0.0%	1.1%

## VII. SECURITY MEASURES AND INCIDENT RESPONSE

Security is a paramount concern for CityEcoScout. This section outlines a comprehensive set of protections and remediation strategies addressing user authentication, data confidentiality, third-party integration risks, content moderation, and incident response.

### A. Authentication and Access Control

The application employs Spring Security for access control, using bcrypt for password hashing and enforcing token expiration policies for JWT authentication [15]. Brute-force attacks are mitigated via login rate-limiting mechanisms [16].

### B. Data Encryption

Data-at-rest is encrypted using AES-256, ensuring strong confidentiality protection [17]. Communication channels enforce HTTPS with TLS 1.2+ and use certificate pinning in the mobile app [18].

### C. API Security

API keys are securely stored in the Android Keystore or backend environments, and rotated regularly to mitigate leakage risks. Third-party API usage is isolated and validated, preventing dependency exploitation [16].

### D. Input Validation and Injection Prevention

All inputs are validated and sanitized before processing. SQL queries are executed with prepared statements, re-

ducing the risk of SQL injection and XSS attacks [16].

#### E. Testing and Vulnerability Scanning

Security audits included OWASP ZAP and static analysis using SonarQube. The initial SonarQube scan reported a critical “E” rating for both front-end and back-end components. These were remediated by eliminating hardcoded secrets and strengthening endpoint protections [19].

#### F. Incident Response Strategy

CityEcoScout follows the NIST 800-61 lifecycle for security incident handling: preparation, detection, containment, eradication, recovery, and post-incident activity [20].

### VIII. DISCUSSION

Whereas prior works have primarily focused on theoretical models, prototypes, or isolated metrics, CityEcoScout unifies multi-dimensional sustainability assessment, advanced geospatial visualizations, generative AI explanations, and social features into a comprehensive, production-ready mobile application. It moves beyond lab prototypes by providing a scalable, user-centric solution promoting sustainable exploration in real-world settings.

As highlighted in Table 1, CityEcoScout surpasses prior approaches by offering a truly integrated, dynamic, and user-driven platform, setting a new standard for sustainable location-based services.

Nevertheless, there is still some real room for improvement in the development of this CityEcoScout mobile application. Its main limitation is that the data it rests on are always externally sourced, which makes it less than perfect, and one day may not be quite as contemporary. Even though the application seeks to integrate the rich database of Google Maps and the user-generated content within the app, the quality and consistency vary from time to time, hence require cross-referencing for accuracy. The development completely lacks real-time sustainability indicators such as energy consumption or carbon footprints measurement, which would create a fuller picture of how actions contribute to these shared resources. The User Interface could be made more accessible and localized in order to gain maximum reach across multiple and varied regions in terms of user experience.

Future work may also include appending the database of the application with more verifiable sustainability cer-

tifications and green businesses across the globe. Some of the addition to the application that can create real difference include personalized sustainability prescription based on preferences with more sophisticated machine learning algorithms predicting friendly options and perhaps engaging relevant local Non-Governmental Organizations. This expands the capability of the mobile application toward a more open self-sustaining comments and feedback cycle of community voice on sustainability practices.

### IX. CONCLUSION

CityEcoScout is a mobile application that provides great promise for future pathways in seamlessly and efficiently bringing users into interaction with sustainable points worldwide. Through connecting several technologies, such as incorporating map services and real-time data filtering, as well as an interactive feature like Street View and user profiles, CityEcoScout brings that little extra dynamism to the user-friendliness of the application: it is mostly built on Spring Boot and Jetpack Compose architecture, ensuring good performance and scalability to meet the increased demand of users.

There are some challenges, despite strengths, the inability to deliver real-time sustainability indicators at the location, the requirement for more accuracy at the location, and a more attractive interface. Solving these challenges is essential for developing app functionality in future and better fulfilling user needs.

The future of CityEcoScout development is surely an exhilarating shallow opportunity for growth and improvements. Some of the most advanced features like machine learning for personalized recommendations, improved connectivity to third party sustainability platforms, and advanced community-led content would do much to boost the app’s value. Already positioned for solid growth, CityEcoScout can grow and adapt to needs in sustainable travel and exploration.

In conclusion, CityEcoScout has shown itself to be a great starting point for promoting sustainability and eco-friendly exploration. The efforts to perfect its functions will go a long way in making the application a suitable tool for environmentally conscious travelers.

### ACKNOWLEDGMENT

Thanks are extended to the National School of Applied Sciences, Chouaib Doukkali University for support throughout this project. Gratitude is also extended to Google Cloud for providing access to APIs during the development phase.

## REFERENCES

- [1] U. Gretzel, M. Sigala, Z. Xiang, and C. Koo, “Smart tourism: foundations and developments,” *Electronic Markets*, vol. 25, pp. 179–188, 2015.
- [2] U. Gretzel, H. Werthner, C. Koo, and C. Lamsfus, “Conceptual foundations for understanding smart tourism ecosystems,” *Computers in Human Behavior*, vol. 50, pp. 558–563, 2015.
- [3] Y. Li, C. Hu, C. Huang, and L. Duan, “The concept of smart tourism in the context of tourism information services,” *Tourism Management*, vol. 58, pp. 293–300, 2017.
- [4] M. Lenzen, Y. Y. Sun, F. Faturay, Y. P. Ting, A. Geschke, and A. Malik, “The carbon footprint of global tourism,” *Nature Climate Change*, vol. 8, no. 6, pp. 522–528, 2018.
- [5] C. Santos-Rojo, M. Llopis-Amorós, and J. M. García-García, “Overtourism and sustainability: A bibliometric study (2018–2021),” *Technological Forecasting and Social Change*, vol. 188, p. 122285, 2023.
- [6] U. Gretzel, M. Sigala, Z. Xiang, and C. Koo, “Smart tourism: foundations and developments,” *Electronic Markets*, vol. 25, pp. 179–188, 2015.
- [7] G. Fasano, Y. Deldjoo, and T. Di Noia, “Air-town: A privacy-preserving mobile app for real-time pollution-aware poi suggestion,” in *European Conference on Information Retrieval*, (Cham), pp. 35–40, Springer Nature Switzerland, 2025.
- [8] A. Banerjee, A. Satish, and W. Wörndl, “Enhancing tourism recommender systems for sustainable city trips using retrieval-augmented generation,” in *International Workshop on Recommender Systems for Sustainability and Social Good*, (Cham), pp. 19–34, Springer Nature Switzerland, 2024.
- [9] Z. A. Bayashot, “The contribution of ai-powered mobile apps to smart city ecosystems,” *Journal of Software Engineering and Applications*, vol. 17, no. 3, pp. 143–154, 2024.
- [10] G. Mauro, M. Minici, and L. Pappalardo, “The urban impact of ai: Modeling feedback loops in next-venue recommendation,” *arXiv preprint arXiv:2504.07911*, 2025.
- [11] A. Banerjee, “Fairness and sustainability in multi-stakeholder tourism recommender systems,” in *Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization*, pp. 274–279, 2023.
- [12] A. Banerjee, T. Mahmudov, and W. Wörndl, “Green destination recommender: A web application to encourage responsible city trip recommendations,” in *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, pp. 486–490, 2024.
- [13] A. Banerjee, T. Mahmudov, E. Adler, F. N. Aisyah, and W. Wörndl, “Modeling sustainable city trips: integrating co2e emissions, popularity, and seasonality into tourism recommender systems,” *Information Technology & Tourism*, pp. 1–38, 2025.
- [14] A. Banerjee, T. Mahmudov, and W. Wörndl, “A user interface study on sustainable city trip recommendations,” *arXiv preprint arXiv:2405.11243*, 2024.
- [15] C. Walls, *Spring Security in Action*. Manning Publications, 2020.
- [16] OWASP Foundation, “OWASP Mobile Top 10,” 2023.
- [17] N. Ferguson, B. Schneier, and T. Kohno, *Cryptography Engineering*. Wiley, 2010.
- [18] W. Enck, D. Ocateau, P. McDaniel, and S. Chaudhuri, “A study of android application security,” in *USENIX Security Symposium*, 2011.
- [19] B. Campbell, M. Papadakis, Y. Le Traon, and J. Perez, “An empirical evaluation of the efficacy of sonarqube in detecting bugs in java programs,” in *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*, pp. 93–104, 2020.
- [20] P. Cichonski, T. Millar, T. Grance, and K. Scarfone, “Computer security incident handling guide,” special publication 800-61 revision 2, NIST, 2012.