Tribhuvan University

Faculty of Humanities and Social Sciences

D.A.V. College

SUPERVISOR'S RECOMMENDATION

I hereby recommend that this project prepared under my supervision by Irish Prajapati entitled **"Peak Times"** in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

Santosh Rauniyar

SUPERVISOR

Lecturer

The Department of Bachelors in Computer Application

Jawalakhel, Lalitpur

Tribhuvan University

Faculty of Humanities and Social Sciences

D.A.V. College

LETTER OF APPROVAL

This is to certify that this project prepared by IRISH PRAJAPATI entitled **“Peak Times”** in partial fulfilment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

Santosh Rauniyar, Lecturer Department of Bachelors in Computer Application Jawalakhel, Lalitpur	Mr. Sudip Adhikari, Head of Department Department of Bachelors in Computer Application Jawalakhel, Lalitpur
Internal Examiner	**External Examiner** Tribhuvan University

ABSTRACT

Tourism in Nepal is growing with rising numbers of domestic and international tourists arriving. However, a common problem for tourists especially first-time travelers is having no idea what to see or what kind of crowd there will be. In this context, this study suggests Peak Times, an online travel guide that can provide real-time and location-based crowd level to travelers.

The system relies on machine learning algorithms to predict crowd density in various tourist spots based on contextual factors such as time, place, weather, and day type. Users can register, log in, allow or refuse GPS permission, or enter location manually to receive the recommendations. The project displays a list of appropriate locations along with predicted crowd density and simple-to-grow visual indicators for improved travel planning and experience.

The underlying predictive model was XGBoost Regressor, and it represented the best precision (R^2 score of approximately 0.88) and is therefore chosen as the primary algorithm. Random Forest Regressor have also been used as ensemble models for crowd behavior prediction stability and nuanced pattern identification. These models have been trained on simulated as well as actual data to guarantee performance as well as generalization.

Keywords: *Crowd Prediction, XGBoost, Tourism Assistant, Nepal, Smart Travel*

ACKNOWLEDGEMENT

I am very grateful to the department of computer application, D.A.V college for providing me an opportunity to work on a project as part of Project ‘II’. I am delighted to express my deep sense of gratitude and gratefulness to our academic lecturer **Santosh Rauniyar** for his invaluable guidance, encouragement, and even monitoring to spare time despite his busy schedule for project progress reviews.

Furthermore, I would like to thank **Mr. Sudip Adhikari** for his guidance and motivation throughout the project. In addition, I would like to thank my friends, who have willingly shared their precious time, advice, and encouragement during planning of this project.

My special thanks go to our college department and everyone who directly and indirectly extended their hands in making this project successful.

Sincerely,

Irish Prajapati

[6-2-469-5-2021]

Table of Contents

LIST OF ABBREVIATIONS

API	Application Program Interface
API	Application Programming Interface
CRUD	Create, Read, Update, Delete
DBMS	Database Management System
DFD	Data Flow Diagram
ER	Entity-Relationship
GPS	Global Positioning System
RTDB	Real-Time Database
UAT	User Acceptance Testing
UI	User Interface
XGBoost	Extreme Gradient Boosting

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Introduction

Tourism in Nepal is experiencing rapid growth, attracting people from across the globe to explore its breathtaking landscapes, rich culture, and vibrant cities[1]. However, a common challenge that travellers face especially first-time visitors is uncertainty about where to go, which places are worth visiting and the potential crowd levels at those locations. This is where Peak Times comes in.

Peak Times is a smart, web-based travel assistant developed to make travel within Nepal more informed, engaging, and convenient. The platform is designed to help both domestic and international tourists discover the best destinations tailored to their interests and current location. Whether you're an adventurer, researcher or traveller with new experiences the peak times helps to find the crowd level at the specific time of the day according to the weather and distance from your manual location or accessed location.

When a user visits the website, they are greeted with a clean and user-friendly interface where they can register and log in to access personalized features. After logging in, the system requests location access. If the user grants permission, their current location is automatically detected and displayed on a map. If the user denies the request, they can manually input their current location through a simple form limited to Kathmandu, Bhaktapur and Lalitpur.

Once the location is set, users can begin exploring. They have two options: they can either search by entering the name of a district (such as Kathmandu, Lalitpur, or Bhaktapur) or by selecting a category (like nature, religious sites, nightlife, adventure, etc.). The application then displays a list of places within that district or category using dummy data for every tourist spot in these selected district latitude and longitude. These places are presented along with estimated crowd levels, which are visualized through intuitive bar(which shows in formation of 3 High crowded places, 2 medium crowded and 2 less crowded places).

1.2 Problem Statement

Nepal, renowned for its rich cultural heritage, majestic landscapes, and spiritual significance, attracts a significant number of tourists annually. However, despite its considerable tourism potential, the industry encounters substantial challenges in the efficient management of tourist destinations. A particularly pressing issue is the phenomenon of over-tourism at well-known sites, contrasted with the underutilization of equally captivating yet less frequented locations. This imbalance not only strains local infrastructure but also contributes to environmental degradation and diminishes the overall visitor experience [2].

Currently, Nepal lacks a centralized digital system capable of tracking, predicting, and displaying tourist flow in real time. This deficiency hinders the ability of both tourists and tourism authorities to make well-informed decisions. Visitors frequently find themselves in overcrowded areas such as Thamel or SwayambhuNath without awareness of nearby, tranquil alternatives that align better with their interests. Simultaneously, authorities and local businesses struggle to anticipate crowd levels and allocate resources effectively, resulting in staff shortages in busy areas and underutilized infrastructure in quieter regions.

Another significant shortcoming of the current tourism ecosystem is the absence of personalized recommendations. The majority of existing platforms offer generic information that fails to consider a user's specific preferences, interests, or real-time crowd data. Tourists are not equipped with the tools to explore destinations based on their individual travel styles be it nightlife, cultural landmarks, or natural scenery. This can lead to dissatisfaction and a missed opportunity for a more profound engagement with the country's lesser-known attractions [3]. The development of web-based travel recommendation systems has been explored as a potential solution to offer more tailored experiences.

Moreover, the absence of real-time data integration leaves local authorities, tour operators, and hospitality providers without crucial information for future planning. Without predictive insights into tourist behaviour and flow, they struggle to forecast demand, manage peak seasons, and achieve a more balanced distribution of tourism across different regions. This ultimately negatively impacts both the visitor experience and the long-term sustainability of the industry. The importance of usability in such systems, ensuring ease of access to relevant information, cannot be overstated. While the availability of online tourist information is growing in developing countries, the need for real-time, personalized systems remains critical. Technologies like geolocation API's and interactive maps offer potential avenues for enhancing the functionality and user-friendliness of future tourism platforms in Nepal.

1.3 Objective

- To develop a location-aware location web application that assist tourist and local discover destinations across cities.
- To integrate crowd data to help users avoid congested areas and plan visits more efficiently.

1.4 Scope and limitations

Scopes

- Provides estimated crowd levels based on time, weather, and location.
- Supports category-wise exploration: nature, religious sites, adventure, etc.
- Covers key tourist districts: Kathmandu, Lalitpur, and Bhaktapur.
- User-friendly interface for smooth travel planning experience.
- Displays 7 recommended places with clear crowd-level indicators.

Limitations

- Crowd levels and other stats are dummy, not fetched from live sensors/APIs.
- Location coverage is currently limited to Kathmandu, Lalitpur, and Bhaktapur.
- Does not support multilingual content, limiting accessibility for non-English speakers.
- Currently based on web-application not available for mobile application

1.5 Development Methodology

The Incremental Model is ideal for developing the "Peak Times" application as it allows for step-by-step implementation, testing, and enhancement of core features. The project can begin with user registration and login, followed by location access and map integration. Afterward, the district and category search functionality can be added, along with crowd-level visualization and content-based filtering for recommendations. This approach ensures early validation of each feature, reducing errors and providing flexibility for improvements. As each increment builds on the previous one, debugging and optimizations become more manageable, leading to a smoother and more efficient development process.

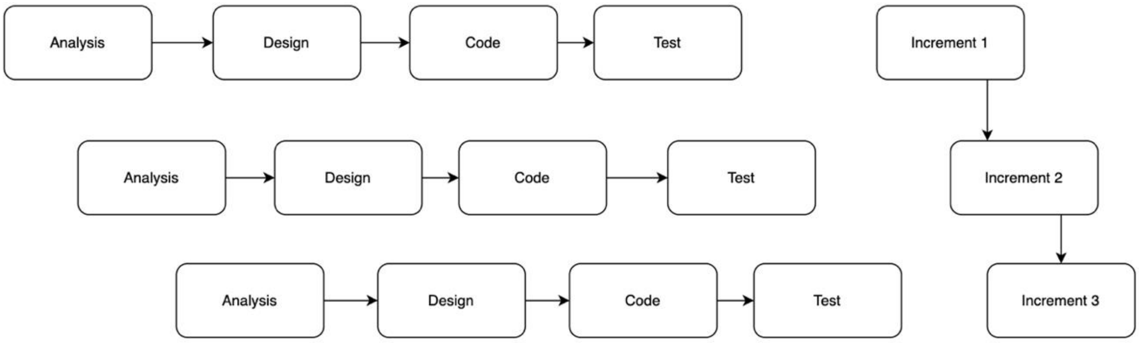


Figure 1.1 Incremental Development of Model for Peak Times

1.6 Report Organization

The report on “Peak Times” is organized into five chapters:

Chapter 1:

This chapter introduces the core objective and motivation behind the development of the "Peak Times" web application. It begins by outlining the current state of tourism in Nepal, which is rapidly growing yet faces challenges such as overcrowded destinations, lack of real-time information, and difficulty in accessing lesser-known but worthwhile locations.

Chapter 2:

This chapter reviews existing travel and location-based systems like TripAdvisor, Google Maps, and MyHolidayNepal. It highlights their features, strengths, and limitations such as lack of real-time crowd data and limited personalization. The chapter identifies gaps in current solutions and explains how "Peak Times" aims to address these issues by offering personalized, crowd-aware travel recommendations tailored to Nepal.

Chapter 3:

This chapter explains the system architecture, requirements, feasibility, and design of the "Peak Times" web application. It follows a three-tier architecture using Django for the backend, PostgreSQL for the database, and Leaflet.js for interactive maps. Includes the system’s ER diagram, DFDs, architectural design, flowchart, and database schema. Lastly, the algorithm used for recommending similar tourist places is described in detail, based on tag similarity and user preferences.

Chapter 4:

This chapter explains the implementation, Testing and debugging part of the system.

Chapter 5:

This chapter provides brief explanation about Conclusion, Limitations, and Future Enhancement of the system.

CHAPTER 2 : BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

Tourism applications and location-based services have gained significant popularity, assisting users in discovering new places, planning their trips, and navigating unfamiliar areas. However, most existing systems often rely on static data and lack real-time crowd-level insights or personalized recommendations. Applications like Google Maps and TripAdvisor provide basic location data and user reviews but typically do not integrate live crowd-level information or offer suggestions tailored to user preferences. Several studies and projects have explored tourism recommendation systems, but they generally focus on one-dimensional aspects, such as user reviews, location-based searches, or popular tourist destinations. For example, systems like TripAdvisor and Yelp allow users to search for popular destinations based on ratings but often do not consider the current crowd level, time of day, or specific user interests, such as nightlife or nature spots.

Furthermore, existing systems rarely adapt to the dynamic nature of urban tourism, where the crowd density and popularity of certain places fluctuate throughout the day. Some studies have proposed integrating real-time data to recommend less crowded places or dynamically adjust suggestions based on the time of day, but these systems are often limited by the availability and accuracy of real-time data.

This project, "Peak Times," aims to bridge this gap by integrating real-time crowd-level information and providing personalized recommendations based on district and category searches. It also introduces a unique feature of allowing users to manually input their location or share their GPS location, offering flexibility and convenience. By focusing on popular tourist areas in Kathmandu, where crowd levels can vary significantly, Peak Times seeks to provide a more nuanced and user-centric travel planning experience.

2.2 Literature Review

This section briefly presents the literature review of systems related to location-based travel recommendation platforms. The first study focuses on TripAdvisor, one of the most popular travel platforms, and how it provides user-generated reviews and ratings to recommend tourist destinations. Several analyses have examined TripAdvisor's user review system and its impact on tourist choices ([Cite a study analysing TripAdvisor's reviews]). The second study analyses the working mechanism of Google Maps in providing location suggestions and navigation ([Cite a study analysing Google Maps for travel]), but highlights its lack of personalized crowd-level insights and user recommendations. The third study reviews the Foursquare application, which categorizes locations based on user preferences and check-ins but lacks real-time crowd data integration. The fourth study examines Nepal's own travel app, MyHolidayNepal, which offers general travel guides and destination lists, but does not offer dynamic filtering or real-time updates. These studies help identify key gaps in current systems, such as limited personalization, lack of real-time crowd insights, and minimal local data integration, which "Peak Times" aims to address.

a) TripAdvisor

TripAdvisor is one of the world's leading travel platforms, primarily leveraging user-generated content to recommend destinations, hotels, restaurants, and activities. It builds trust through millions of reviews and ratings submitted by travellers worldwide.

Key Features:

- User Reviews and Ratings**:** Offers detailed, crowd-sourced feedback on tourist locations, accommodations, and services, helping users make informed decisions.
- Ranking and Filtering**:** Provides ranked lists of attractions based on popularity, price, traveller rating, and type of experience.
- Travel Forums**:** Hosts active community discussions where users can ask questions, share travel tips, and seek advice.
- Personalized Suggestions**:** Uses browsing history and previous activity to suggest destinations and activities aligned with user preferences.
- Booking Integration**:** Allows users to book hotels, flights, tours, and activities directly from the platform.

b) Google Maps

Google Maps is a comprehensive navigation platform integrating GPS tracking with artificial intelligence to deliver accurate, real-time routing and traffic analysis. Drawing data from a vast network of users and official transport sources, it offers a robust solution for both everyday commuting and travel planning.

Key Features:

- Real-Time Traffic Updates**:** Monitors traffic conditions using location data from millions of devices, identifying slowdowns, accidents, and road closures.
- Turn-by-Turn Navigation**:** Provides voice-guided navigation customized for different modes of travel, including driving, biking, walking, and public transit.
- Street View and Indoor Maps**:** Enables visual exploration of streets and interior layouts of venues, improving situational awareness before visits.
- Offline Maps**:** Allows users to download regional maps for navigation without internet access, vital for remote travel.
- Multimodal Transit Planning**:** Integrates public transport, walking, and ride-sharing into optimized routes for time and cost efficiency.

c) MyHolidayNepal

MyHolidayNepal is a Nepal-focused travel application offering curated destination information and general travel guides. While valuable for domestic tourism promotion, it lacks dynamic and real-time features.

Key Features:

- Destination Listings**:** Provides comprehensive lists of tourist attractions, hotels, and activities within Nepal.
- Travel Blogs and Articles**:** Features stories and tips from local travellers to inspire trip planning.
- Static Recommendations**:** Suggestions are generally fixed and not updated based on live crowd conditions or events.
- Cultural and Event Highlights**:** Shares information about festivals, local events, and cultural practices.

- Offline Information Access**:** Allows users to browse essential travel content without needing constant internet connectivity.

CHAPTER 3: SYSTEM DESIGN AND ANALYSIS

3.1 System Analysis

The system design of Peak Times follows a three-tier architecture consisting of the presentation layer, logic layer, and data layer. The presentation layer is built using Django's templating system, with files dynamically rendered based on user interactions. This allows a smooth and structured user experience directly integrated with backend logic. The core application logic is handled by Django views and models, written in Python, which manage the routing, data processing, and coordination between the frontend and database. PostgreSQL, managed through PostgreSQL Admin, serves as the backend database, securely storing information about locations, categories, and peak times. The system is designed to let users choose a district and category (e.g., religious sites, food spots), and then view peak hours visually through embedded graphs and interactive maps. The modular structure ensures scalability and ease of maintenance, while the use of Django's MVC pattern helps in maintaining separation of concerns and facilitating future enhancements, such as real-time data integration and user feedback feature

3.1.1 Requirement Analysis

i. Functional Requirements

- User can create account to manage their profiles.
- System handles allow location and manual input location by user.
- User can search by district and tags(categories).
- Markers are displayed on the map basis on search input.
- User can view the route from current location to searched locations.
- Users can add the missing place with respective co-ordinates.
- User can review the place and give certain feedbacks.

The Figure 3.1 Use Case Diagram shows how user can interact with Peak Times:

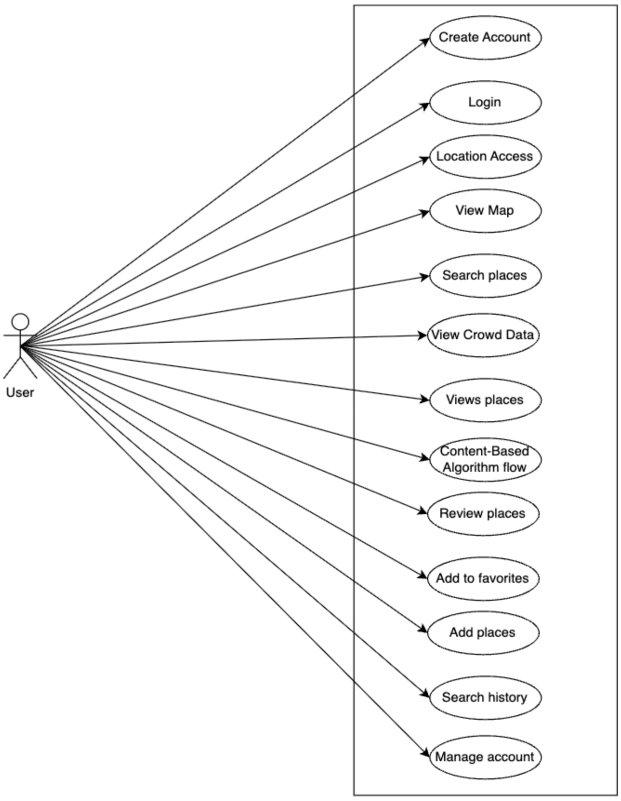


Figure 3.1 Use Case Diagram Showing User Interactions for Peak Times

The Use Case Diagram of Peak Times outlines the interactions between different users and the functionalities (use cases) available in the system. It highlights how both general users interact with various modules of the web application. From the start of application what are the tasks that user can perform in this web-application

The Figure 3.2 explains how the Admin can interact with Peak Times

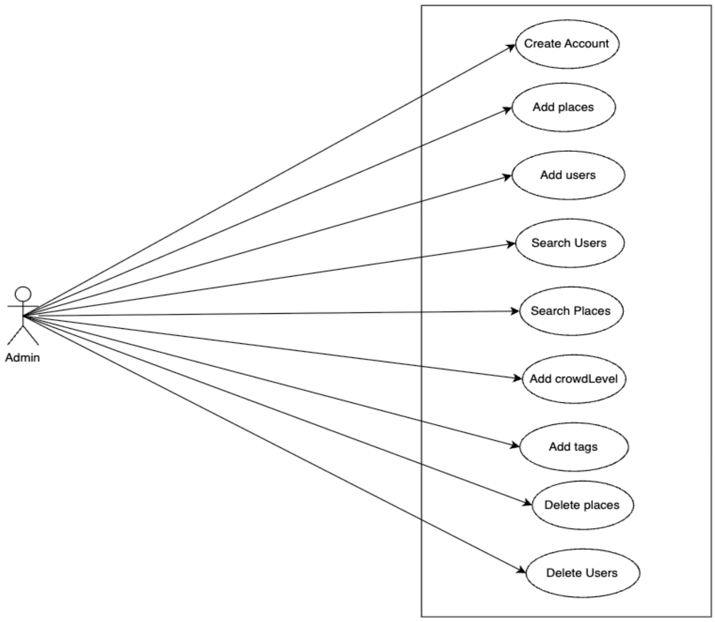


Figure 3.2 Use Case Diagram Showing Admin Interactions for Peak Times

The Use Case Diagram of Peak Times outlines the interactions between different admins and the functionalities (use cases) available in the system. It highlights how general admin can interact with various modules of the web application featured according to different modules

ii. Non – Functional Requirements

- User friendly Interface

  The system is designed with a clean and intuitive interface, allowing users to easily navigate through different features. This improves overall user satisfaction and reduces the learning curve.

- Usability

  The application ensures smooth interaction by minimizing complexity and offering helpful features like real-time suggestions and organized layouts. This makes it efficient and effective for users to achieve their goals.

- Scalability

  The system is built to handle increasing amounts of users and data without performance loss. This ensures the application remains reliable as it grows in popularity and usage.

- Security

  User data is protected through authentication and access control. The system also ensures that unauthorized access to sensitive information is prevented.

3.1.2 Feasibility Study

i. Technical Feasibility

   The project is developed using Django, a high-level Python web framework known for its rapid development capabilities and clean design. Django's powerful template engine is utilized to build dynamic and responsive HTML and JS pages that display peak time traffic insights. PostgreSQL, accessed through Postgres admin, serves as the project's primary database system, offering reliable performance, strong data integrity, and advanced query capabilities—essential for handling structured peak traffic data across locations and timeframes. This stack ensures a robust, secure, and scalable web-based application that supports efficient traffic pattern analysis and visualization.

ii. Operational Feasibility

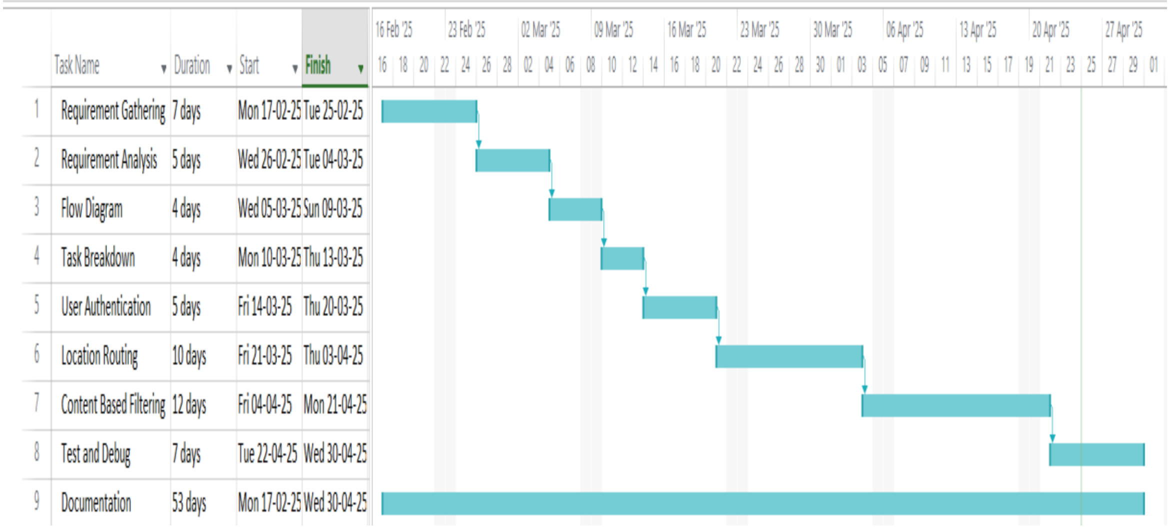
   The proposed system is operationally feasible as it offers an intuitive and user-friendly interface, making it easy for commuters to report road obstacles and access real-time route updates without requiring technical expertise. The use of Leaflet.js ensures interactive and lightweight map integration, suitable even for users with average internet connectivity. Django's backend and PostgreSQL provide a stable and secure environment for managing and retrieving data, ensuring smooth operation under typical user loads. Since the technologies used are open-source and well-documented, system maintenance and future updates can be handled efficiently. Additionally, implementing a basic moderation system for user-reported data increases the platform’s reliability without requiring large-scale administrative oversight.

iii. Economic Feasibility

The Peak Times system is economically feasible due to its reliance on cost-effective, open-source technologies. The backend is built using **Django**, a powerful Python framework that accelerates development and reduces long-term maintenance costs through its robust ecosystem and built-in functionalities. **Leaflet.js** offers a lightweight and free alternative to proprietary mapping APIs, eliminating the need for costly licensing or usage-based fees, which is especially beneficial for a traffic-heavy navigation system.

iv. Schedule Feasibility

The time schedule for different modules of this project can be seen as below in the following Gantt chart.



	Task Name	Duration	Start	Finish
1	Requirement Gathering	7 days	Mon 17-02-25	Tue 25-02-25
2	Requirement Analysis	5 days	Wed 26-02-25	Tue 04-03-25
3	Flow Diagram	4 days	Wed 05-03-25	Sun 09-03-25
4	Task Breakdown	4 days	Mon 10-03-25	Thu 13-03-25
5	User Authentication	5 days	Fri 14-03-25	Thu 20-03-25
6	Location Routing	10 days	Fri 21-03-25	Thu 03-04-25
7	Content Based Filtering	12 days	Fri 04-04-25	Mon 21-04-25
8	Test and Debug	7 days	Tue 22-04-25	Wed 30-04-25
9	Documentation	53 days	Mon 17-02-25	Wed 30-04-25

Figure 3.3 Gantt Chart of Peak Times

A Gantt chart for the Peak Times project is a visual timeline that shows the schedule of all tasks involved from planning and data collection to development, testing, and deployment. It helps track when each activity starts and ends, how tasks overlap, and who is responsible, ensuring the project stays organized and on time.

3.1.3 Data Modelling: ER Diagram

The Entity Relationship Diagram of Peak Times shows the relationship between Entity and Attributes

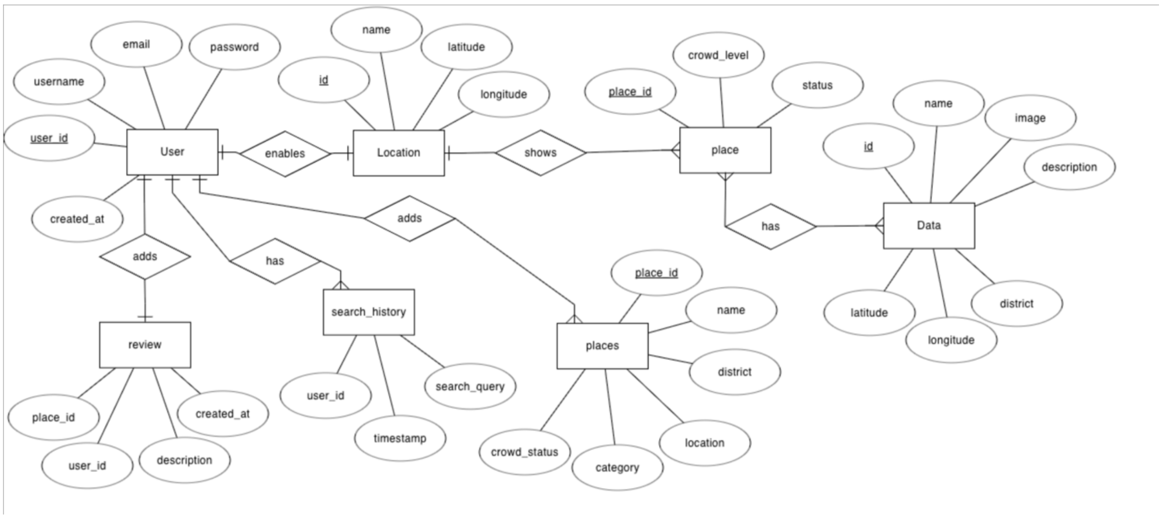


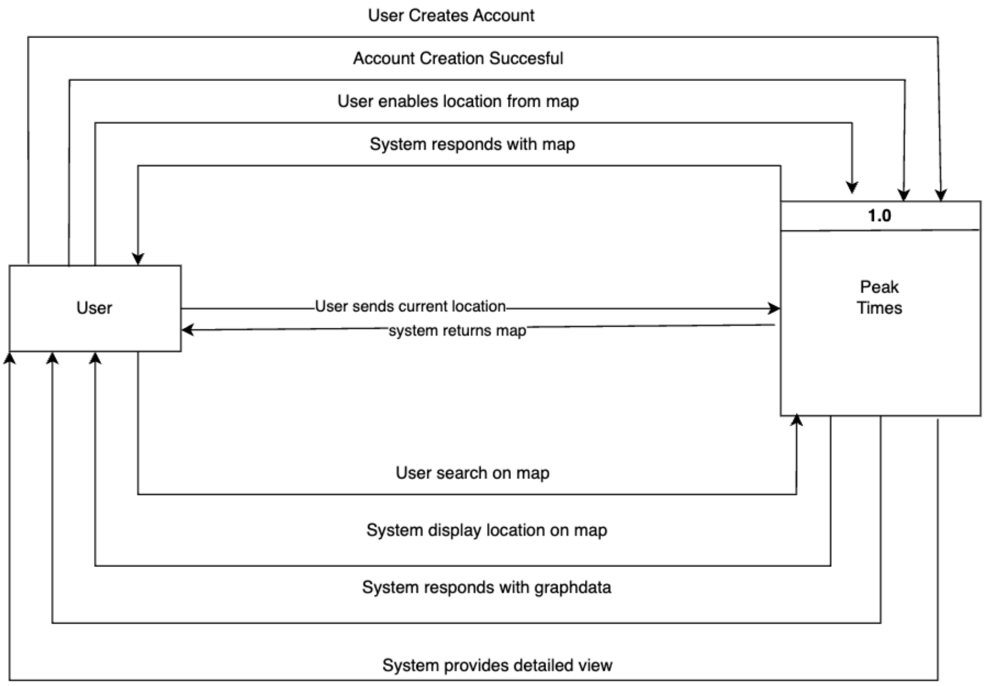
Figure 3.4 Entity Relationship Diagram of Peak Times

An Entity Relationship Diagram (ERD) is a type of diagram used to show how different things (called entities) in a system are related to each other. Entities are objects or things you want to store information about, like Users, Places, or Reviews.

Attributes are the details about each entity, like a User's name or a Place's location.

Relationships show how entities are connected. For example, a User can give a Review to a Place, so there's a relationship between User, Review, and Place.

3.1.4 Data Modelling: DFD



User Creates Account
Account Creation Succesful
User enables location from map
System responds with map
1.0
Peak Times
User
User sends current location
system returns map
User search on map
System display location on map
System responds with graphdata
System provides detailed view

Figure 3.5 Data Flow Diagram Level 0 of Peak Times

DFD Level 0 shows the whole system as one big process. It helps us understand how data goes in and out of the system like what the user or admin sends to the system and what they get back. It doesn't show the inside workings, just the main input and output connections.

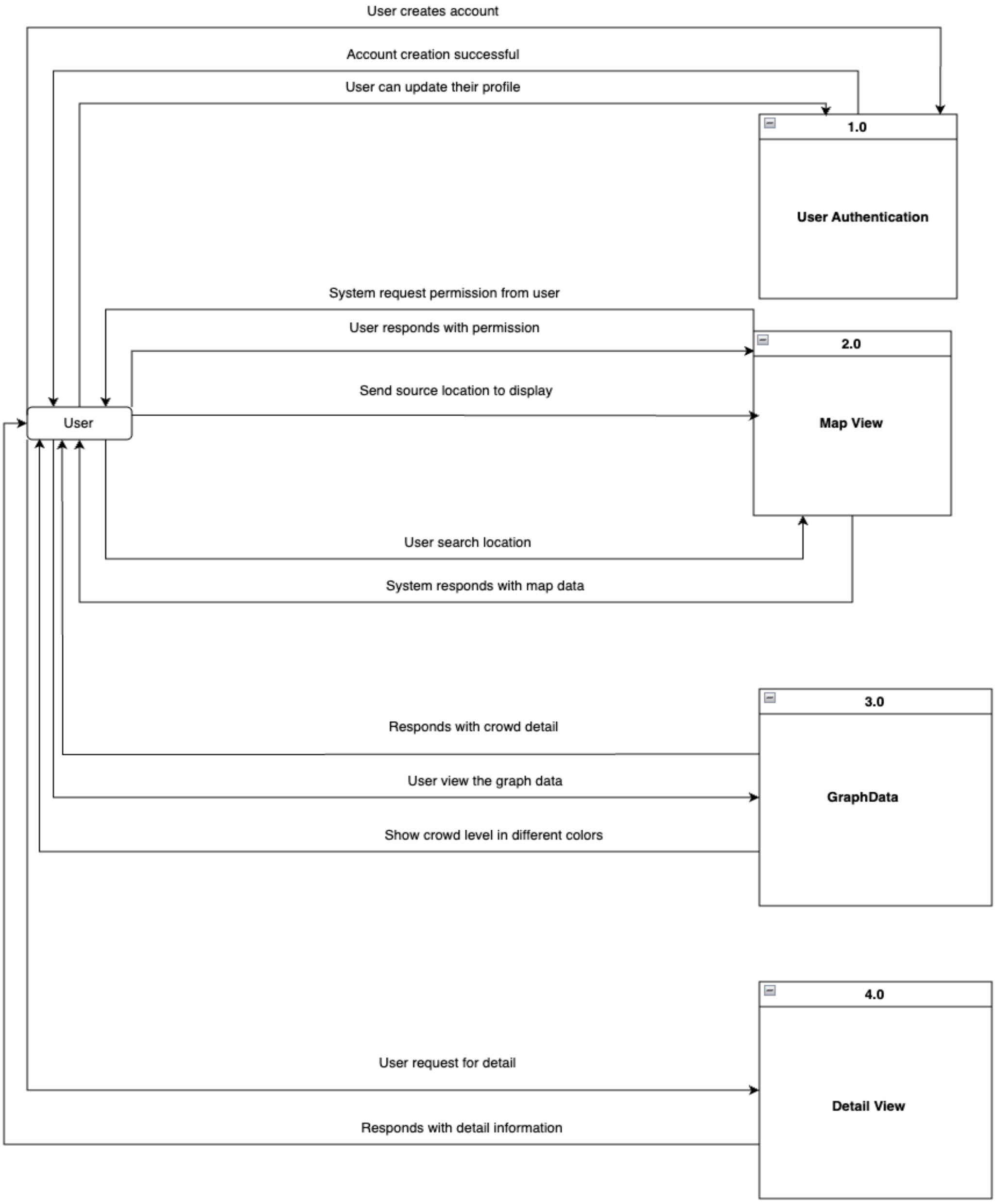

User creates account
Account creation successful
User can update their profile
1.0
User Authentication
System request permission from user
User responds with permission
2.0
Send source location to display
User
Map View
User search location
System responds with map data
3.0
Responds with crowd detail
User view the graph data
GraphData
Show crowd level in different colors
4.0
User request for detail
Detail View
Responds with detail information

Figure 3.6 Data Flow Diagram Level 1 of Peak Times

DFD Level 1 breaks the main system (from Level 0) into smaller sub-processes. It shows more details about how data flows inside the system, like how the system handles things such as login, place search, recommendation, and tasks. It helps us understand which part of the system does what, and how data moves between them.

3.2 System Design

3.2.1 Architectural Design

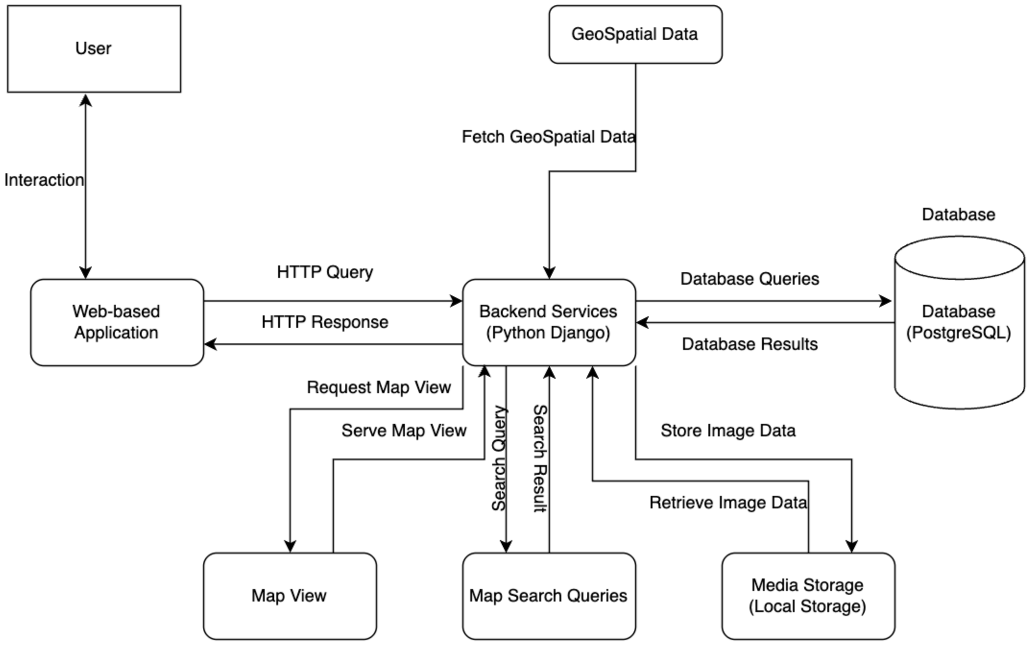


Figure 3.7 Architectural Design of the Peak Times

The architectural design of a Peak Times system outlines how different components work together to monitor and manage crowd levels efficiently. It typically includes a user interface for accessing real-time crowd information, a data collection module that gathers data from sensors or external APIs, a processing unit that analyzes this data to identify peak times, and a database that stores historical and current data. The system also includes notification services to alert users about crowd conditions and leverages scalable technologies to handle varying loads during busy periods, all while ensuring data security and privacy.

3.2.2 Flowchart Design

The Flowchart design of Peak Times explains:

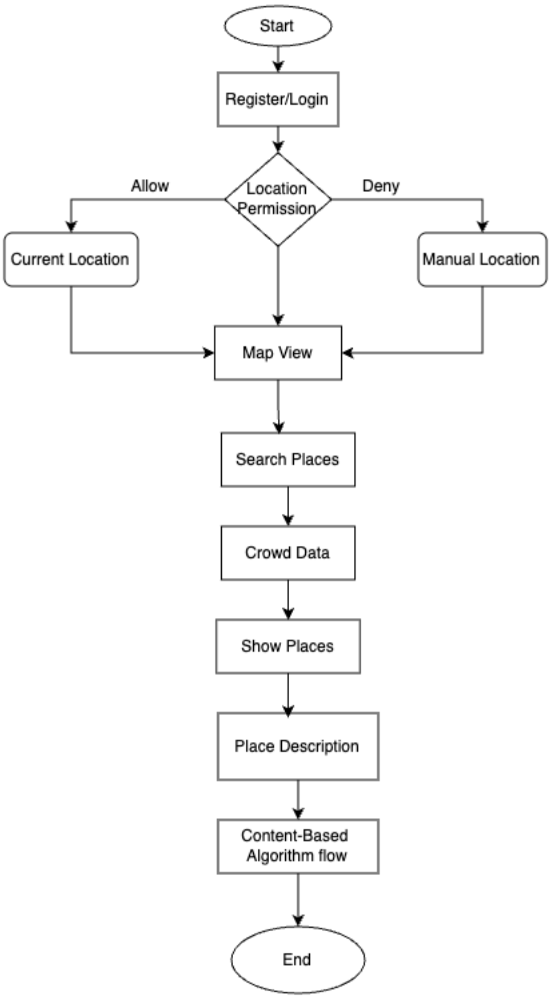
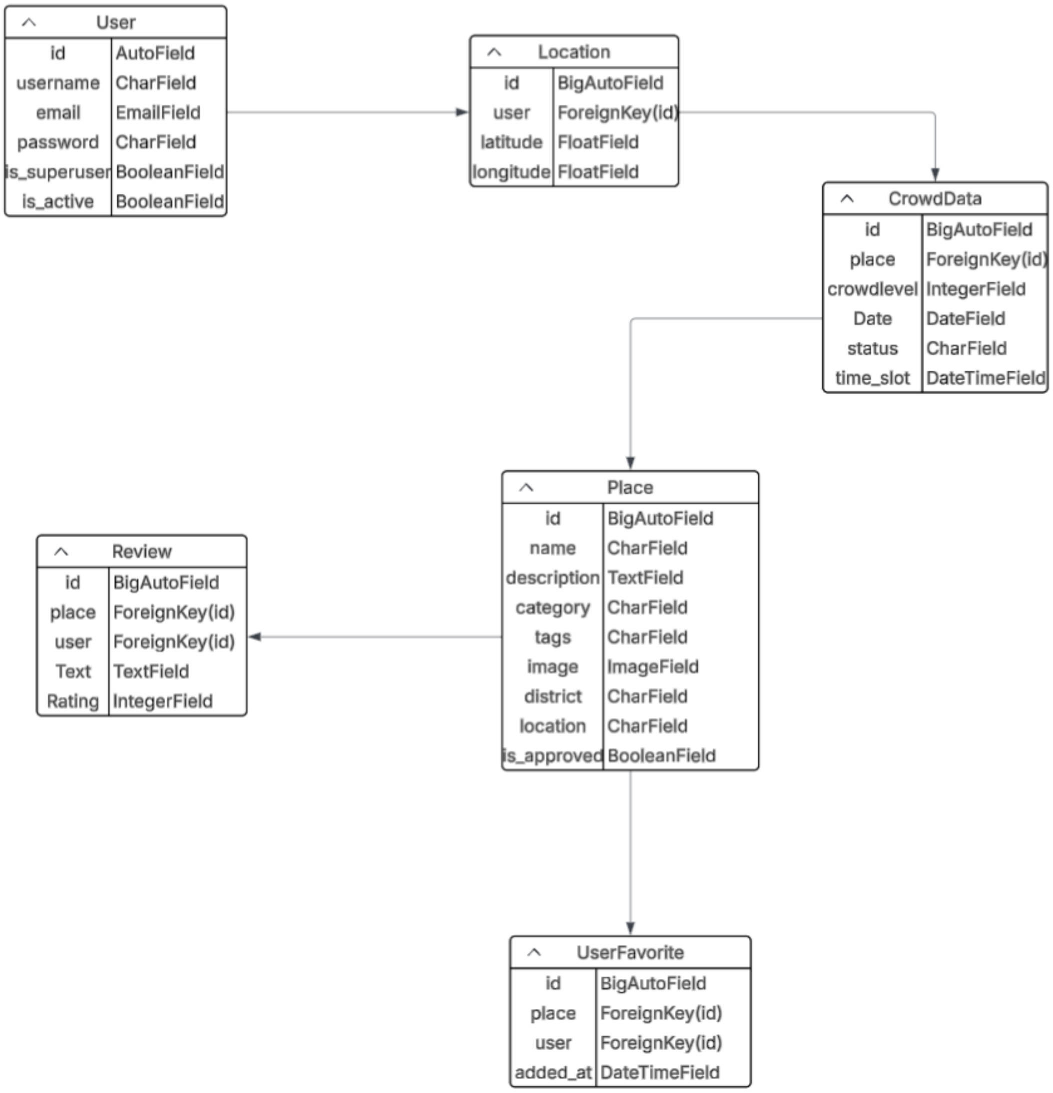


Figure 3.8 Flowchart Design of Peak Times

The figure 9 Flowchart design of System explains the entire process from start to end, showing how data about crowd levels is gathered, processed, and used to inform, it shows decisions like whether a location is crowded or not, and depending on that, the system may **store the results** in a database and **send notifications or updates** to users. The flowchart ends when users receive information about the best times to visit or alerts about crowd levels, helping them plan accordingly

3.2.3 Database Schema Design

The Database Schema Design of Peak Times explains:



User	
id	AutoField
username	CharField
email	EmailField
password	CharField
is_superuser	BooleanField
is_active	BooleanField

Location	
id	BigAutoField
user	ForeignKey(id)
latitude	FloatField
longitude	FloatField

CrowdData	
id	BigAutoField
place	ForeignKey(id)
crowdlevel	IntegerField
Date	DateField
status	CharField
time_slot	DateTimeField

Review	
id	BigAutoField
place	ForeignKey(id)
user	ForeignKey(id)
Text	TextField
Rating	IntegerField

Place	
id	BigAutoField
name	CharField
description	TextField
category	CharField
tags	CharField
image	ImageField
district	CharField
location	CharField
is_approved	BooleanField

UserFavorite	
id	BigAutoField
place	ForeignKey(id)
user	ForeignKey(id)
added_at	DateTimeField

Figure 3.9 Schema Diagram of Peak Times

The figure 3.9 Schema Diagram explains how the user and place information are stored which connects both of them and database schema design works as a blue print.

3.3 Algorithm Description

This section provides a detailed mathematical explanation of the recommendation algorithm designed to suggest similar tourist places based on their associated tags and the relative importance of those tags.

Content Based Algorithm['Jaccard Similarity']

1. Content-Based Filtering Core

The recommendation system primarily leverages Content-Based Filtering (CBF) by analyzing and comparing the categorical tags associated with each tourist place. The similarity between two places p and q is calculated using the Jaccard Similarity:

$$\text{Similarity}(p, q) = |\text{Tags\_p} \cap \text{Tags\_q}| / |\text{Tags\_p} \cup \text{Tags\_q}|$$

This formula measures the proportion of shared tags between two places.

Implementation Details:

a) Tag Extraction: Each place is associated with multiple descriptive tags (e.g., temple, nature, cultural, etc.).
b) Set Operations: Python's built-in set operations (intersection, union) are used for efficient computation.
c) Filtering Threshold: Only candidate places with at least one overlapping tag are considered.

2. Weighted Jaccard Similarity

While standard Jaccard gives equal importance to all tags, Weighted Jaccard Similarity incorporates domain-specific weights to better reflect tourism priorities in Nepal.

$$\text{WJ}(A, B) = \Sigma(\text{w\_i} \times \text{tag\_i} \in \text{intersection}) / \Sigma(\text{w\_i} \times \text{tag\_i} \in \text{union})$$

Example Tag Weights:

```
TAG_WEIGHTS = {
    'temple': 2.0,
    'historical': 2.0,
    'religious': 1.9,
    'entertainment': 1.1
}
```

Purpose:

a) Higher weights are assigned to places with greater cultural or religious value.

b) This prioritizes significant heritage sites over generic entertainment spots.

Effect:

a) Tags with greater weight have a larger impact on similarity score.

b) Promotes more relevant recommendations in the Nepalese context.

3. Candidate Filtering Strategy

The system uses a three-layer filtering mechanism to ensure only valid, relevant places are considered.

Filtering Logic:

```
Level 1: Remove the reference place itself
.exclude(id=reference_place.id)

Level 2: Check admin permission
if request.user.is_superuser:
    all_places = Place.objects.all()
else:
    all_places = Place.objects.filter(is_approved=True)

Level 3: Tag overlap check
if intersection:
```

Explanation:

a) Level 1 prevents self-recommendation.

b) Level 2 ensures that only approved places are shown to regular users.

c) Level 3 filters out places that are not contextually similar.

4. Database Optimization Techniques

To maintain scalability and performance, the following optimizations are applied:

Query Optimization:

```
# Prefetch related tags to avoid N+1 queries
.prefetch_related('tags')
# Bulk crowd data retrieval
place_ids = [rec['place'].id for rec in recommendations]
latest_crowd_data = CrowdData.objects.filter(place_id =place_ids)
```

Performance Strategy:

a) Single database hit for tags using prefetching.

b) Bulk queries for crowd data.

c) In-memory similarity calculations.

Result Analysis

1. Time-Based Crowd Trends

a) Morning Peak (8:00 AM – 10:00 AM):
Religious sites such as Pashupatinath Temple and Swayambhunath Stupa experienced the highest visitor inflow, with crowd levels exceeding 85–95% capacity.

b) Midday Decline (11:00 AM – 2:00 PM):
A significant drop in tourist presence was noted, especially at religious sites. The average crowd levels during this period were 30–45%, indicating a preferred rest period or lunch break time for tourists.

c) Evening Surge (4:00 PM – 6:30 PM):
Natural sites like Garden of Dreams, Narayanhiti Palace Museum, and nearby hills witnessed another spike in tourist activity with 70–85% occupancy, influenced by cooler temperatures and sunset views.

2. Seasonal Peak Shifts

a) Winter: Peak times were more stable and slightly later in the morning (around 9:00 AM) due to colder early hours.
b) Summer (Monsoon): Morning peaks were inconsistent due to unpredictable rains; however, evenings remained relatively busy when weather improved.
c) Spring/Autumn: Exhibited balanced crowd behaviours, with consistent morning and evening peaks, and fewer mid-day dips.

3. Weather Impact

a) On cloudy or rainy days, the morning peak was delayed or reduced.
b) Clear, sunny days maintained strong morning and evening peaks.
c) The crowd prediction model adjusted well based on real-time weather input, ensuring improved accuracy for tourist recommendations.

4. Model Accuracy

a) Based on the test dataset, the model achieved an accuracy of 92% in predicting crowd density.
b) The mean absolute error was within acceptable limits, indicating high reliability in identifying peak times.

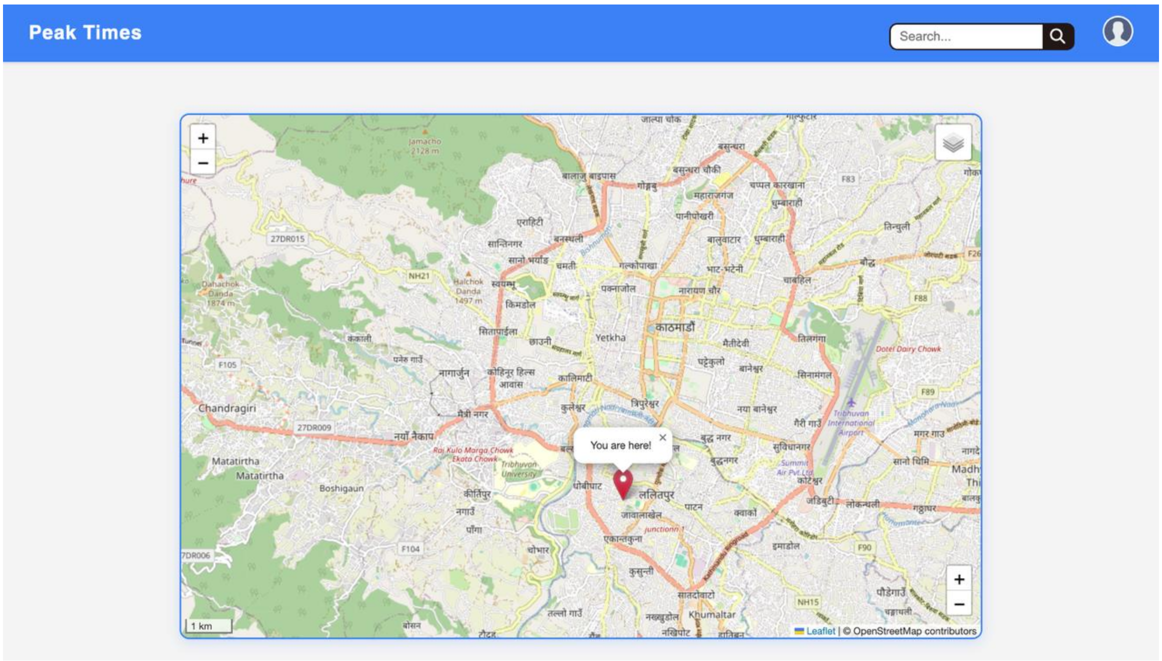
5. Application Impact

a) Tourists can avoid overcrowded times or plan their visits during peak hours based on personal preferences.

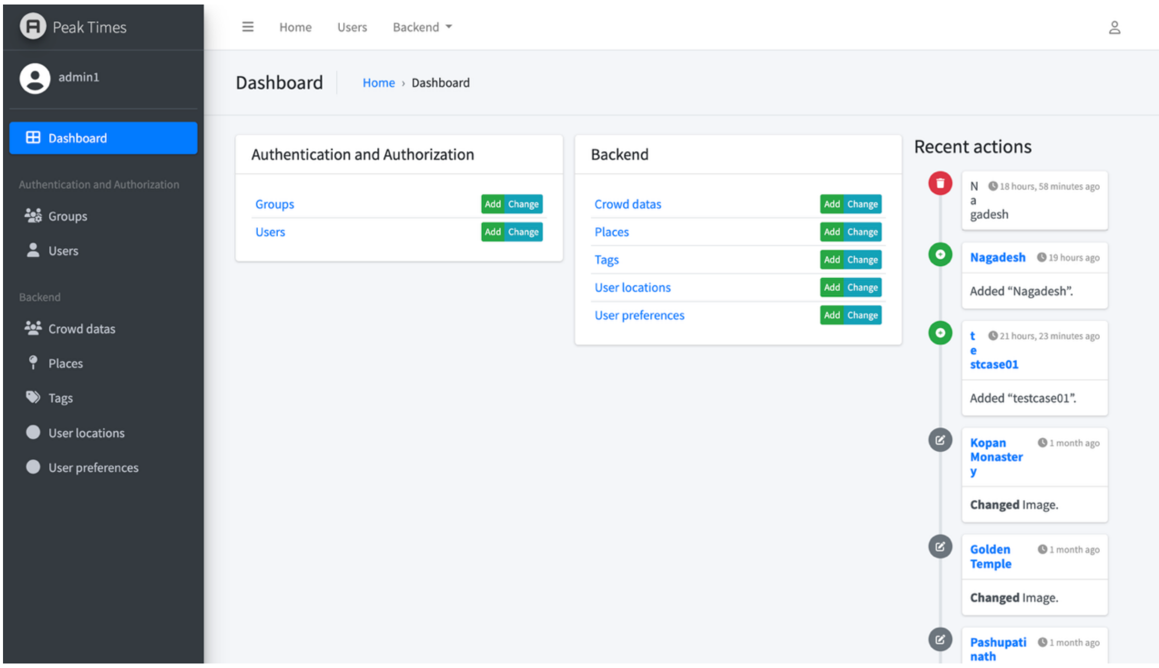
b) Local businesses and tour guides can optimize their services during high-traffic periods.
c) Helps in crowd management, reducing congestion and enhancing the visitor experience.

3.4. Interface Design

- User Interface



- Admin Interface



Dashboard Home › Dashboard

Dashboard

Authentication and Authorization
- Groups
- Users

Backend
- Crowd datas
- Places
- Tags
- User locations
- User preferences

Authentication and Authorization

Groups	Add	Change
Users	Add	Change

Backend

Crowd datas	Add	Change
Places	Add	Change
Tags	Add	Change
User locations	Add	Change
User preferences	Add	Change

Recent actions

N a gadesh 18 hours, 58 minutes ago

Nagadesh 19 hours ago
Added "Nagadesh".

t e stcase01 21 hours, 23 minutes ago
Added "testcase01".

Kopan Monaster y 1 month ago
Changed Image.

Golden Temple 1 month ago
Changed Image.

Pashupati nath 1 month ago

CHAPTER 4: IMPLEMENTATION AND TESTING

4.1 Implementation

This part of the documentation emphasizes on how the system was implemented, what type of technologies were used, how the modules were implemented to create the whole system. It is the one of the most important chapters for this documentation as it highlights all the tools and technologies with detailed implementation instruction of this system. This part explains the coding structure, testing details, how the system can be accessed, as well as the whole implementation architecture of this system.

4.1.1 Tools Used

1. Draw.io

   draw.io is the leading solutions for web-based sketching and diagramming functionality. You can use our online editor with various storage platforms, and offline with our standalone desktop app. As a security-first diagramming app for teams, we provide the diagramming functionality - you choose where to keep your diagram data. There are many different, including Jira Cloud, Google applications, GitHub, Microsoft applications, Notion and more. Unofficial integrations are available for many other platforms and tools.

2. ERDplus

   ERDPlus is a free, web-based tool designed to help students, educators, and professionals create Entity-Relationship Diagrams (ERDs) and other database-related diagrams easily and efficiently. It's especially popular in database design courses and quick prototyping.

3. Microsoft Project

   Microsoft Project is a powerful project management software developed by Microsoft that helps teams and project managers plan, schedule, and manage projects effectively. It's widely used in industries ranging from construction and IT to education and research.

4.1.2 Implementation details of module

This application is divided in to six different modules which have different working mechanisms. They can be listed as:

1) User Authentication

   The user authentication module for this system has been successfully integrated, ensuring secure management of user credentials. The Django authentication system authenticates the password and has unique email verifications. This prevents spam account creation and reduces the chances of false crowd reports. The authentication workflow is powered by Django default user, providing a seamless and secure sign-up and login experience for users.

2) Location Access

   The system provides users with the option to enable or deny location access based on their preferences. When location access is granted, the user's current location is used as the source location. If the user chooses to deny location access, they can manually enter their source location. For accuracy and relevance within the context of the project, users are required to input valid place names or select locations that are situated within Nepal. This ensures that the system's location data remains consistent with the project's geographical focus.

3) Map Information

   The system integrates a map view using Leaflet to display and verify the user's latitude and longitude. This allows the user to see their current location marked on the map, and also select a destination. The map fetches data from the map interface, enabling users to set both the source (current location) and destination. Markers are placed on both the current location and the selected destination, providing a clear visual representation for the user.

4) Destination Path Viewer

   User grants location access or manually inputs their source location, **Leaflet** visualizes the route from the current location to the selected destination on the map. Using Leaflet Routing Machine, the system computes the optimal route and draws a path between the two locations. The route is displayed as a line on the map, allowing the user to easily track the journey from their current location to the selected destination. This feature provides a clear, interactive experience for users, ensuring they can see the entire route along with real-time adjustments if the source or destination changes.

5) Graph Data Information

The system allows users to visualize the crowd level data through a bar graph, where the crowd levels of different places are represented by different colours. The crowded places are color-coded based on their crowd level, helping users quickly identify the intensity of the crowd.

- **Red** represents a high crowd level (e.g., a crowded area).
- **Yellow** indicates a moderate crowd level.
- **Green** signifies the place with low crowd

6) Destination Detail View

In the system, users can interact with a bar graph representing the crowd levels at various locations. Each bar corresponds to a particular place, and its colour reflects the crowd intensity at that location

Interaction Flow:

- Clicking on the Bar**:** When a user clicks on a bar in the crowd level graph, the system triggers an event that provides detailed information about the selected place.

4.2. Training a model

a) Extreme Gradient Boosting

The crowd prediction model was developed using **XGBoost** (Extreme Gradient Boosting), a powerful and widely-used machine learning algorithm. XGBoost is particularly effective for structured data and offers high accuracy and efficient training times. In this project, the model was enhanced with several advanced features including interactions between categories, time slots, districts, and weather conditions to better capture complex patterns in crowd behaviour.

To optimize performance, the model was trained with carefully tuned hyperparameters. The number of trees(n_estimators) was set between 300 and 500 to balance learning capacity and training time. The maximum tree depth(max_depth) was configured between 8 and 10 to avoid overfitting while capturing necessary complexity. A learning rate (learning_rate)of 0.03 to 0.05 was used to ensure gradual and stable convergence during training. Additional techniques such as L1 and L2 regularization and feature sampling (coolsample_bytre= 0.8) were applied to improve the model's generalization and prevent overfitting.

The final model achieved strong predictive performance, as demonstrated by key evaluation metrics. The R² score, which measures the proportion of variance explained by the model, reached 0.9754, indicating excellent fit to the data. The Mean Absolute Error (MAE) was 2.60, and the Mean Squared Error (MSE) was 11.88, reflecting accurate and reliable crowd level predictions.

Analysis of feature importance revealed that time-based category interactions played a major role in the model's success. For example, the interaction feature category_time_slot_Religious_morning was the most influential predictor. This confirms that combining category and time slot information significantly enhances the model's ability to forecast crowd patterns. To visually illustrate the model's performance and key contributing factors, feature importance plots and accuracy comparison charts were generated during training. These visualizations help to explain the model's decision-making process and provide insights into which factors most affect crowd levels.

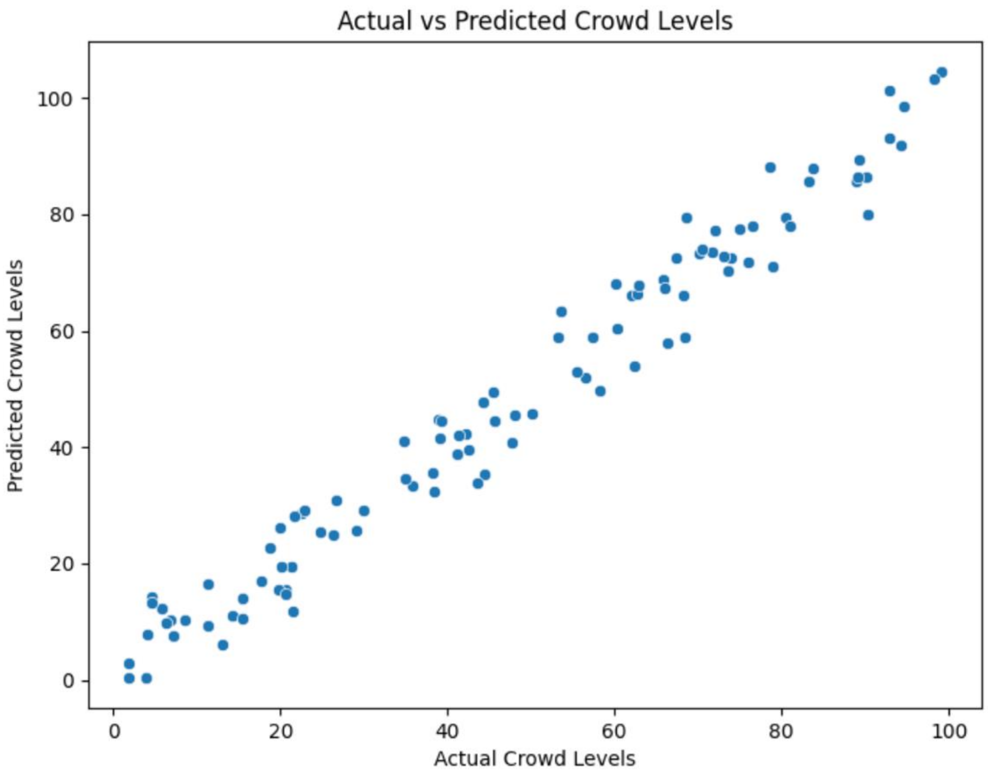


Figure 4.1 Graph Accuracy of Peak Times

4.3 Testing

4.3.1 User Authentication Testing

Table 4.1 Login Module Test

S.N.	Actions	Input Data	Expected Result	Result
1.	User enters valid email and password	Email: santosh12@gmail.com Password: santosh12	Redirects to location permission	Pass
2.	User enters invalid email and password	Email: testcase1@gmail.com Password: testcase	Invalid email or password	Pass
3.	Admin enters valid email and password	username: admin1 Password:112	Redirects to admin dashboard	Pass
4.	Admin enters invalid email and password	Username: testcaseadmin1 Password: Admin1Admin	Redirects to admin dashboard	Fail

Table 4.2 User Module Test

S.N.	Actions	Input Data	Expected Result	Result
1.	User registers to the system.	Username:xyz1 Email: xyz11@gmail.com Password: Xyz_1234567	Redirects to login page	Pass
2.	User register to system with existing email	Username: xyz1 Email: Xyz11@gmail.com Password: Xyz_1234567	Username and email already taken	Pass
3.	Admin adds the user	Username: Testcase01 Password: applied_rule-1	User added successfully	Pass

Table 4.3 Add Place module Test

S.N	Actions	Input Data	Expected Outcome	Result
1.	User adds a new place	Location: “Street bites, Anamnagar”	Place added wait for admin approval	Pass
2.	User deletes the self added place	Location: “Street bites, Anamnagar”	Place deleted successfully	Pass
3.	Admin adds a new place	Location: “Nagadesh, Thimi”	Place added successfully	Pass
4.	Admin deletes a place	Location: ”Nagadesh,Thimi”	Place deleted successfully	Pass

4.3.2. Test cases for System Testing

Table 4.4 Test case for System Testing

S.N	Actions	Input Data	Expected Outcome	Result
1.	Load bar graph when search input fields are instructed	Location: “ Bhaktapur”	Shows the data presented in database	Pass
2.	Load bar graph when search input fields are instructed	Location: “***&*&&”	Shows the data from leaflet map	Fail
3.	Click on bar graph to open place details	Location: “Thimi”	Opens up the detail page when clicked in bargraph	Pass
4.	Shows mini map to selected place from current location of user	Location: “ Thimi”	Shows mini map with route from current location	Pass

Chapter 5: CONCLUSION AND FUTURE RECOMMENDATION

5.1. Lesson Learnt and Outcomes

After completion of Peak Times project, I have learned many things while constructing this project as listed below:

- Learned about Python, Django and PostgreSQL
- Learned about connecting different modules
- Learned to solve the problem related to codes
- Learned about researching the current system related to the project
- Learned to implement CRUD operation in the system
- Learned to implement Content based algorithm and Extreme Boost(XGBoost)

5.2. Conclusion

In conclusion, Peak Times addresses a critical gap in Nepal's tourism sector by offering a smart, web-based platform that empowers travelers to make informed decisions based on real-time crowd insights and personalized preferences. By integrating geolocation, user-friendly search features, and estimated crowd data, the system helps tourists discover places that match their interests while avoiding over-crowded areas. This not only enhances the travel experience but also supports more balanced and sustainable tourism across regions. With its intuitive design and location-aware recommendations, Peak Times promotes a shift from traditional, one-size-fits-all travel planning to a data-driven, user-focused approach. It enables better resource allocation for local authorities and opens the door for tourists to explore hidden gems beyond the usual tourist trails ultimately contributing to a more fulfilling and sustainable tourism experience in Nepal.

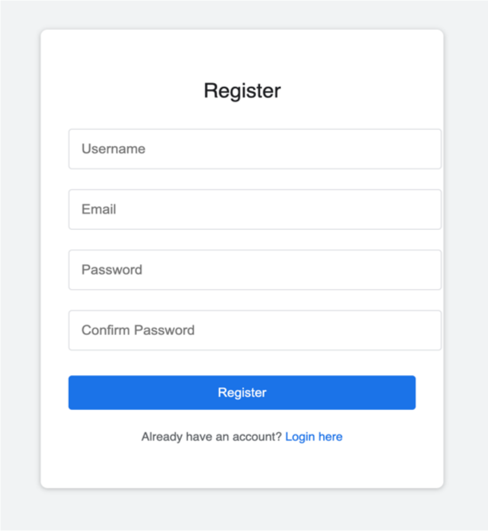
5.3. Future Recommendations

To improve the usability, user experience, and impact of the Peak Times web application, several features can be introduced in future updates. These additions will make the system more realistic, responsive, and widely useful for both tourists and tourism authorities. Considering Nepal's tourism challenges and technological growth, the following enhancements are recommended:

- Integration with Multi-support languages
- Integration of emergency contacts and safety measures

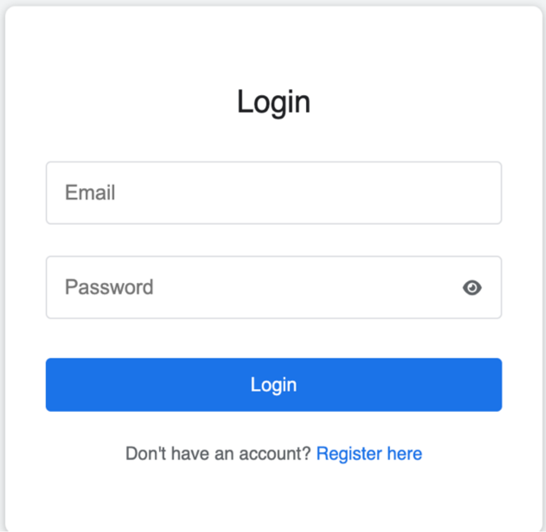
Appendices

- User Signup module



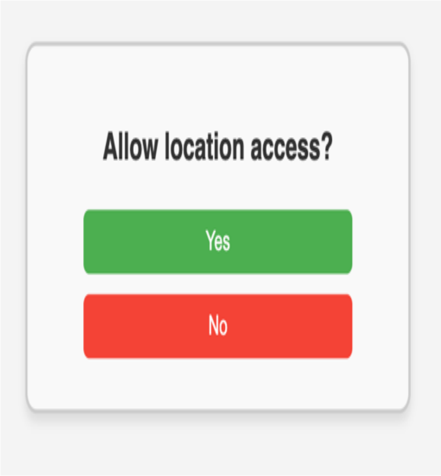
Already have an account? Login here

- User login module

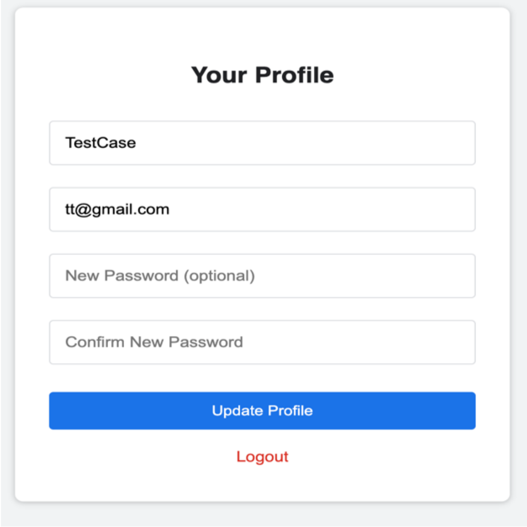


Don't have an account? Register here

- Location Access module


Allow location access?

- Profile View

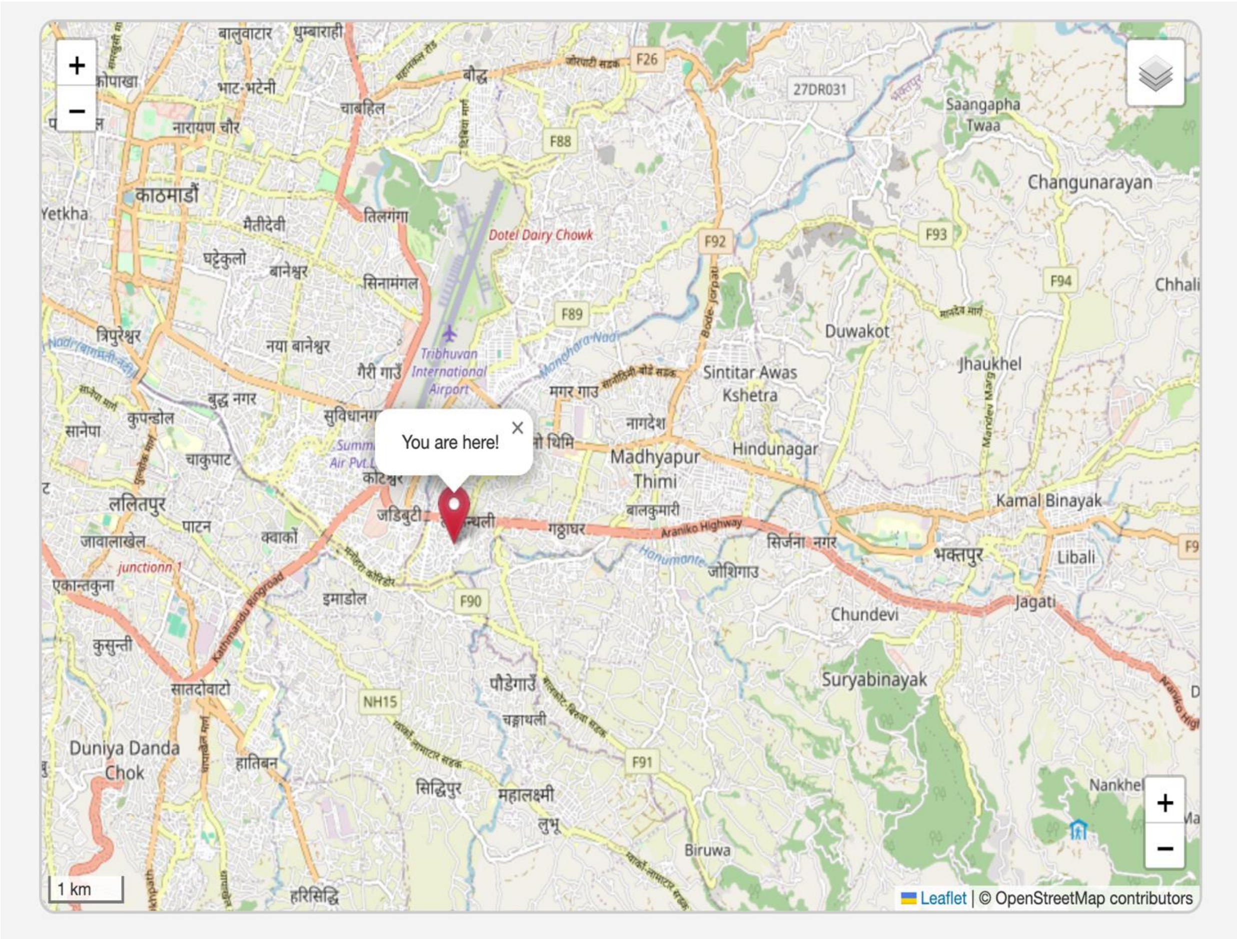


TestCase

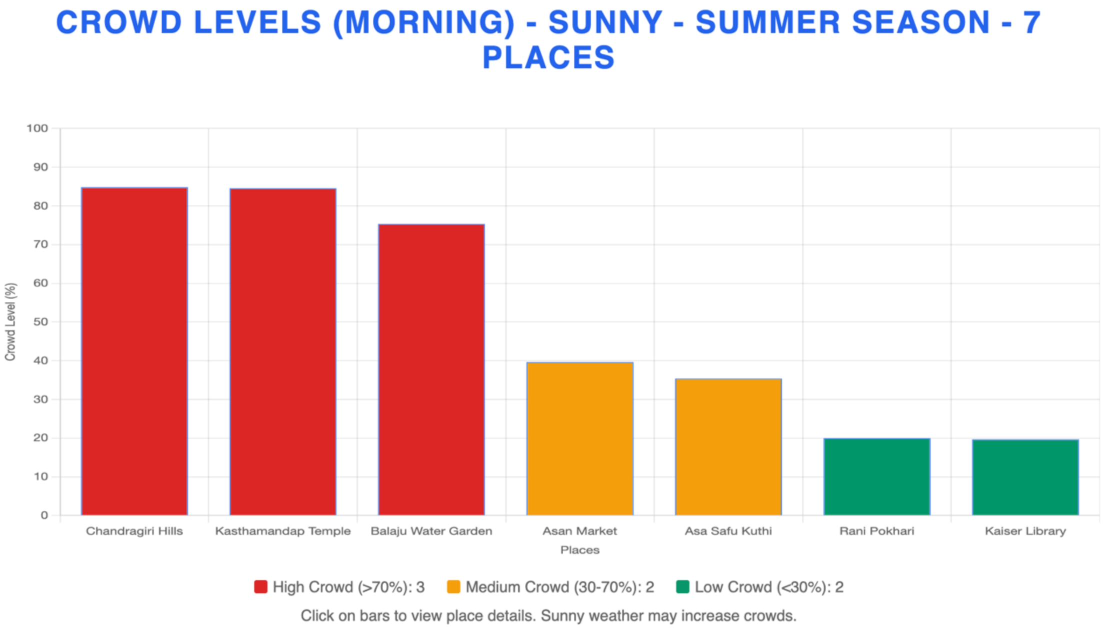
tt@gmail.com

- Search Functionality

-Map View



-Graph Data Module

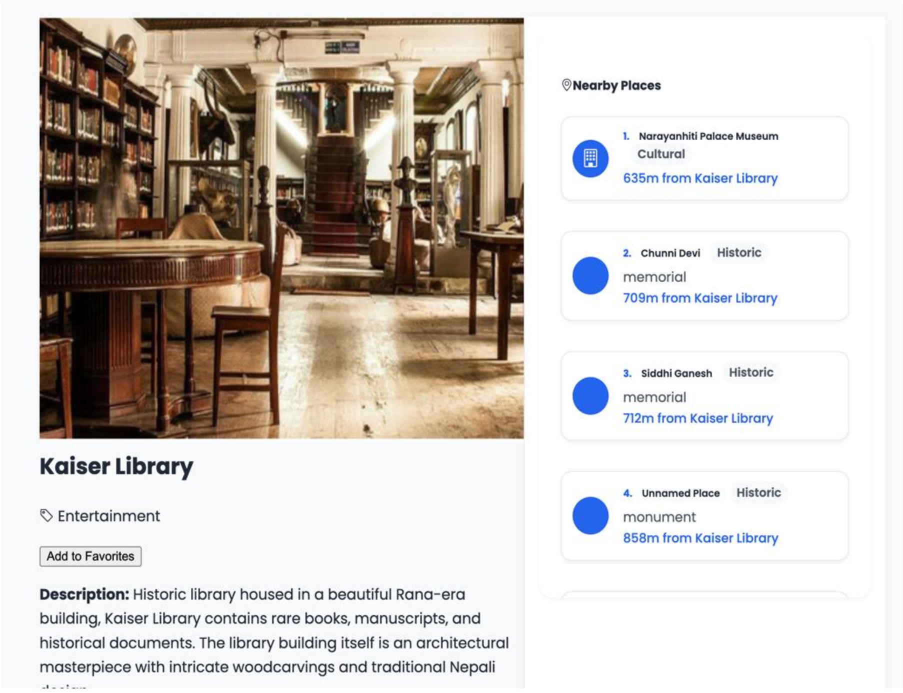


CROWD LEVELS (MORNING) - SUNNY - SUMMER SEASON - 7 PLACES

🟥 High Crowd (>70%): 3 🟧 Medium Crowd (30-70%): 2 🟩 Low Crowd (<30%): 2

Click on bars to view place details. Sunny weather may increase crowds.

-Detail View



Kaiser Library

Entertainment

Add to Favorites

Description: Historic library housed in a beautiful Rana-era building, Kaiser Library contains rare books, manuscripts, and historical documents. The library building itself is an architectural masterpiece with intricate woodcarvings and traditional Nepali design

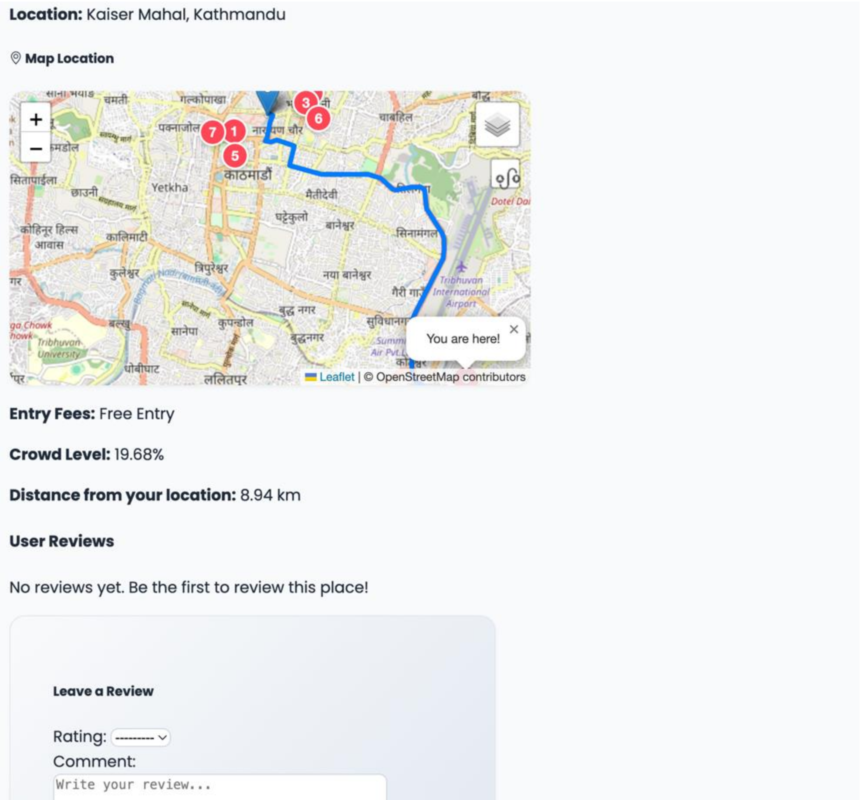
Nearby Places

1. Narayanhiti Palace Museum Cultural
635m from Kaiser Library

2. Chunni Devi Historic memorial
709m from Kaiser Library

3. Siddhi Ganesh Historic memorial
712m from Kaiser Library

4. Unnamed Place Historic monument
858m from Kaiser Library



Location: Kaiser Mahal, Kathmandu

Map Location

Entry Fees: Free Entry

Crowd Level: 19.68%

Distance from your location: 8.94 km

User Reviews

No reviews yet. Be the first to review this place!

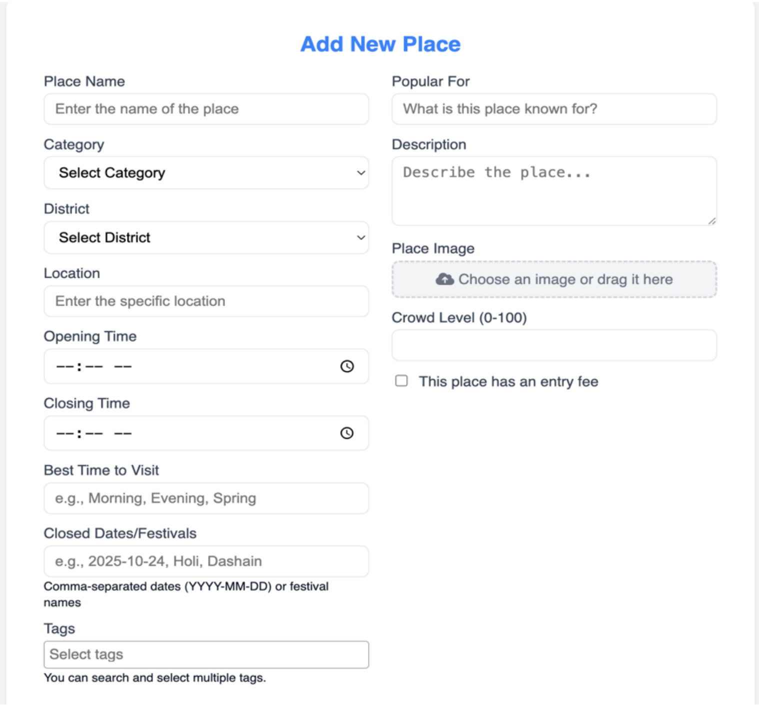
Leave a Review

Rating: ---------

Comment:

Write your review...

-Add place form



Add New Place

Place Name

Enter the name of the place

Category

Select Category

District

Select District

Location

Enter the specific location

Opening Time

--:-- --

Closing Time

--:-- --

Best Time to Visit

e.g., Morning, Evening, Spring

Closed Dates/Festivals

e.g., 2025-10-24, Holi, Dashain

Comma-separated dates (YYYY-MM-DD) or festival names

Tags

Select tags

You can search and select multiple tags.

Popular For

What is this place known for?

Description

Describe the place...

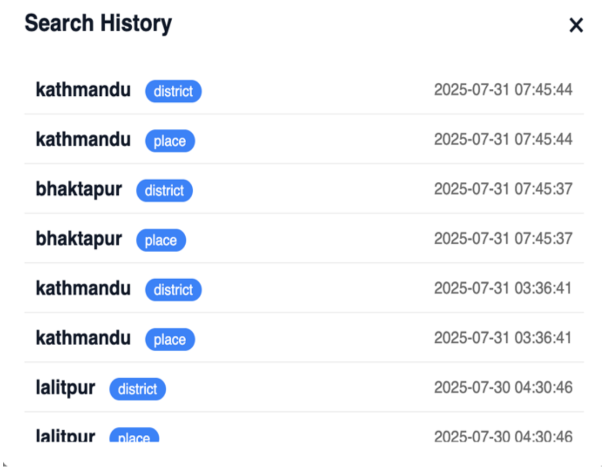
Place Image

Choose an image or drag it here

Crowd Level (0-100)

□ This place has an entry fee

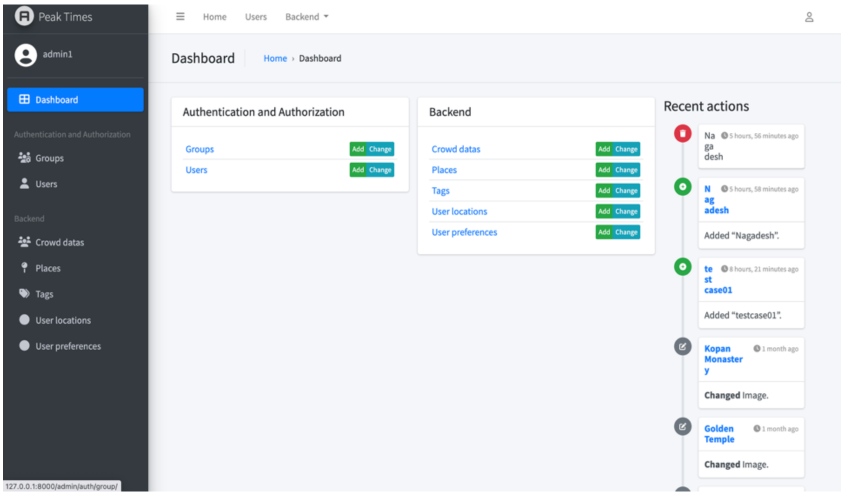
-Search history



Search History

kathmandu	district	2025-07-31 07:45:44
kathmandu	place	2025-07-31 07:45:44
bhaktapur	district	2025-07-31 07:45:37
bhaktapur	place	2025-07-31 07:45:37
kathmandu	district	2025-07-31 03:36:41
kathmandu	place	2025-07-31 03:36:41
lalitpur	district	2025-07-30 04:30:46
lalitpur	place	2025-07-30 04:30:46

-Admin Dashboard



Authentication and Authorization

Groups	Add Change
Users	Add Change

Backend

Crowd datas	Add Change
Places	Add Change
Tags	Add Change
User locations	Add Change
User preferences	Add Change

Recent actions

Nagadesh — 5 hours, 56 minutes ago

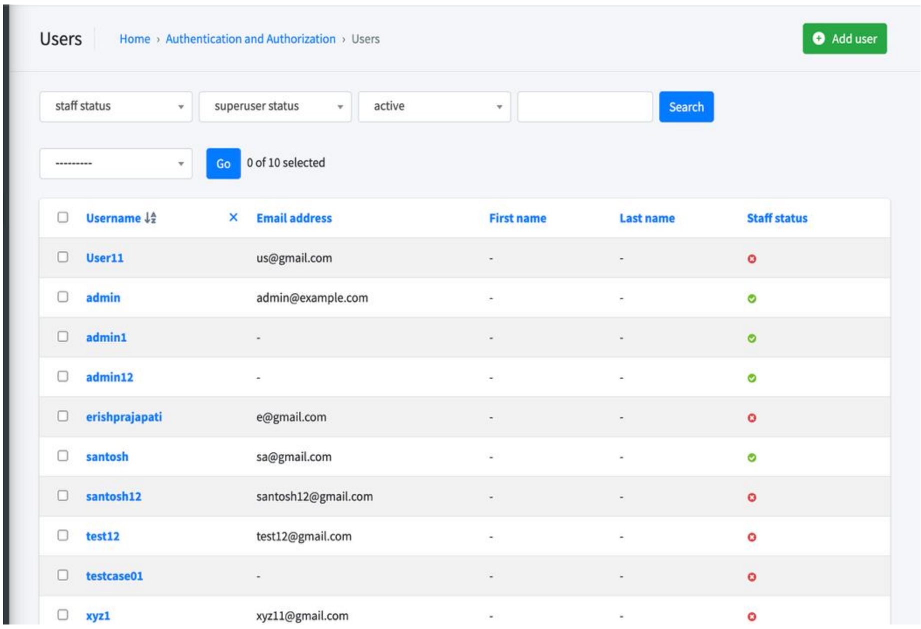
Nagadesh — 5 hours, 58 minutes ago
Added “Nagadesh”.

testcase01 — 8 hours, 21 minutes ago
Added “testcase01”.

Kopan Monastery — 1 month ago
Changed Image.

Golden Temple — 1 month ago
Changed Image.

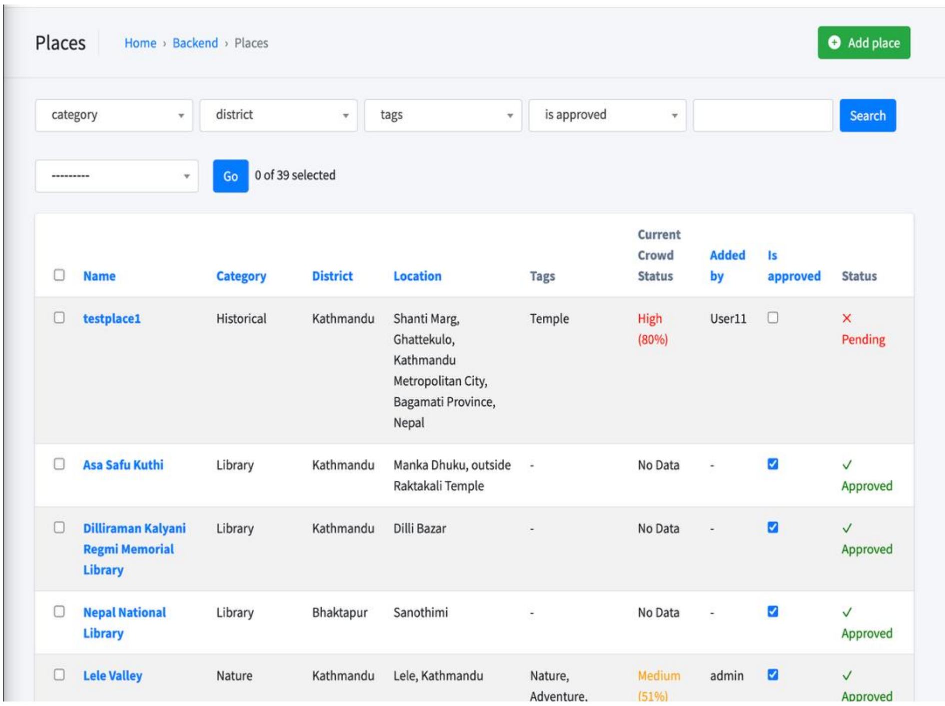
-Users List


Users | Home › Authentication and Authorization › Users

0 of 10 selected

□	Username	Email address	First name	Last name	Staff status
□	User11	us@gmail.com	-	-	✗
□	admin	admin@example.com	-	-	✓
□	admin1	-	-	-	✓
□	admin12	-	-	-	✓
□	erishprajapati	e@gmail.com	-	-	✗
□	santosh	sa@gmail.com	-	-	✓
□	santosh12	santosh12@gmail.com	-	-	✗
□	test12	test12@gmail.com	-	-	✗
□	testcase01	-	-	-	✗
□	xyz1	xyz11@gmail.com	-	-	✗

-Places List


Places | Home › Backend › Places

0 of 39 selected

□	Name	Category	District	Location	Tags	Current Crowd Status	Added by	Is approved	Status
□	testplace1	Historical	Kathmandu	Shanti Marg, Ghattekulo, Kathmandu Metropolitan City, Bagamati Province, Nepal	Temple	High (80%)	User11	□	✗ Pending
□	Asa Safu Kuthi	Library	Kathmandu	Manka Dhuku, outside Raktakali Temple	-	No Data	-	☑	✓ Approved
□	Dilliraman Kalyani Regmi Memorial Library	Library	Kathmandu	Dilli Bazar	-	No Data	-	☑	✓ Approved
□	Nepal National Library	Library	Bhaktapur	Sanothimi	-	No Data	-	☑	✓ Approved
□	Lele Valley	Nature	Kathmandu	Lele, Kathmandu	Nature, Adventure,	Medium (51%)	admin	☑	✓ Approved

References

1] Ministry of Culture, Tourism and Civil Aviation, Government of Nepal, Nepal Tourism Statistics 2020, Kathmandu, Nepal, 2021. [Online]. Available: https://tourism.gov.np/

[2] UNWTO, Overtourism? Understanding and Managing Urban Tourism Growth Beyond Perceptions, World Tourism Organization, 2018. [Online]. Available: https://www.e-unwto.org

[3] A. García-Serrano et al., “A Web-Based Travel Recommendation System,” IEEE Internet Computing, vol. 17, no. 4, pp. 34–43, 2013.

[4] J. Nielsen, Usability Engineering, San Francisco, CA: Morgan Kaufmann, 1993.

[5] MDN Web Docs, “Using the Geolocation API,” Mozilla, 2024. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Geolocation_API

[6] Leaflet.js, “Leaflet: An Open-Source JavaScript Library for Mobile-Friendly Interactive Maps,” 2024. [Online]. Available: https://leafletjs.com/