ENHANCED CRIME HOTSPOT PREDICTION AND VISUALIZATION FOR WOMEN'S SAFETY THROUGH DEEP LEARNING

A PROJECT REPORT

Submitted by

GAYATHRI.B	**812821104017**
IDASCADAR ANBARASI.R	**812821104026**
LYDIA JOICE.A	**812821104036**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

MOOKAMBIGAI COLLEGE OF ENGINEERING, PUDUKKOTTAI

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2025

ANNA UNIVERSITY: 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**ENHANCED CRIME HOTSPOT PREDICTION AND VISUALIZATION FOR WOMEN'S SAFETY THROUGH DEEP LEARNING**" is the Bonafide work of "**GAYATHIRI, IDASCADAR ANBARASI, LYDIA JOICE** " who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.P.VALARMATHI, M.E., M.B.A., Ph.D.,	**Mrs.D.SUDHA, M.E.,**
HEAD OF THE DEPARTMENT,	SUPERVISOR,
Professor,	Assistant Professor,
Computer Science & Engg,	Computer Science & Engg,
Mookambigai College of Engg,	Mookambigai College of Engg,
Srinivasa Nagar,	Srinivasa Nagar,
Pudukkottai-622 502	Pudukkottai-622 502

DATE:

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

Crime hotspots are geographic areas with elevated levels of criminal activity compared to other regions. Women in these hotspots are at increased risk of experiencing various forms of criminal behavior, including sexual harassment, assault, domestic violence, stalking, and human trafficking. Identifying these hotspots is crucial for effective crime prevention and resource allocation by law enforcement agencies. This project presents Safety Locator, a predictive system that uses multimodal deep learning to identify and map crime hotspots where women are particularly vulnerable. The system utilizes the Deep Explainable Decision Tree model, a machine learning algorithm that analyzes historical crime data to predict the likelihood of criminal activity in specific areas. The Deep Explainable Decision Tree model classifies data into crime hotspot categories and visualizes these areas on Google Maps. By analyzing factors such as the number and type of reported crimes, the timing of incidents, and crime locations, the system generates a detailed map highlighting high-risk areas. This map can be shared with the public to increase awareness and promote safety. The development process includes data pre-processing, feature selection, model training, evaluation, hyperparameter tuning, and prediction. The model's performance is assessed using metrics like accuracy, precision, recall, and F1-score, with hyperparameters optimized through cross-validation. Safety Locator aims to assist law enforcement agencies in preventing crime and enhancing public safety by pinpointing areas with high crime probabilities.

ACKNOWLEDGMENT

We sincerely express our whole hearted thanks to our Chief Executive, **Mr.D.Ravi** and Director, **Dr.P.Vasudevan**, **M.E., Ph.D.,** Mookambigai college of Engineering.

We thank **Dr.R.Elangovan**, **M.E., Ph.D.,** Principal, Mookambigai College of Engineering, for furnishing every essential facility for doing our project.

We would also like to express our sincere thanks to our HOD **Dr.P.Valarmathi, M.E., M.B.A., Ph.D.,** and Project Coordinator **Mrs.D.Ramya Cauvery, M.E., (Ph.D),** for their encouragement and guidance in each and every step of the Project.

We also express our gratitude to our Project Guide **Mrs.S.Sudha, M.E.,** for her consistent support and guidance throughout the Project.

We thank all other faculty members of the Department of Computer Science and Engineering for their valuable help, care and guidance throughout the Project.

Last but not least we are extremely indebted to the almighty God, my friends and family members for their adorable support throughout the Project work

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

BPTT	Back Propagation Through Time
DEDT / xDT	Deep Explainable Decision Tree
GIS	Geographic Information System
GPL	GNU General Public License
NCRB	National Crime Records Bureau
RBF	Radial Basis Function
SVM	Support Vector Machine
TDNN	Time Delay Neural Network
UML	Unified Modelling Language
URL	Uniform Resource Locator
UAT	User Acceptance Testing
X^2 / Chi^2	Chi-Square Test
XML	eXtensible Markup Language

CHAPTER 1
INTRODUCTION

1.1. OVERVIEW

Crimes against women are of various types as crimes involving sex for economic gains including prostitution, wrongful confinement, trafficking, dowry extortion, rape, assault, harassment at work place, gang-rape, acid-attack, kidnapping, and other immoral acts are injurious to the society. In present scenario cases of murder, rape, molestation, sexual abuse, and eve-teasing etc. Women feel discrimination, harassment and domestic violence etc at work. Use of money and muscle power to save accused. Politics in the name of caste, religion etc take benefit as well. India at 108th place according to Global Gender Gap Index 2017. A 2017 report by Global Peace Index had claimed India to be the fourth most dangerous country for women travellers.

Violence against women and girls is rooted in gender-based discrimination and social norms. Women are so helpless in the Indian society where many female goddesses are worshipped but still people don’t respect women. It is high time when the social revolution is needed to root out this evil from Indian society and given respect and recognition to our women.

Crime Hotspot

Crime hotspots are areas that have high crime intensity. These are usually visualized using a map. They are developed for researchers and analysts to examine geographic areas in relation to crime. Hot spots are areas with a high occurrence of crime. These areas can be anywhere. They can be bars, malls, neighbourhoods, pretty much anywhere criminals target.

These hot spots are areas where crime is more likely to occur, and law enforcement agencies often focus their resources and efforts in these areas to prevent and reduce criminal activity. Crime hot spots can be identified through various methods such as analysing crime data, conducting community surveys, and mapping criminal activity patterns. By identifying crime hot spots, law enforcement can use targeted strategies to prevent and reduce crime in those areas. The goal of this project is to develop a Crime Hot Spot Prediction System that specifically focuses on women's safety using Machine Learning. The system will be designed to be user-friendly, accessible, and easy to use, with the aim of empowering women to take control of their safety. The Crime Hot Spot Prediction System using Machine Learning will be a valuable tool for law enforcement agencies, policymakers, and community organizations, enabling them to develop targeted strategies for preventing and reducing crime in hot spot areas.

1.2. PROBLEMS DEFINITION

Crime against women is a significant problem in India, with many incidents of violence, harassment, and discrimination reported each year. According to the National Crime Records Bureau (NCRB), there were over 4 lakh cases of crimes against women reported in India in 2019. These crimes include rape, sexual assault, domestic violence, kidnapping, and dowry deaths. Despite efforts by the Indian government to address the issue, including the introduction of laws such as the Protection of Women from Domestic Violence Act and the Criminal Law (Amendment) Act, 2013, there is still a long way to go to ensure women's safety and security in India. Women's safety is a major concern in many parts of the world, and crime hot spots are one of the major contributors to the feeling of insecurity among women. A crime hot spot prediction system can be developed to identify and predict areas where criminal activity is more likely to occur, allowing women to avoid those areas or take necessary precautions. One potential problem that may arise in the Crime Hot Spot Prediction System for Women Safety using data mining techniques is the issue of overfitting. Overfitting occurs when a model is trained too well on a specific set of data, resulting in poor performance when applied to new data. In this case, if the system is trained on a particular set of crime data, it may not perform well when applied to new data, leading to inaccurate predictions of crime hot spots. Another potential problem is the potential for bias in the data used to train the system. If the training data is biased or incomplete, it may result in inaccurate predictions, and the system may not perform well in identifying crime hot spots. It is crucial to ensure that the data used to train the system is representative of the actual crime patterns in the area and does not perpetuate biases or stereotypes. Another challenge may be the accuracy of the input data, which is essential for the system's effectiveness. The system relies on accurate and up-to-date crime data to identify crime hot spots. However, if the crime data is incomplete or inaccurate, it may lead to inaccurate predictions and undermine the system's effectiveness. Finally, the system's effectiveness may also depend on the availability and accessibility of relevant data. Data on crimes against women may not be readily available or may be underreported in some areas, which could limit the system's effectiveness in identifying crime hot spots. It is crucial to ensure that the system has access to relevant and accurate data to ensure its effectiveness.

1.3. MACHINE LEARNING

Machine learning is a branch of AI. Other tools for reaching AI include rule-based engines, evolutionary algorithms, and Bayesian statistics. While many early AI programs, like IBM's Deep Blue, which defeated Garry Kasparov in chess in 1997, were rule-based and dependent on human programming, machine learning is a tool through which computers have the ability to teach themselves, and set their own rules. In 2016, Google's DeepMind beat the world champion in Go by using machine learning–training itself on a large data set of expert moves.

1.3.1. Machine Learning Work

Machine Learning algorithm is trained using a training data set to create a model. When new input data is introduced to the ML algorithm, it makes a prediction on the basis of the model.

The prediction is evaluated for accuracy and if the accuracy is acceptable, the Machine Learning algorithm is deployed. If the accuracy is not acceptable, the Machine Learning algorithm is trained again and again with an augmented training data set. This is just a very high-level example as there are many factors and other steps involved.

1.3.2. Types of Machine Learning

- **Supervised Learning: More Control, Less Bias – Train Me!**

Supervised machine learning algorithms apply what has been learned in the past to new data using labelled examples to predict future events. By analysing a known training dataset, the learning algorithm produces an inferred function to predict output values. The system can provide targets for any new input after sufficient training. It can also compare its output with the correct, intended output to find errors and modify the model accordingly.

- **Unsupervised Learning: Speed and Scale - I am self-sufficient in learning**

Unsupervised machine learning algorithms are used when the information used to train is neither classified nor labelled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabelled data. At no point does the system know the correct output with certainty. Instead, it draws inferences from datasets as to what the output should be.

- **Reinforcement Learning: Rewards Outcomes -** My life My rules! (Hit & Trial)

Reinforcement machine learning algorithms are a learning method that interacts with its environment by producing actions and discovering errors or rewards. The most relevant characteristics of reinforcement learning are trial and error search and delayed reward. This method allows machines and software agents to automatically determine the ideal behaviour within a specific context to maximize its performance. Simple reward feedback — known as the reinforcement signal — is required for the agent to learn which action is best.

1.3.3. Decision Tree Algorithm

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too. The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data (training data). In Decision Trees, for predicting a class label for a record we start from the root of the tree. Compare the values of the root attribute with the record's attribute. Decision trees classify the examples by sorting them down the tree from the root to some leaf/terminal node, with the leaf/terminal node providing the classification of the example. Each node in the tree acts as a test case for some attribute, and each edge descending from the node corresponds to the possible answers to the test case. This process is recursive in nature and is repeated for every subtree rooted at the new node.

Important Terminology related to Decision Trees

- **Root Node:** It represents the entire population or sample and this further gets divided into two or more homogeneous sets.
- **Splitting:** It is a process of dividing a node into two or more sub-nodes.
- **Decision Node:** When a sub-node splits into further sub-nodes, then it is called the decision node.
- **Leaf / Terminal Node:** Nodes do not split is called Leaf or Terminal node.
- **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.
- **Branch / Sub-Tree:** A subsection of the entire tree is called branch or sub-tree.
- **Parent and Child Node:** A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.

Types of Decision Trees

Types of decision trees are based on the type of target variable we have. It can be of two types:

- **Categorical Variable Decision Tree:** Decision Tree which has a categorical target variable then it called a **Categorical variable decision tree.**
- **Continuous Variable Decision Tree:** Decision Tree has a continuous target variable then it is called **Continuous Variable Decision Tree.**

CHAPTER 2
LITERATURE SURVEY

2.1 TITLE: A Parallel Corpus-Based Approach to the Crime Event Extraction for Low-Resource Languages.

AUTHORS: Nina Khairova; Orken Mamyrbayev

OBJECTIVE: To extract crime-related events from web news articles in low-resource languages for monitoring, analysis, and comparison of police or criminal activities.

METHODOLOGY: Closed-domain event extraction methodology utilizing parallel corpora.

ALGORITHM: Enhanced pattern-based method using multilingual synonyms dictionary and logic-linguistic equations. Cross-lingual crime-related event extraction transfer technique.

MERITS: High F1-measure (82%) for source language. - Extraction of TRANSFER, CRIME, and POLICE events and seven subtypes.

DEMERITS: Limited to specific event types and subtypes.

2.2 TITLE: CrowdSPaFE: A Crowd-Sourced Multimodal Recommendation System for Urban Route Safety.

AUTHORS: Syeed Abrar Zaoad; Md. Mamun-Or-Rashid

OBJECTIVE : Introduce a recommendation system for urban route safety that adapts to dynamic crime reports, real-time user feedback, and prioritizes safety in navigation services.

METHODOLOGY: Population-based algorithm (CrowdSPaFE) incorporating crowd-sourced and historical data, considering the most recent feedback, and addressing navigation in areas with minimal feedback.

ALGORITHM: SPaFE algorithm, crowd-sourced data, historical data, crime reports, user feedback.

MERITS: Adapts to dynamic crime reports. - Prioritizes the most recent feedback. - Considers navigation in areas with negligible feedback. - Balances distance and safety. - Outperforms state-of-the-art algorithms.

DEMERITS: Algorithm performance may vary depending on the dataset.

2.3 TITLE: Unsupervised Domain Adaptation for Crime Risk Prediction Across Cities..

AUTHORS: Binbin Zhou; Longbiao Chen

OBJECTIVE: Develop an unsupervised domain adaptation model for crime risk prediction across cities, addressing the challenge of inconsistent context data between cities.

METHODOLOGY: Unsupervised domain adaptation model (UDAC) that identifies similar source the target city, and uses a dense convolutional network to learn high-level representations and domain-invariant features.

ALGORITHM: UDAC, dense convolutional network, unsupervised domain adaptation.

MERITS: Utilizes a dense convolutional network for accurate crime risk prediction.

DEMERITS: Potential limitations related to generalizability to diverse cities.

2.4 TITLE: A Holistic Framework for Crime Prevention, Response, and Analysis With Emphasis on Women Safety Using Technology and Societal Participation.

AUTHORS: Meetha V. Shenoy; Smriti Sridhar

OBJECTIVE: Develop a holistic system for women's safety in smart cities by integrating crime analysis and mapping, crime prevention, and emergency response with a focus on societal participation.

METHODOLOGY: Utilizes Geographic Information System (GIS) for hotspot identification and crime pattern analysis. Incorporates data generated from mobile applications and wearable devices

ALGORITHM: Not specific algorithm mentioned in the provided information.

MERITS: covering crime analysis, prevention, and emergency response.

DEMERITS: Applicability and scalability to different regions not discussed.

2.5 TITLE: Crime Spatiotemporal Prediction With Fused Objective Function in Time Delay Neural Network.

AUTHORS: Anahita Ghazvini; Siti Norul

OBJECTIVE: Focus on predicting upcoming serial crime distance, time, and suspect biographies, with an emphasis on enhanced activation functions for the Time Delay Neural Network (TDNN).

METHODOLOGY: Utilizes Time Delay Neural Network (TDNN) with an enhanced NARX model (eNARX) featuring fused activation functions of hyperbolic tangent (Tansig) and Radial Basis Function (RBF) to improve prediction accuracy.

ALGORITHM: Time Delay Neural Network (TDNN), Back Propagation Through Time (BPTT), Nonlinear Autoregressive with Exogenous Input (NARX).

MERITS: Introduction of enhanced NARX (eNARX) model with fused activation functions. Higher prediction accuracy compared to other techniques.

DEMERITS: Limited discussion on real-world implementation and practical challenges.

CHAPTER 3
SYSTEM ANALYSIS

3.1. EXISTING SYSTEM

There are several existing systems for Crime Hot Spot Prediction, but very few systems specifically focus on predicting crimes against women. Some of the existing systems are:

- **Manual Policing System**

Manual system for Crime Hot Spot Prediction System for Women Safety involves the use of traditional policing methods to identify crime hot spots. This typically involves analysing historical crime data, conducting interviews with witnesses and victims, and collecting data on the physical environment of the area.

- **Law Enforcement Agencies**

Law enforcement agencies use this information to determine the likelihood of criminal activity in a particular area and allocate resources accordingly. For example, they may increase patrols in high crime areas, install surveillance cameras, or provide additional lighting in poorly lit areas.

- **Clustering**

This technique is used to group similar crime incidents together based on their location, time, and type of crime. By identifying patterns in these groups, law enforcement agencies can identify crime hot spots and allocate resources accordingly.

- **Association rule mining**

This technique is used to identify correlations between different variables, such as the time of day and the type of crime. By identifying these correlations, law enforcement agencies can identify potential crime hot spots and take proactive measures to prevent criminal activity.

- **Logistic Regression:**

This algorithm is used to predict the probability of criminal activity based on a set of input variables. It is particularly useful for Crime Hot Spot Prediction System for Women Safety because it can handle both categorical and continuous data.

- **Random Forest**

This algorithm is used to develop a model that can predict the likelihood of criminal activity based on multiple decision trees. It is particularly useful for Crime Hot Spot Prediction System for Women Safety because it can handle non-linear data and identify complex patterns in crime data.

- **Crime Mapping System**

This system uses geographic information system (GIS) technology to map crime data and identify crime hot spots. The system has been used by law enforcement agencies in many cities around the world.

3.1.1. DISADVANTAGES

- Limits to identify crime hot spots accurately
- No Crime Hot Spot Map Visualizer
- Manual systems involve the labour-intensive process of collecting and analysing crime data.
- Many existing systems are resource-intensive and require a lot of time and resources to analyse crime data and identify crime hot spots.
- Many existing systems rely heavily on historical data to identify crime hot spots.
- May not be able to identify crime hot spots that are specifically related to crimes against women.

3.2. PROPOSED SYSTEM

The primary aim of this project is to develop a robust and user-centric system focused on predicting and localizing crime hot spots, specifically tailored for ensuring the safety of women. The system will utilize an Explainable Decision Tree (xDT) algorithm for transparent and interpretable crime prediction, and integrate Google Map API for precise geospatial localization.

- **Explainable Decision Tree (xDT)**

The system employs an xDT algorithm to ensure transparency in decision-making, providing clear insights into the factors influencing crime predictions. This interpretable model enhances user understanding and trust in the system.

- **Google Map API Integration**

Utilizing the Google Map API enables accurate geospatial visualization of predicted crime hot spots. The real-time mapping feature offers up-to-date information on safety concerns, contributing to precise crime localization and dynamic response strategies.

- **User-Friendly Interface**

The design prioritizes an intuitive web-based interface for accessibility. Emphasis is placed on ease of use and understanding, catering to a diverse user base. This ensures that users, including those with limited technological proficiency, can navigate the system effortlessly.

- **Real-time Crime Prediction**

Algorithms for real-time crime prediction empower law enforcement and individuals to proactively respond to emerging safety concerns. This feature aids in timely decision-making and crime prevention strategies.

- **Alert and Notification System**

An integrated alert system notifies users about potential safety concerns in their vicinity. The system allows customization of notification preferences, aligning with individual comfort levels and ensuring a user-centric approach to safety notifications. This proactive alert mechanism enhances overall user awareness and safety.

3.2.1. ADVANTAGES

- Prior knowledge of criminal activities in those locations.
- Interpretable predictions aid in building user trust.
- Accurate mapping of crime hot spots for proactive safety measures.
- User-friendly interface with Google Map integration.
- Predict the crime patterns of the states with respect to time.
- Cities must be made safer and more inclusive for women and everyone.
- Visualize the crime Hotspots Location in Map.
- Send alerts to users about nearby unsafe areas.
- Easy to use for all types of users.
- Helps police to plan better safety measures.
- Shows crime data clearly on the map.

CHATPER 4

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

- **Processor** : Intel Xeon or AMD Ryzen series processor
- **RAM** : Minimum 16GB RAM, recommended 32GB or higher for better performance
- **Storage** : SSD storage for faster data access
- **Network** : Gigabit Ethernet for network connectivity

4.2. SOFTWARE REQUIREMENTS

- **Programming** : Python 3.7.4(64-bit) or (32-bit)
- **Framework** : Flask 1.1.
- **Database** : MySQL 5.
- **Web Server** : Wampserver 2i
- **Packages** : Pandas, Sikit Learn, Numpy, matplotlib, seaborn
- **Map** : Google Map API

CHAPTER 5
SOFTWARE DESCRIPTION

5.1. PYTHON 3.8

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985-1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages. Python is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc. The biggest strength of Python is huge collection of standard libraries which can be used for the following:

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)

- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Scientific computing
- Text processing and many more.

Pandas

pandas are a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language. pandas are a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python.

Pandas is mainly used for data analysis and associated manipulation of tabular data in Data frames. Pandas allows importing data from various file formats such as comma-separated values, JSON, Parquet, SQL database tables or queries, and Microsoft Excel. Pandas allows various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning, and data wrangling features. The development of pandas introduced into Python many comparable features of working with Data frames that were established in the R programming language. The panda’s library is built upon another library NumPy, which is oriented to efficiently working with arrays instead of the features of working on Data frames.

NumPy

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed.

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

Scikit Learn

Scikit-learn is a Python module for machine learning built on top of SciPy and is distributed under the 3-Clause BSD license.

Scikit-learn (formerly scikits. learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

5.2. MYSQL 5

MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by Oracle Company. MySQL database that provides for how to manage database and to manipulate data with the help of various SQL queries. These queries are: insert records, update records, delete records, select records, create tables, drop tables, etc. There are also given MySQL interview questions to help you better understand the MySQL database.

MySQL is currently the most popular database management system software used for managing the relational database. It is open-source database software, which is supported by Oracle Company. It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise applications. It is developed, marketed, and supported by MySQL AB, a Swedish company, and written in C programming language and C++ programming language. The official pronunciation of MySQL is not the My Sequel; it is My Ess Que Ell. However, you can pronounce it in your way. Many small and big companies use MySQL. MySQL supports many Operating Systems like Windows, Linux, MacOS, etc. with C, C++, and Java languages.

5.3. WAMPSERVER

WampServer is a Windows web development environment. It allows you to create web applications with Apache2, PHP and a MySQL database. Alongside, PhpMyAdmin allows you to manage easily your database.

5.4. BOOTSTRAP 4

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.

Easy to use: Anybody with just basic knowledge of HTML and CSS can start using Bootstrap

Responsive features: Bootstrap's responsive CSS adjusts to phones, tablets, and desktops

Mobile-first approach: In Bootstrap, mobile-first styles are part of the core framework

Browser compatibility: Bootstrap 4 is compatible with all modern browsers (Chrome, Firefox, Internet Explorer 10+, Edge, Safari, and Opera)

5.5. FLASK

Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

Flask is often referred to as a micro framework. It aims to keep the core of an application simple yet extensible. Flask does not have built-in abstraction layer for database handling, nor does it have formed a validation support. Instead, Flask supports the extensions to add such functionality to the application. Although Flask is rather young compared to most Python frameworks, it holds a great promise and has already gained popularity among Python web developers. Let's take a closer look into Flask, so-called "micro" framework for Python. Flask is part of the categories of the micro-framework. Micro-framework are normally framework with little to no dependencies to external libraries. This has pros and cons. Pros would be that the framework is light, there are little dependency to update and watch for security bugs, cons is that some time you will have to do more work by yourself or increase yourself the list of dependencies by adding plugins.

5.6. Google Map API

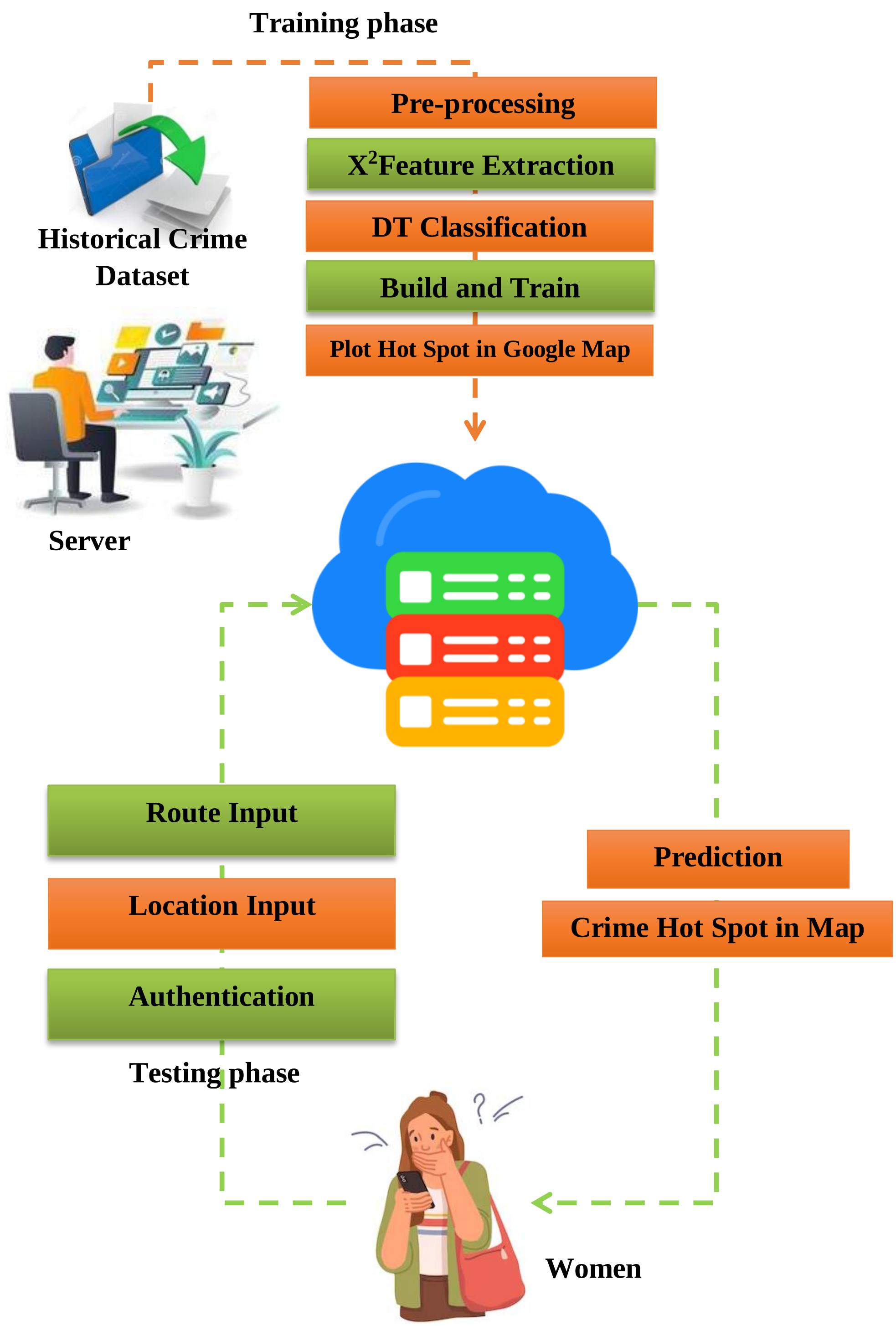
Google Maps Platform is a set of APIs and SDKs for retrieving location-based data from Google and embedding Google Maps imagery into mobile apps and web pages. The Maps API returns helpful data about places and locations. It is called by javascript. To use the Maps, Embed API you must have an API key. The API includes language localization for over 50 languages, region localization and geocoding, and has mechanisms for enterprise developers who want to utilize the Google Maps API within an intranet.

The Street View Static API lets you embed a static (non-interactive) Street View panorama or thumbnail into your web page, without the use of JavaScript. The Directions API is a service that calculates directions between locations. You can search for directions for several modes of transportation, including transit, driving, walking, or cycling. The Roads API identifies the roads a vehicle was traveling along and provides additional metadata about those roads, such as speed limits. The Maps Static API service creates your map based on URL parameters sent through a standard HTTP request and returns the map as an image you can display on your web page. The API HTTP services can be accessed over a secure (HTTPS) connection by Google Maps API Premier customers.

CHAPTER 6

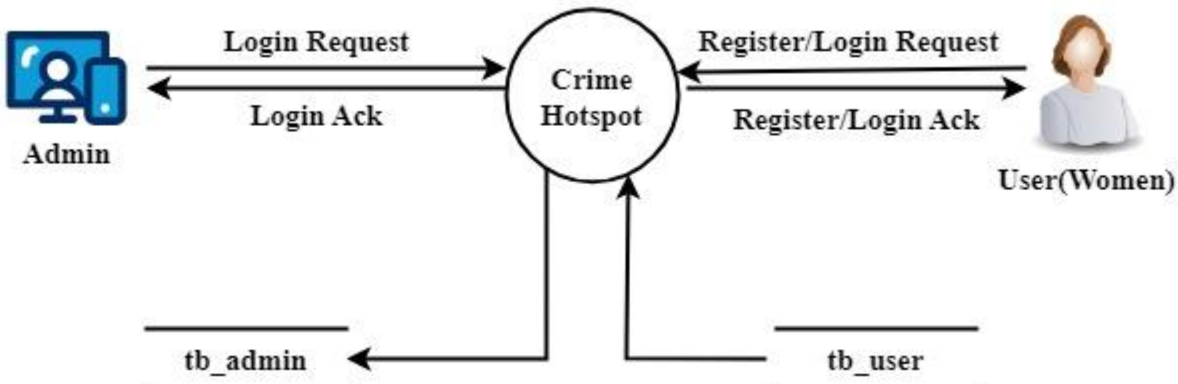
SYSTEM DESIGN

6.1. SYSTEM ARCHITECTURE

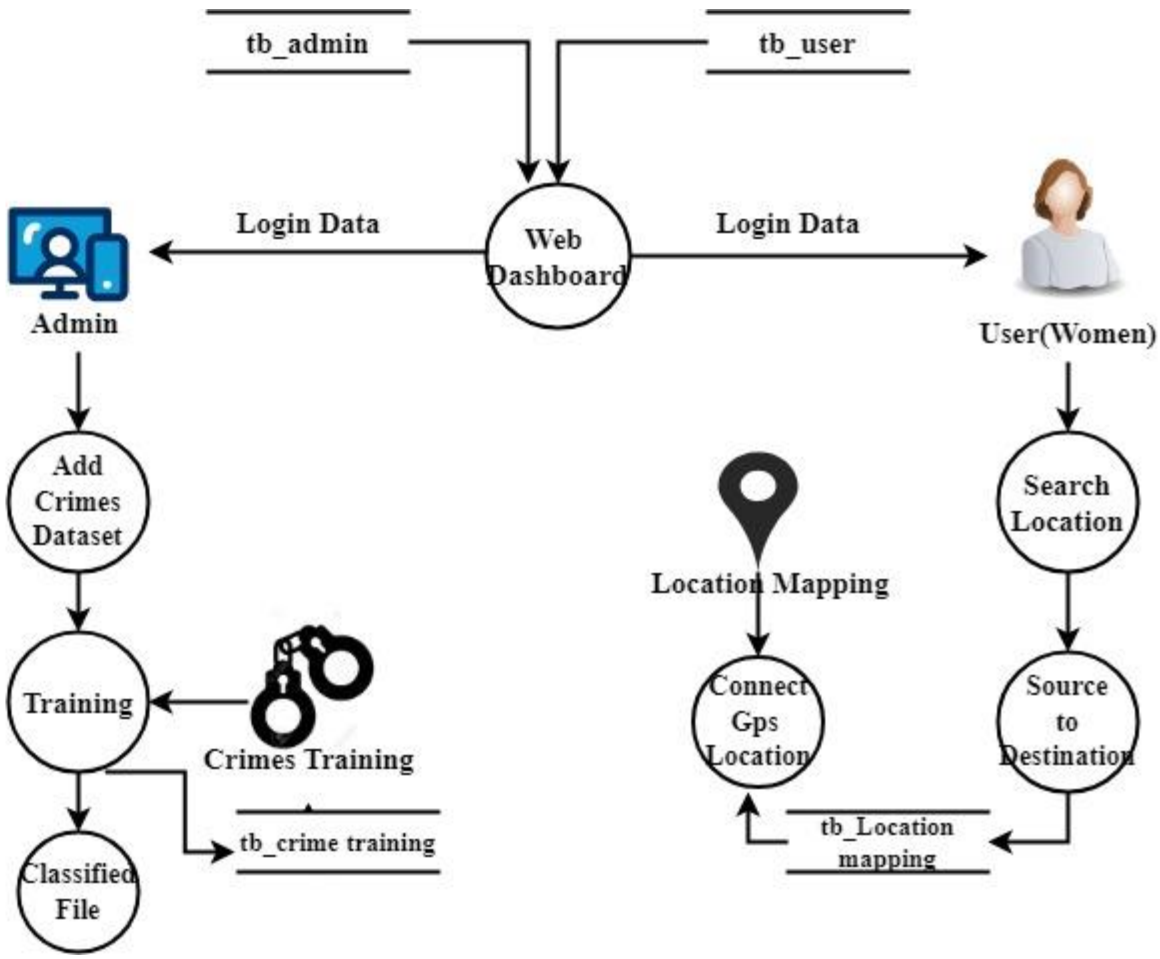


6.2. DATA FLOW DIAGRAM

LEVEL 0

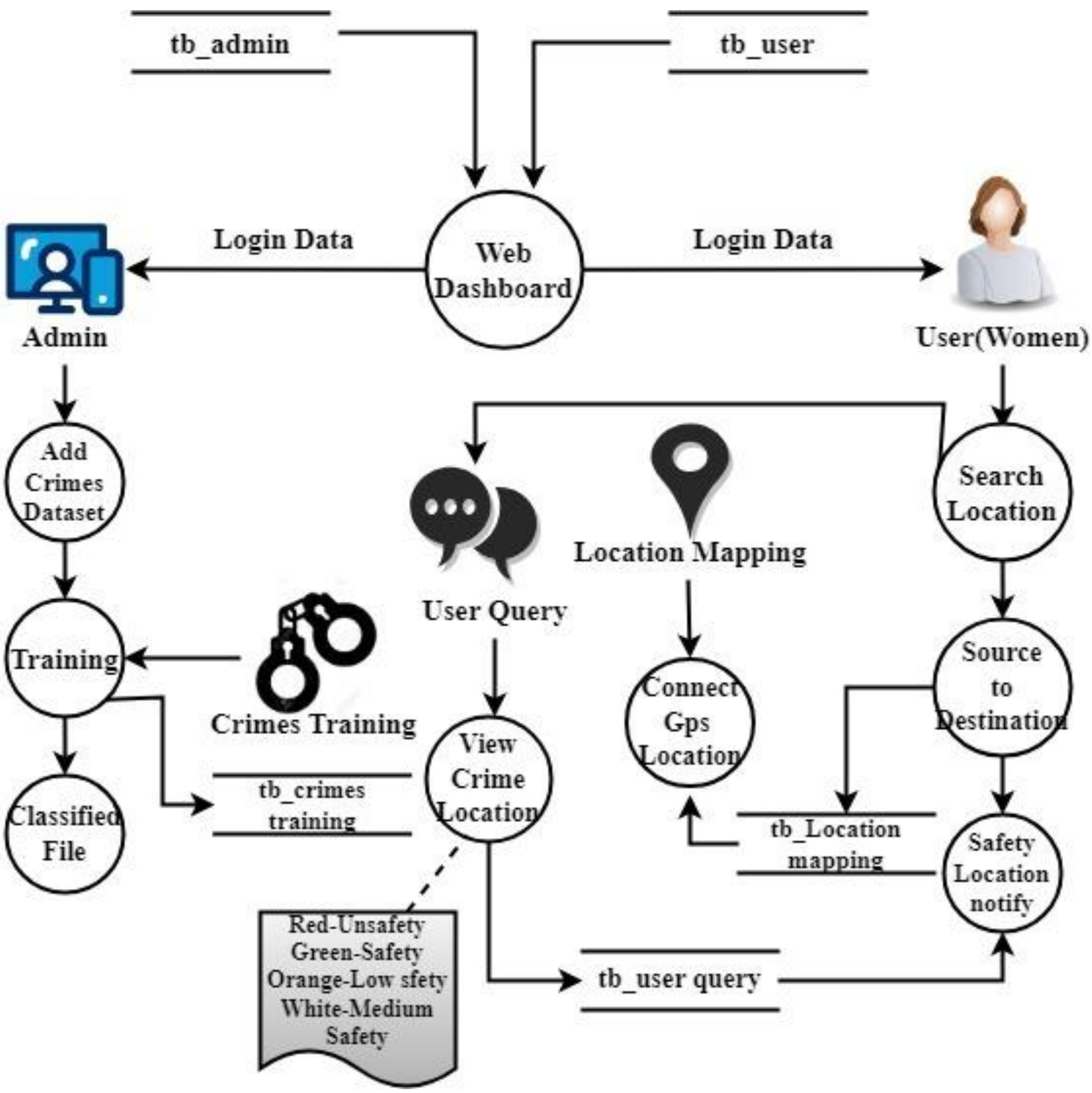


LEVEL 1



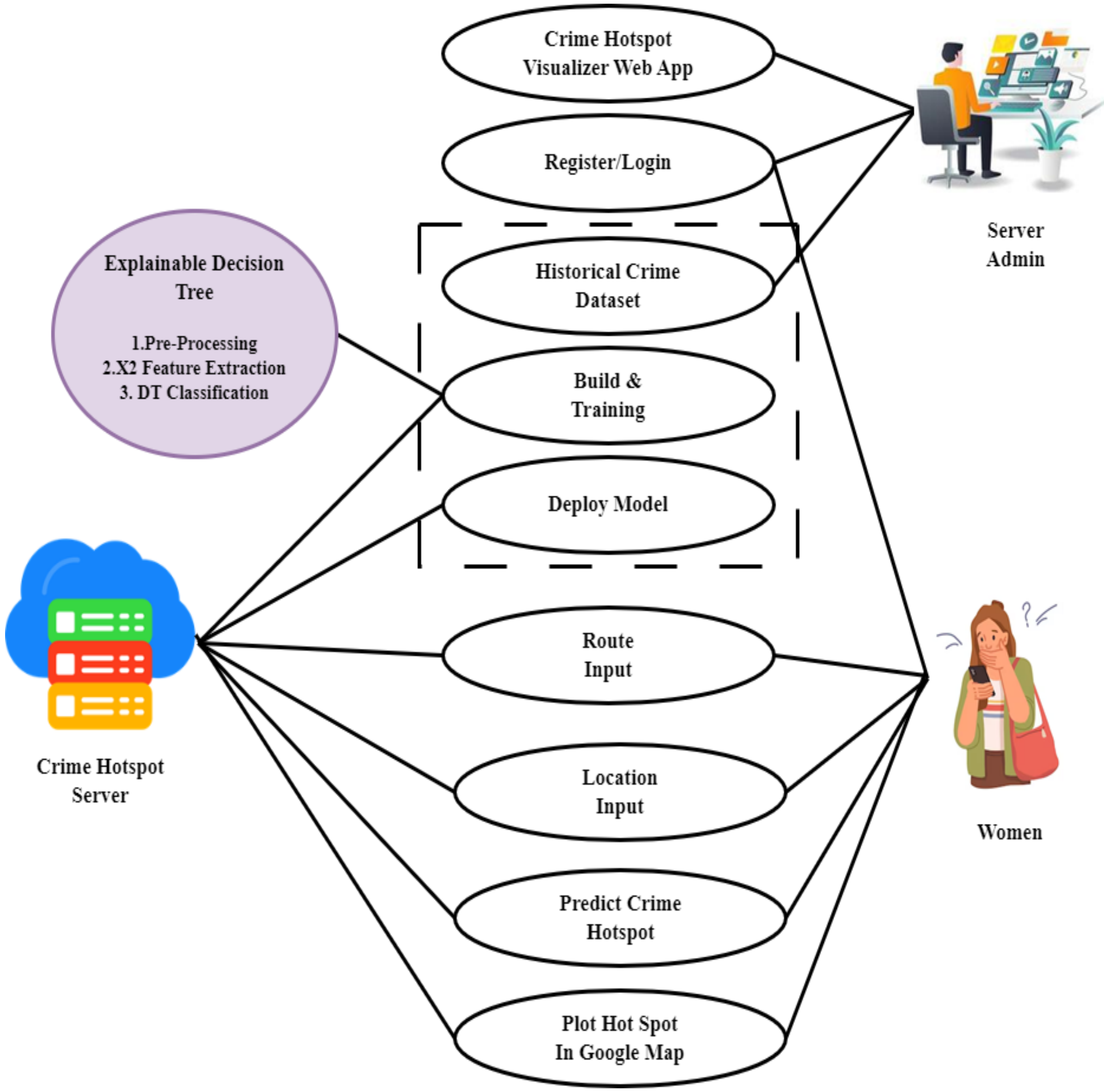
tb_admin
tb_user
Login Data
Web Dashboard
Login Data
Admin
User(Women)
Add Crimes Dataset
Search Location
Location Mapping
Training
Crimes Training
Connect Gps Location
Source to Destination
tb_crime training
tb_Location mapping
Classified File

LEVEL 2

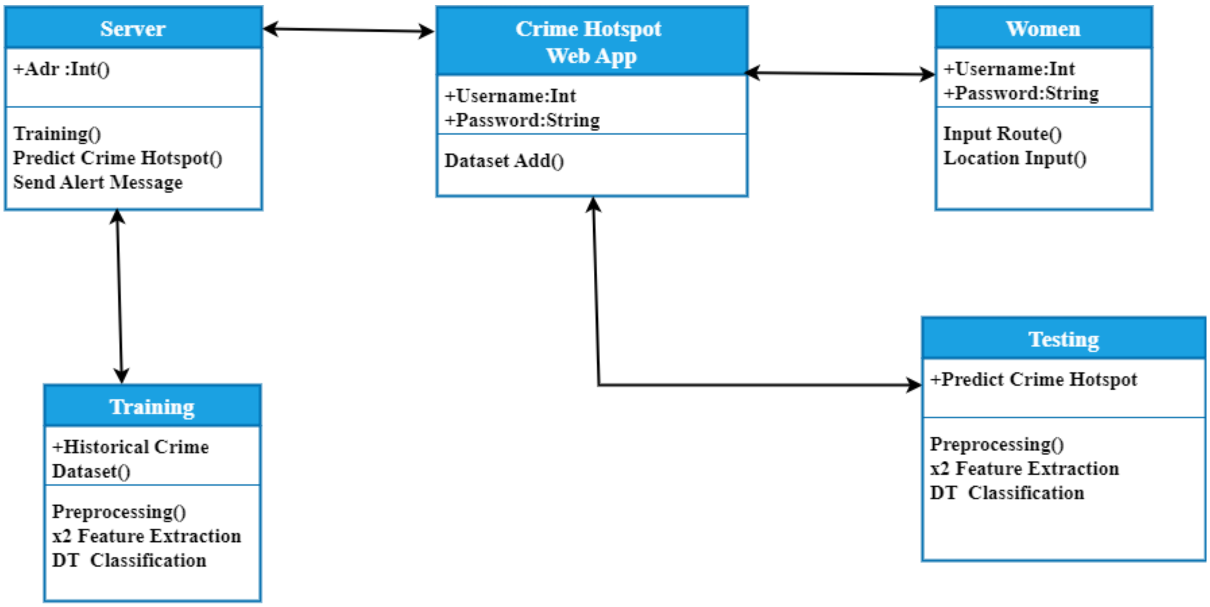

tb_admin
tb_user
Login Data
Web Dashboard
Login Data
Admin
User(Women)
Add Crimes Dataset
Search Location
User Query
Location Mapping
Training
Crimes Training
Source to Destination
Connect Gps Location
View Crime Location
tb_crimes training
Classified File
tb_Location mapping
Safety Location notify
Red-Unsafety
Green-Safety
Orange-Low sfety
White-Medium Safety
tb_user query

6.3. UML DIAGRAM

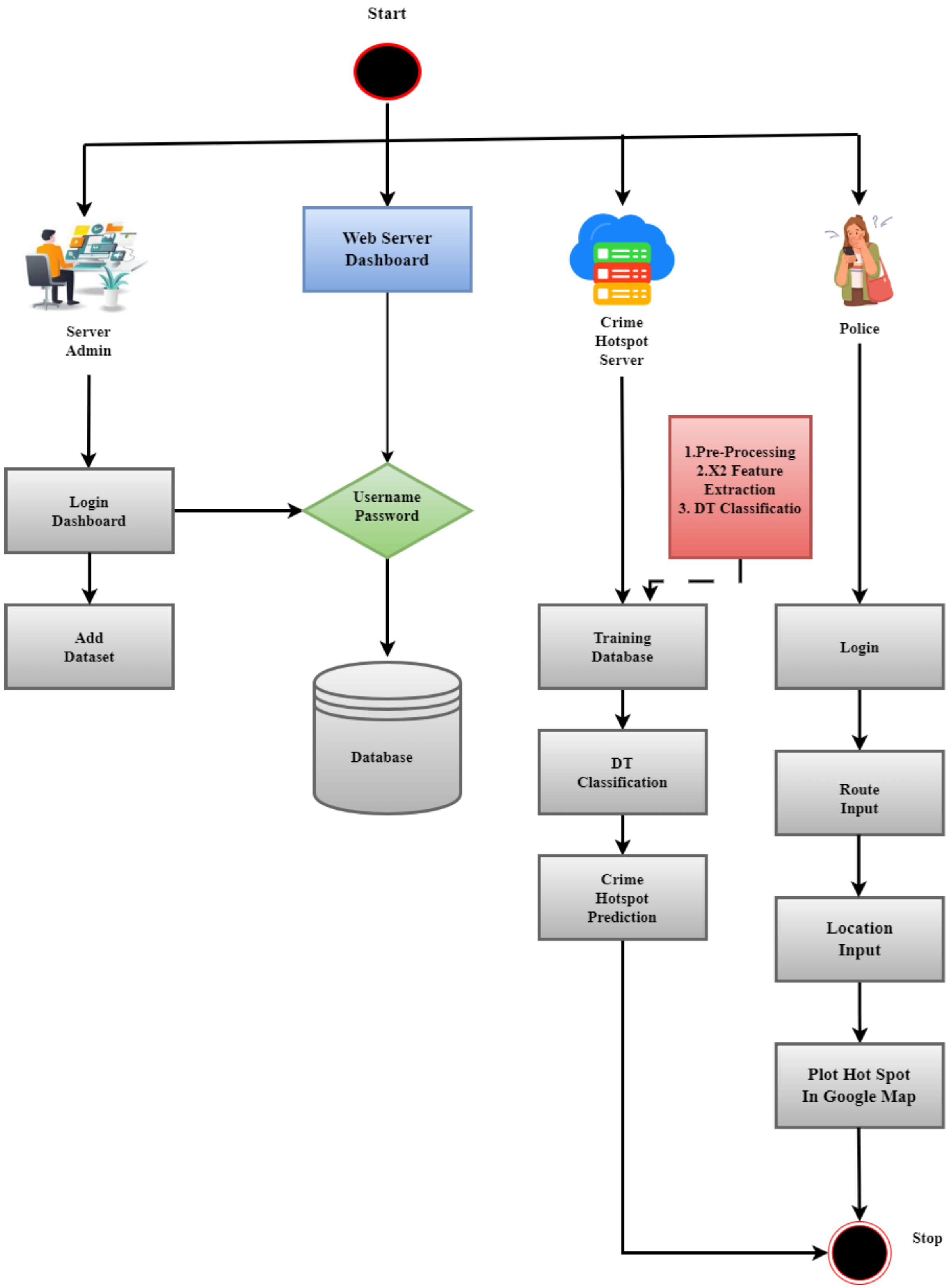
6.3.1. USE CASE



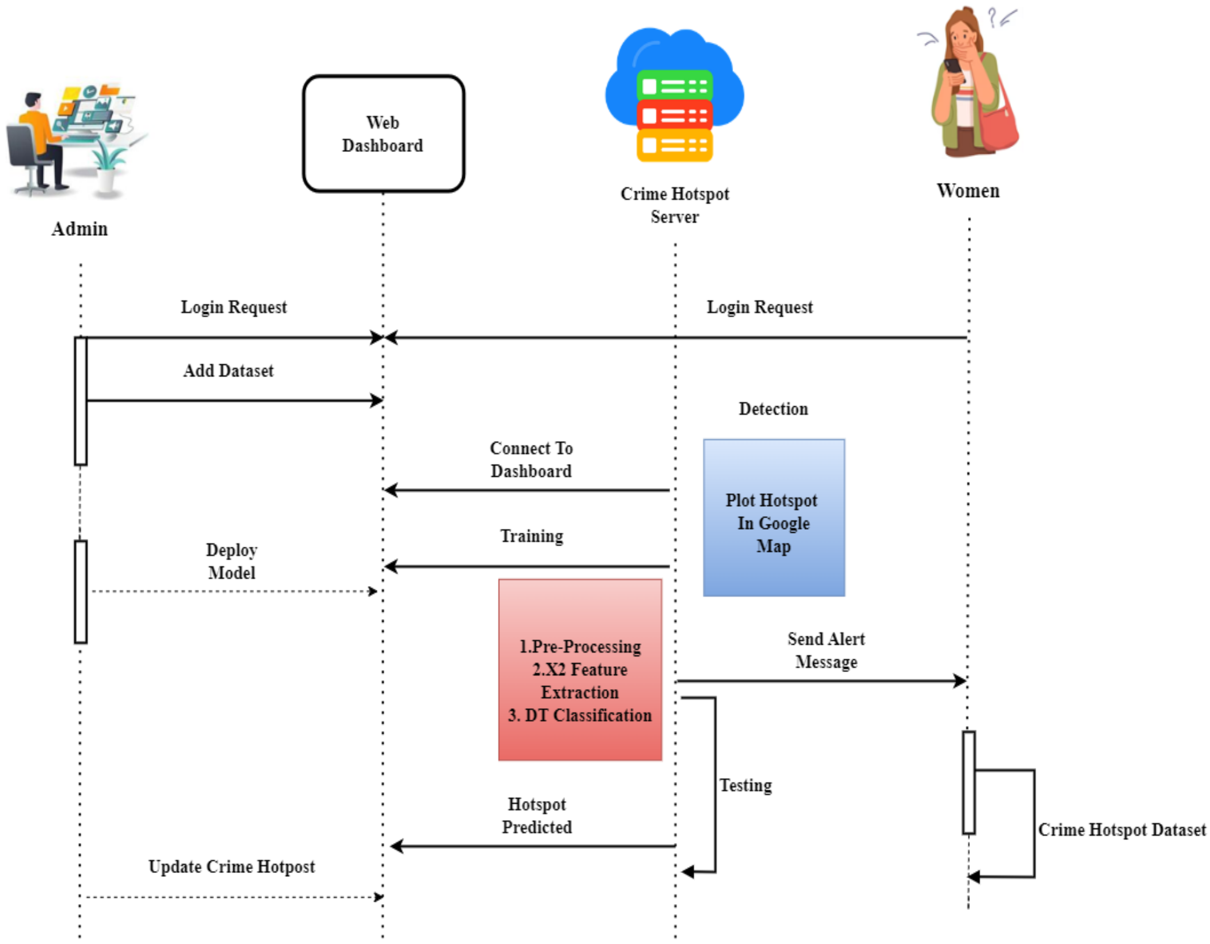
6.3.2. CLASS DIAGRAM



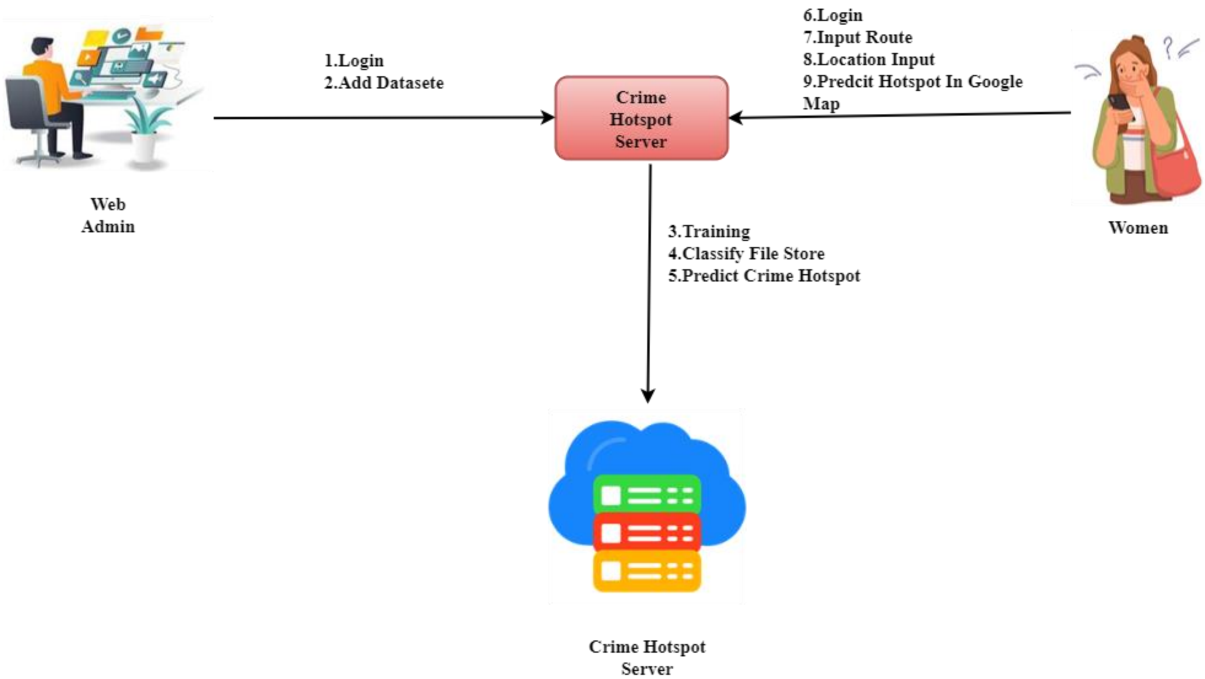
6.3.3. ACTIVITY DIAGRAM



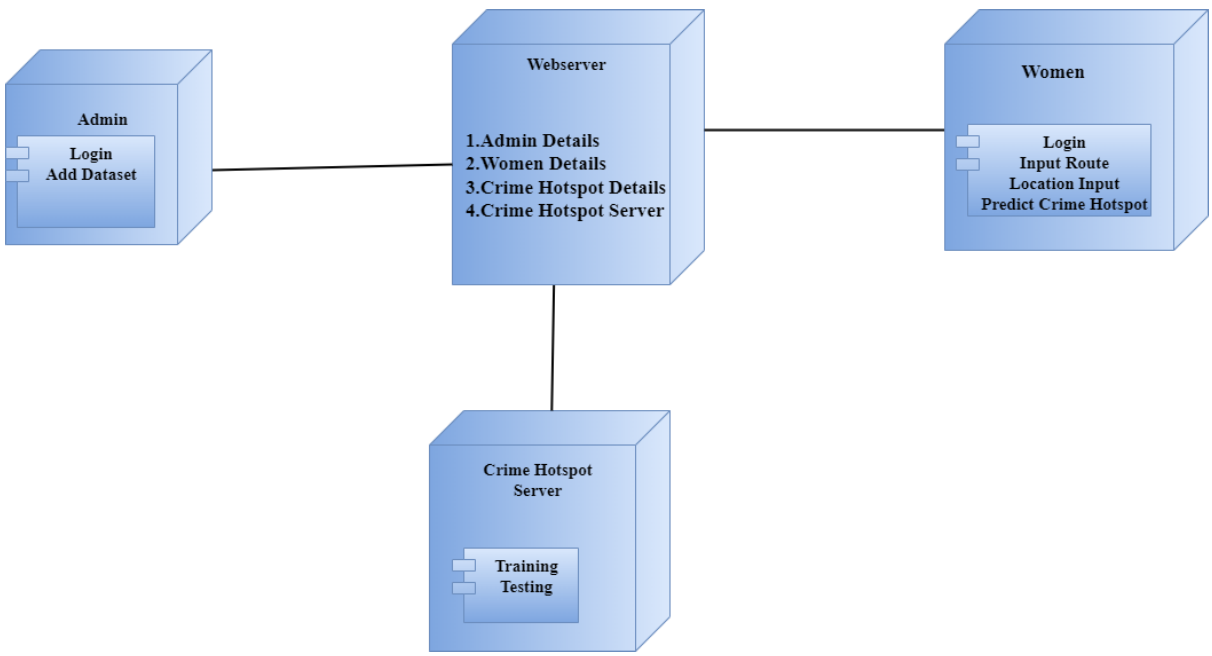
6.3.4. SEQUENCE DIAGRAM



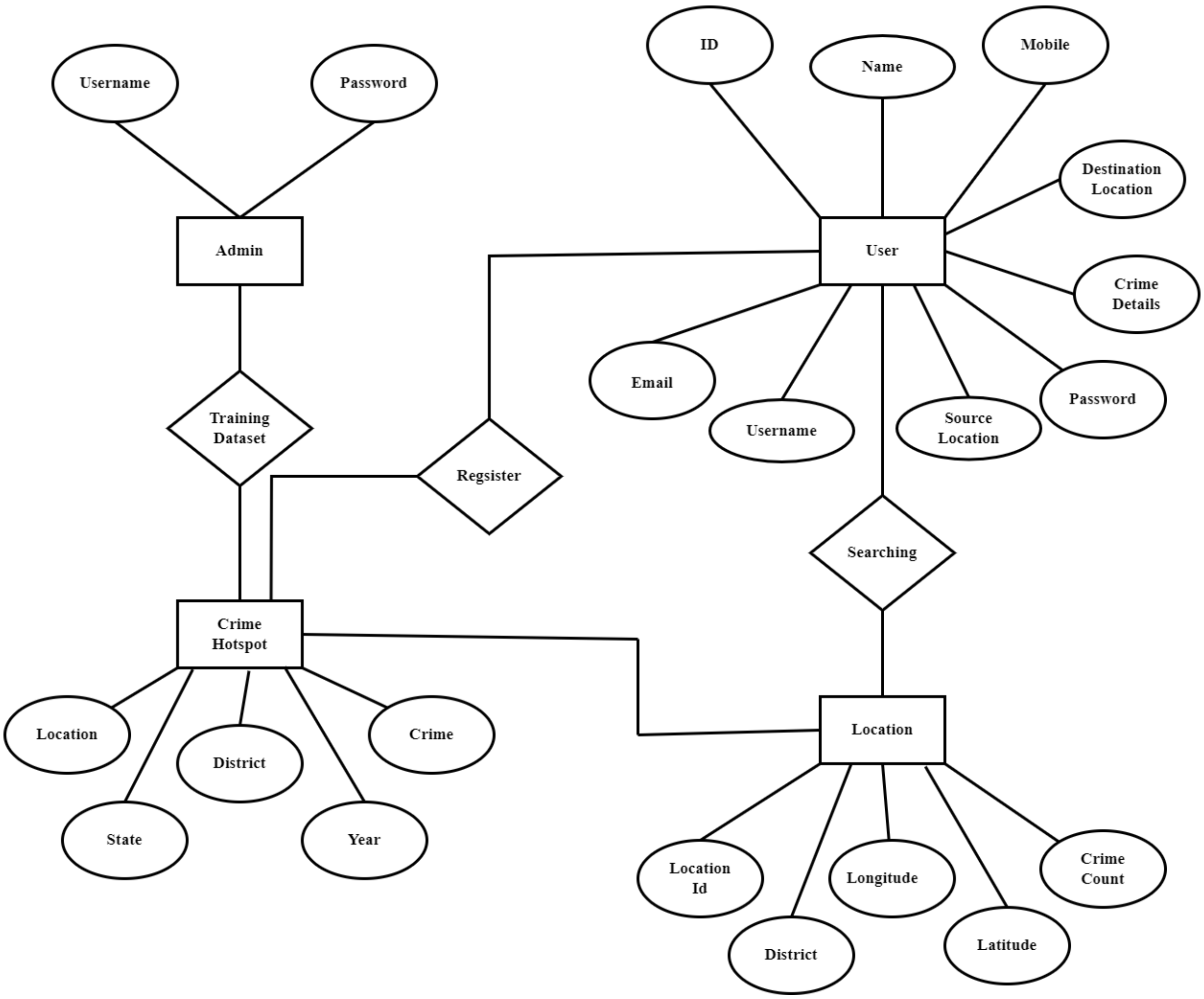
6.3.5. COLLABORATION DIAGRAM



6.3.6. DEPLOYMENT DIAGRAM



6.4. ER DIAGRAM



CHAPTER 7
SYSTEM TESTING

7.1. SOFTWARE TESTING

Testing is an essential part of the software development life cycle, and it helps to ensure that the system is working as intended and meets the specified requirements. In the case of the web-based crime hot spot prediction and localization system for women safety using xDT and Google Map API, the following types of testing can be performed:

7.1.1 Types of testing

- **Unit Testing:** This type of testing focuses on testing individual software components or modules. It ensures that each module works as expected and meets the requirements. In the case of this system, each module, such as the Crime Hotspot Classifier and Crime Hotspot Finder Web App, can be tested individually.
- **Integration Testing:** This type of testing is used to test how different modules work together when integrated. This ensures that the system functions as a whole and all components are integrated and working correctly. For the Crime Hot Spot Prediction and Localization System, integration testing can be conducted to ensure that all the modules work together as intended.
- **Performance Testing:** This type of testing is done to verify the system's ability to perform under various workloads. It involves testing the system's speed, scalability, and stability under various load conditions. Performance testing for this system would involve testing the system's response time and its ability to handle multiple user requests simultaneously.

- **User Acceptance Testing (UAT):** This type of testing involves the end-users testing the system in a simulated production environment. It is done to ensure that the system meets the end-users' expectations and is ready for deployment.

7.2. TEST CASE

1. **Test case for Crime Hotspot Classifier using xDT**
    - Test input: A dataset of crime incidents in a city with the corresponding features (location, time, type of crime, etc.)
    - Test output: xDT classifier accuracy score, confusion matrix, and classification report
2. **Test case for Crime Hotspot Pre-processing**
    - Test input: Raw crime incident data in various formats (CSV, Excel, JSON, etc.)
    - Test output: Cleaned and pre-processed crime data in a structured format (e.g. MySQL database)
3. **Test case for Crime Hotspot Prediction using xDT**
    - Test input: A location on the map where a woman is planning to go
    - Test output: A heat map of predicted crime hotspots around that location with different levels of intensity and colours
4. **Test case for Crime Hotspot Visualization**
    - Test input: A heat map of crime hotspots with different levels of intensity and colours
    - Test output: A visual representation of the crime hotspots on a Google Map with different icons and colours

Test Cases and Results

1. **Test Case 1 - Input Validation**
    - Test the system by entering invalid input, such as alphabets or symbols in the location input field.
    - Expected Result: The system should display an error message prompting the user to enter a valid location.
2. **Test Case 2 - xDT Model Accuracy**
    - Test the system by entering a location and checking if the predicted crime hotspots match with the actual crime hotspots in the area.
    - Expected Result: The system should provide accurate predictions with a high level of precision and recall.
3. **Test Case 3 - Google Maps Integration**
    - Test the system by entering a location and checking if the map correctly displays the predicted crime hotspots.
    - Expected Result: The system should accurately display the predicted crime hotspots on the Google Map.
4. **Test Case 4 - User Feedback**
    - Test the system by allowing users to provide feedback on the accuracy of the predictions.
    - Expected Result: The system should use the user feedback to improve the accuracy of the predictions over time.
5. **Test Case 5 - Administrator Access**
    - Test the system by logging in as an administrator and checking if the administrator has access to the system's settings and user data.
    - Expected Result: The administrator should be able to access and modify the
    - system settings and user data as necessary.

7.3. TEST REPORT

Introduction

The purpose of this test report is to evaluate the performance of the Crime Hot Spot Prediction and Localization System for Women Safety using xDt and Google Map API developed for predicting and localizing crime hotspots to improve women's safety.

Test Environment

The testing of the system was carried out on a local server running on Windows 10 operating system. The web application was developed using Python Flask framework, and the data was stored in MySQL database. The testing was carried out using Google Chrome and Mozilla Firefox web browsers.

Test Results

All the test cases were executed successfully without any errors, and the system met the expected results. Therefore, it can be concluded that the system is functioning correctly and meets the requirements.

Conclusion

Crime Hot Spot Prediction and Localization System for Women Safety using xDT and Google Map API has been tested successfully and meets the expected results. The system can predict and localize crime hotspots accurately, provide recommendations for women's safety, and visualize the results on Google Map. The system can be deployed in real-time for enhancing women's safety.

CHAPTER 8
SYSTEM IMPLEMENTATION

8.1. PROJECT DESCRIPTION

The project is designed to enhance the safety of women in urban environments. Leveraging the Explainable Decision Tree (xDT) algorithm and Google Map API, the system provides real-time predictions of potential crime hotspots, empowering users to make informed decisions about their safety. The system features a user-friendly web application with modules such as the Dashboard, Login/Registration, and Location Search, offering a seamless experience for users to access crime prediction, localization, safety alerts, and crime reports. Targeting women, law enforcement agencies, and NGOs focused on women's safety, the system tailors its features to meet diverse user needs. The core of the system lies in the Crime Hotspot Training Model, which utilizes xDT to analyze historical crime data, pre-process raw data, select relevant features, and build/train a crime hotspot prediction model. Additionally, the Crime Hot Spot Predictor System takes user input to predict potential crime hotspots, while the Crime Hotspot Map Visualization module visualizes predicted hotspots on a map for user understanding. The Recommendation Module provides safety recommendations based on predicted crime hotspots, ensuring proactive safety measures. Overall, the Crime Hot Spot Prediction and Localization System is a powerful tool that combines cutting-edge technology with a user-centric approach to create a safer urban environment for women.

DATASET PREPARATION

This dataset contains complete information about various aspects of crimes that happened in India in 2019. There are many factors that can be analysed from this dataset. National Crime Records Bureau (NCRB), Govt of India has published this dataset on their website and also has shared on Open Govt Data Platform India portal under Govt. Open Data License - India. This dataset consists of the count of a variety of crimes agaist women in India in 10 years (2011-2021) in various states. I have tried to visualize perform EDA on this dataset to understand 1. how bad the crimes against women in South India really has been in this period of time 2. what type of crimes have been most common? Some crimes that are included are Rape, Kidnapping and Abduction, Dowry Deaths etc.

Total Rows - 12

Columns – 396

8.2. MODULES DESCRIPTION

1. Crime Hotspot Finder Web App

The Crime Hotspot Finder Web App Module is a key component of the Crime Hot Spot Prediction and Localization System for Women Safety using XDT and Google Map API. It is built using Python Flask and MySQL and provides a user-friendly interface for accessing the system's features.

1.1. Dashboard Module

This module provides an overview of the system's key features, including crime prediction and localization, safety alerts, and crime reports. Users can access various functions of the system from this dashboard, such as setting up their safety preferences and viewing recent crime reports.

1.2. Login/Registration Module

This module allows users to create an account and log in securely to the system. User credentials are stored in a MySQL database, which ensures the security of the user's personal information.

1.3. Location Search

The location search functionality allows users to search for crime hotspots in a particular location. Users can enter the name of a city, town, or address and the system will display the crime hotspots in that location.

2. End User

The end-users for the Web-based Crime Hot Spot Prediction and Localization System for Women Safety using XDT and Google Map API would be women who want to ensure their safety and avoid high-risk areas. Law enforcement agencies and NGOs working on women's safety can also use this system to identify high-risk areas and take preventive measures to reduce crime against women. Additionally, the system can be used by researchers and analysts to study crime patterns and trends in specific areas.

2.1. Women or General user

The Women End User module is a crucial part of the web-based Crime Hotspot Prediction System for Women Safety using XDT and Google Map API. This module provides a user-friendly interface for women to access the system and receive alerts and recommendations based on their location.

The Women End User module includes the following features:

- **Registration:** Women can register themselves on the system by providing their basic details such as name, email address, contact number, and location.
- **Login:** Once registered, women can log in to the system using their email address and password.
- **Location Input:** Women can input their current location or allow the system to access their location through GPS.

- **Hotspot Prediction:** Based on the input location, the system predicts the crime hotspot areas in the vicinity and displays them on a map.
- **Alert Notifications:** If a woman is in or near a predicted crime hotspot area, the system sends an alert notification to her registered email address and phone number, advising her to stay cautious.
- **Recommendation:** The system provides recommendations for safe routes and nearby safe locations based on the predicted crime hotspots.
- **Feedback:** Women can provide feedback on the system's predictions and recommendations, which will help to improve the accuracy of the system.

Overall, the Women End User module aims to provide women with a sense of security by predicting crime hotspots and providing recommendations for safer routes and locations.

2.2. Web Admin

The administrator module of the crime hotspot prediction system is designed for system administrators to manage and maintain the system. The module provides the following functionalities:

- **User Management**

The administrator can manage user accounts, including creating new accounts, modifying existing accounts, and deleting accounts.

- **Database Management**

The administrator can manage the system's database, including backing up and restoring the database, creating and modifying database tables, and managing data records.

- **System Configuration**

The administrator can configure the system's settings, including setting the threshold values for crime hotspot prediction, managing the XDT algorithm parameters, and configuring the Google Maps API settings.

3. Crime Hotspot Training Model

The Crime Hotspot Classifier using xDT module is a core component of the Crime Hot Spot Prediction and Localization System for Women Safety. This approach involves training a Explainable Decision Tree algorithm with historical crime data to identify patterns and predict future crime incidents in the area. It is responsible for predicting crime hotspots using the DT algorithm.

3.1. Data Set Annotation

The data collected must be comprehensive, accurate, and relevant to the area being analysed. Law enforcement agencies often collaborate with data scientists to gather data from various sources and ensure that it is properly collected, stored, and managed.

3.2. Pre-processing

It is responsible for processing and preparing raw crime data for analysis and prediction using machine learning algorithms. Once the data has been collected, it must be cleaned and pre-processed to remove irrelevant information and prepare it for analysis. Removing irrelevant or redundant information

- Handling missing or incorrect data
- Formatting the data into a consistent format
- Removing outliers or extreme values
- Normalizing the data to reduce the impact of different scales
- Encoding categorical variables

These steps help to prepare the data for analysis and improve the accuracy of the predictions made using data science.

3.3. Feature Selection

Feature selection involves identifying the most relevant features, such as location, time, and type of crime, that can be used to predict crime hotspots accurately. Feature engineering involves transforming and encoding the selected features into a suitable format that can be fed into machine learning algorithms. In feature selection, irrelevant features are eliminated and the most important or relevant features are applied to the network. Thus, if we supply all features to DT, some features may be noisy and if they are learned in the training process, they may degrade generalization of the network although the network will show good performance on the training data. That is why large number of features are also considered one of the main causes of over fitting. Thus, searching out optimal subset of features by eliminating noisy features can help DT to show good performance on both training and testing data. In this module, we use X^2 statistical model to eliminate irrelevant features. In the feature's elimination process, we compute X^2 statistics between each non-negative feature F_i and class i.e., y. The X^2 model performs X^2 test that measures dependence between the features and class. Hence, the model is capable of eliminating those features which are more likely to be independent of class. Because, these features can be regarded as irrelevant for classification.

3.4. DT – Build and Train Crime Hotspot Model

The Crime Hotspot Prediction using DT module is one of the primary components of the Crime Hot Spot Prediction and Localization System for Women Safety. It is responsible for predicting crime hotspots using the DT. The module takes pre-processed crime data as input, and uses the DT algorithm to learn patterns and relationships between crime incidents and their locations, time, and other relevant features. The DT algorithm is a supervised learning algorithm that can classify data into different categories by finding a hyperplane that maximally separates the different classes. The Crime Hotspot Prediction using DT module first splits the pre-processed crime data into training and testing datasets, and then trains the DT model using the training dataset. The DT algorithm uses a classification approach to predict the crime hotspots. The algorithm divides the crime data into two categories: positive and negative. The positive category includes areas where the crime rate is higher, while the negative category includes areas where the crime rate is lower.

4. Crime Hot Spot Predictor System

The algorithm takes as input various features, such as location, time, and type of crime, to make the prediction. Based on the input features, the algorithm classifies the area as a potential crime hotspot or a safe area. Location-based prediction of crime hot spots can be a useful addition to the existing crime hot spot prediction system. It can allow users to enter their current location or intended destination and receive information about potential crime hot spots in that area. This feature can be implemented using a combination of geolocation and data mining techniques.

5. Crime Hotspot Map Visualization

The Crime Hotspot Visualization module in the Crime Hot Spot Prediction and Localization System for Women Safety using DT and Google Map API is responsible for visualizing the predicted crime hotspots on a map. This module receives the predicted hotspots generated by the Crime Hotspot Prediction using DT module and plots them on a map using Google Map API. The module uses the latitude and longitude coordinates of the predicted hotspots to create markers on the map. These markers are then displayed with the help of the Google Maps JavaScript

API. The markers are color-coded based on the predicted crime intensity, with red markers indicating high crime intensity and green markers indicating low crime intensity. The module also provides an option for the user to filter the hotspots based on the crime type. The user can select a specific crime type from a dropdown menu, and only the hotspots related to that crime type will be displayed on the map. Overall, the Crime Hotspot Visualization module enhances the user's understanding of the predicted hotspots and provides an easy-to-use interface for exploring the data.

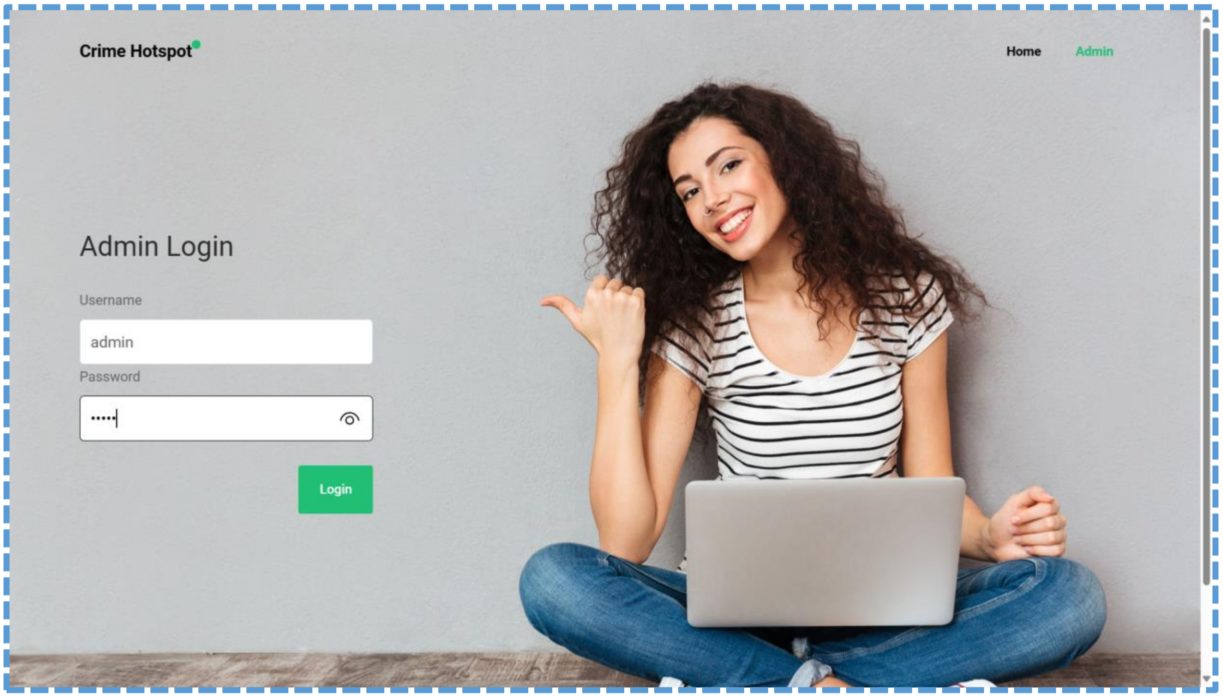
6. Recommendation Module

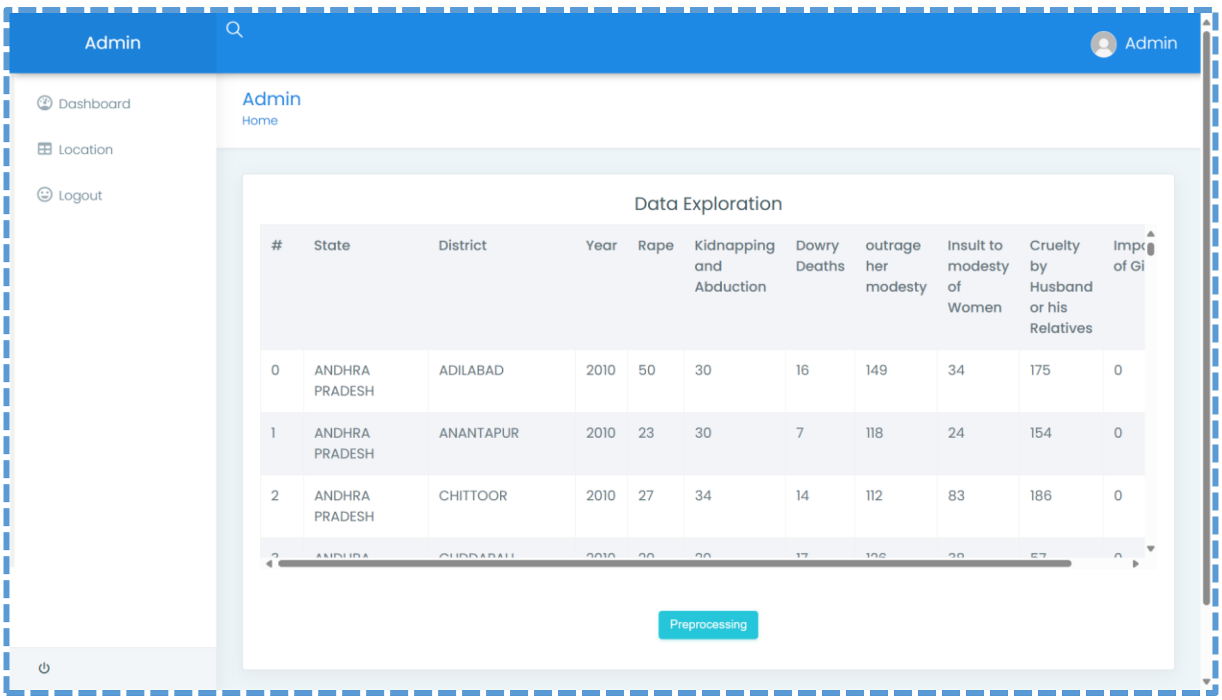
The Recommendation module in the Crime Hot Spot Prediction and Localization System for Women Safety using xDT and Google Map API is designed to provide safety recommendations to users based on the predicted crime hotspots in a particular area. This module takes into account the type of crimes that are most likely to occur in the predicted hotspots and suggests safety measures that can be taken to avoid such incidents. The recommendation module uses a rule-based approach to provide safety recommendations. These rules are based on the type of crime predicted in the hotspots and the best practices recommended by law enforcement agencies to prevent such crimes. For example, if the predicted hotspot is for theft, the recommendation module may suggest that users should avoid carrying expensive jewelry or electronics while travelling in that area. Similarly, if the predicted hotspot is for assault, the module may suggest that users should avoid travelling alone during late hours and to stay in well-lit areas. The recommendations provided by the module are displayed to the users in the form of pop-up messages on the Crime Hotspot Finder Web App. The recommendations are also stored in the system's database for future reference. The module also allows users to provide feedback on the recommendations provided, which can be used to improve the effectiveness of the recommendation engine in future versions of the system.

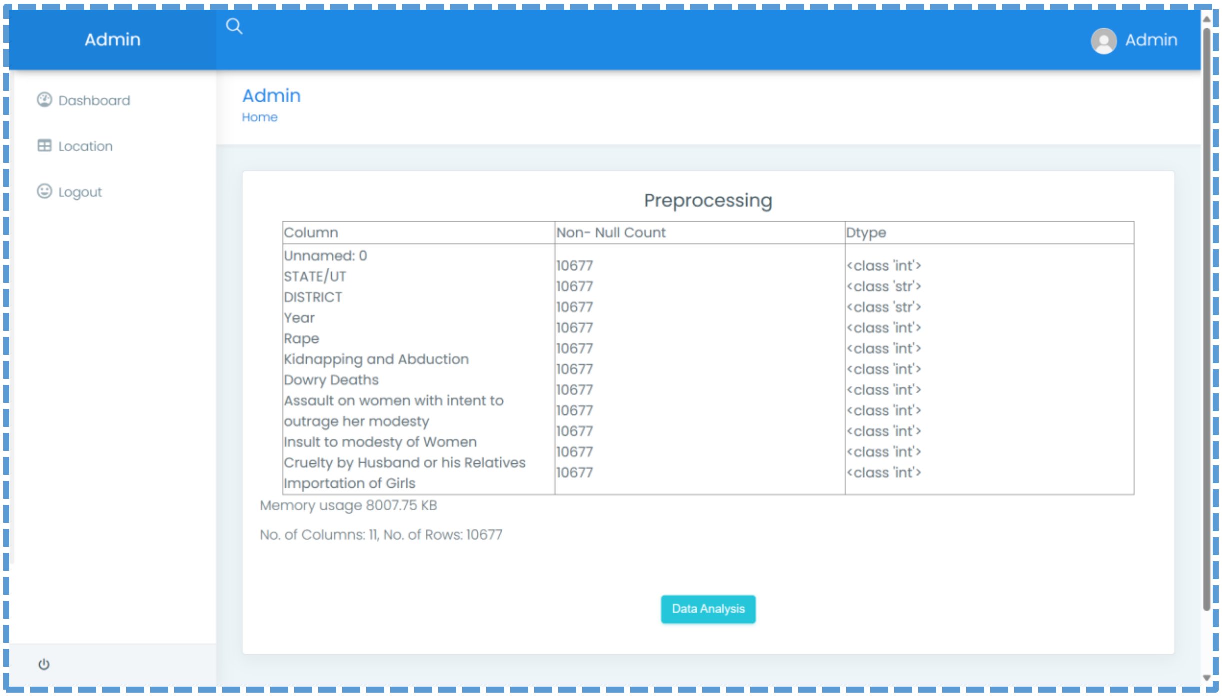
CHAPTER 9

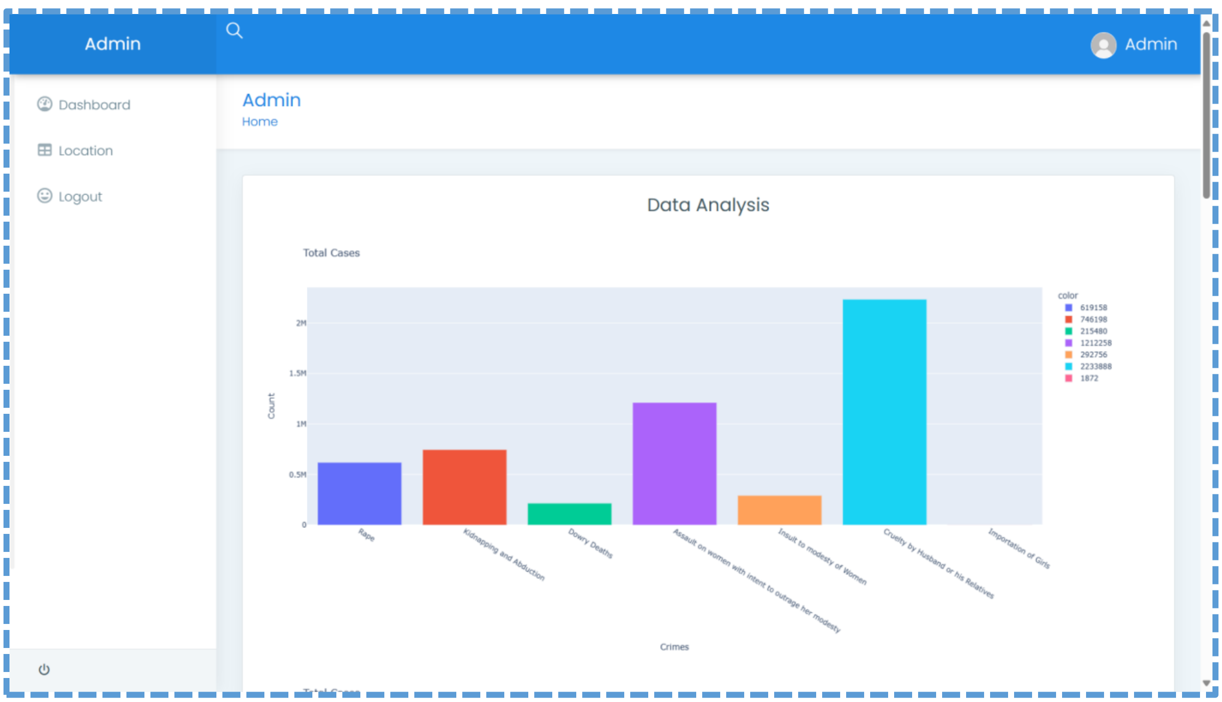
APPENDICES

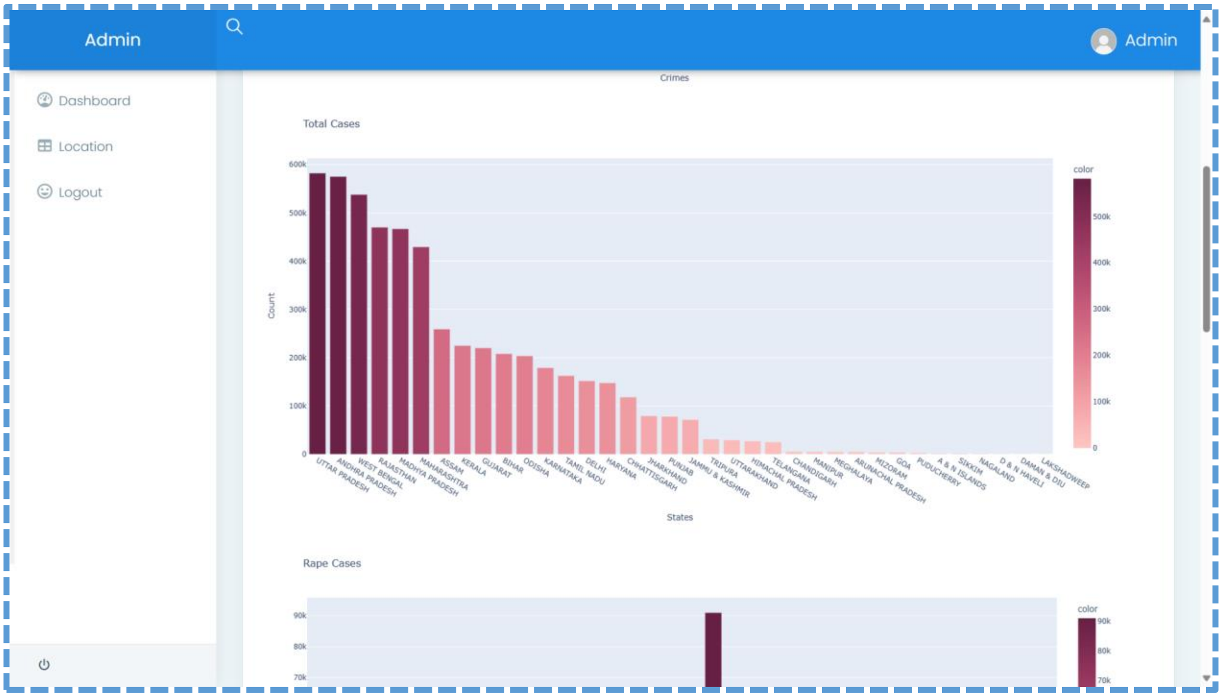
9.1. SCREENSHOTS



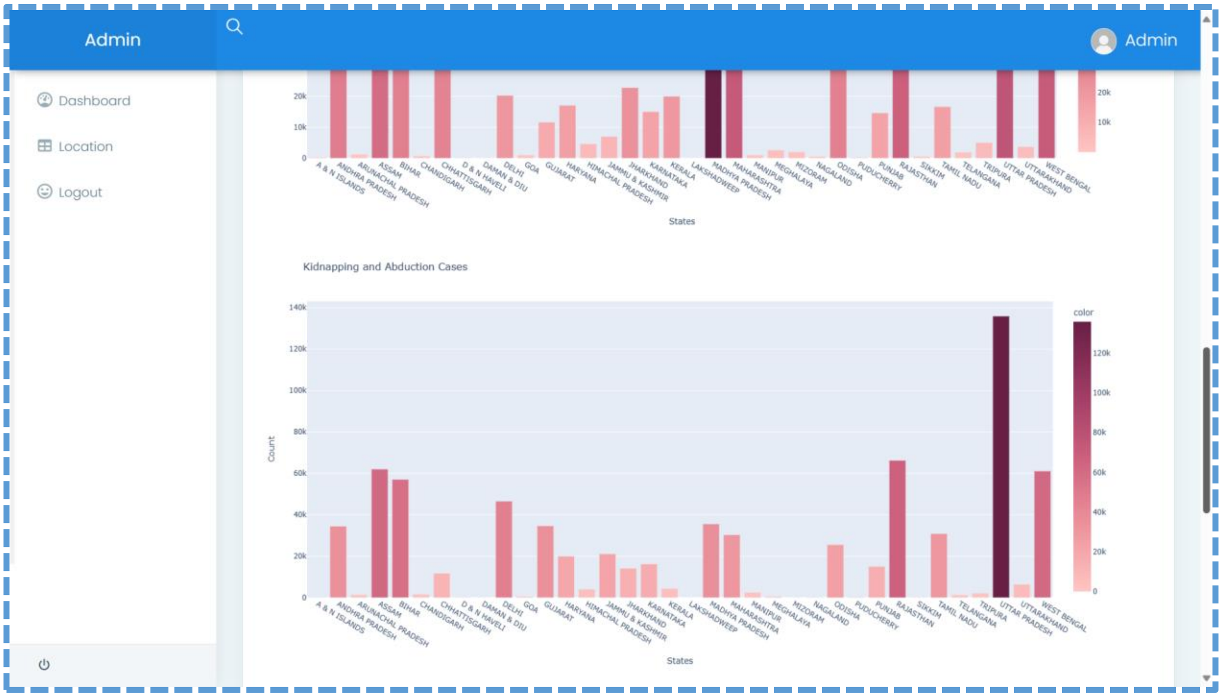



Admin
Admin
Dashboard
Location
Logout
Admin
Home
Preprocessing
Column | Non- Null Count | Dtype
Unnamed: 0
STATE/UT | 10677 | <class 'int'>
DISTRICT | 10677 | <class 'str'>
Year | 10677 | <class 'str'>
Rape | 10677 | <class 'int'>
Kidnapping and Abduction | 10677 | <class 'int'>
Dowry Deaths | 10677 | <class 'int'>
Assault on women with intent to outrage her modesty | 10677 | <class 'int'>
Insult to modesty of Women | 10677 | <class 'int'>
Cruelty by Husband or his Relatives | 10677 | <class 'int'>
Importation of Girls | 10677 | <class 'int'>
10677 | <class 'int'>
Memory usage 8007.75 KB
No. of Columns: 11, No. of Rows: 10677
Data Analysis

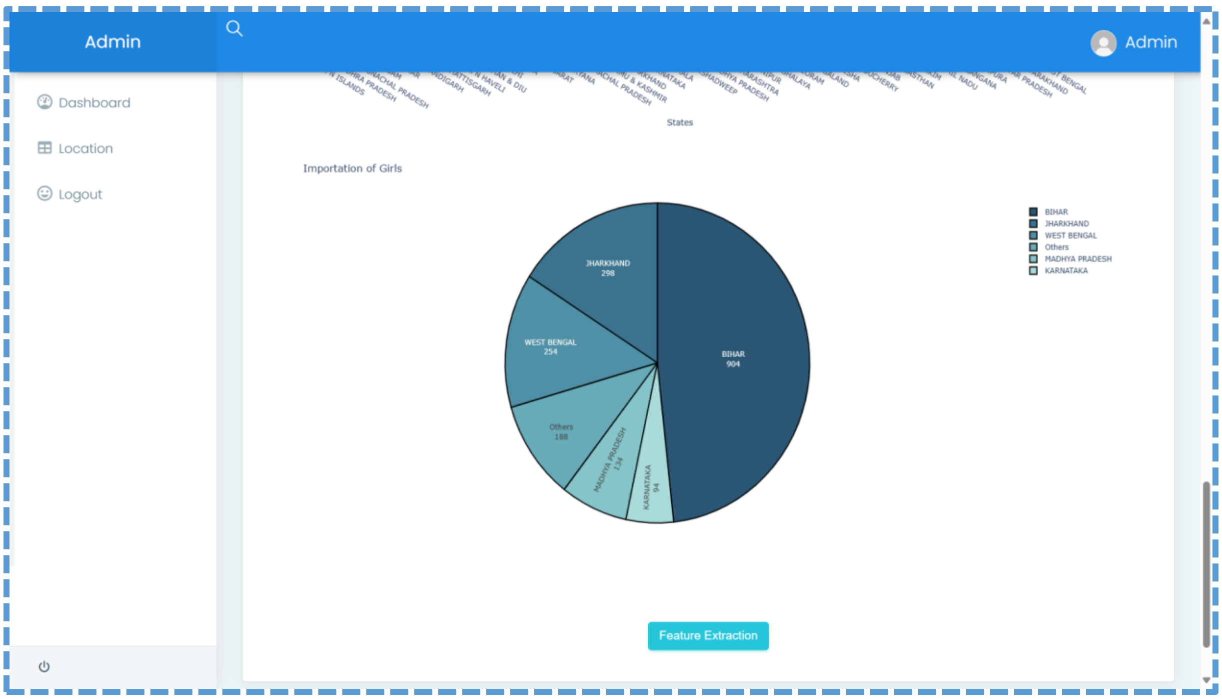

Admin
Admin
Dashboard
Location
Logout
Admin
Home
Data Analysis
Total Cases
color
619158
746198
215480
1212258
292756
2233888
1872
Count
Rape
Kidnapping and Abduction
Dowry Deaths
Assault on women with intent to outrage her modesty
Insult to modesty of Women
Cruelty by Husband or his Relatives
Importation of Girls
Crimes


Admin
Admin
Dashboard
Location
Logout
Crimes
Total Cases
Count
States
Rape Cases

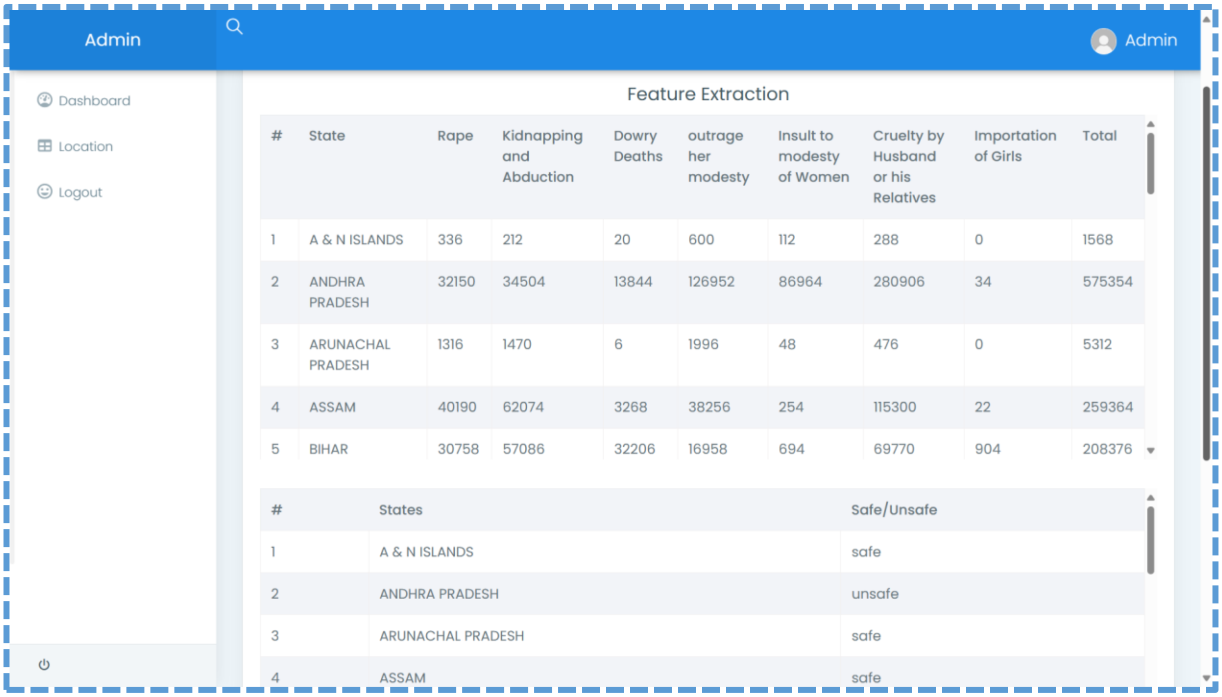

Admin
Admin
Dashboard
Location
Logout
Count
States
Kidnapping and Abduction Cases



Admin
Admin
Dashboard
Location
Logout
States
Kidnapping and Abduction Cases
Count
color
States

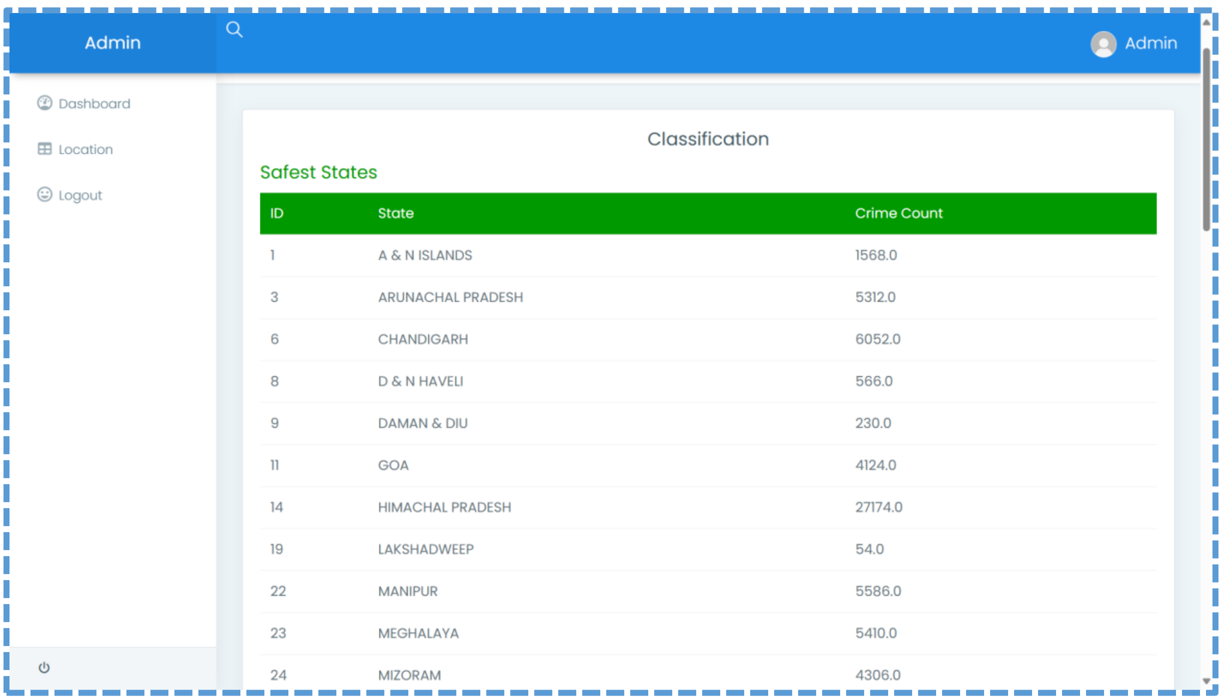


Admin
Admin
Dashboard
Location
Logout
States
Importation of Girls
BIHAR
904
JHARKHAND
298
WEST BENGAL
254
Others
188
MADHYA PRADESH
134
KARNATAKA
94
Feature Extraction



#	State	Rape	Kidnapping and Abduction	Dowry Deaths	outrage her modesty	Insult to modesty of Women	Cruelty by Husband or his Relatives	Importation of Girls	Total
1	A & N ISLANDS	336	212	20	600	112	288	0	1568
2	ANDHRA PRADESH	32150	34504	13844	126952	86964	280906	34	575354
3	ARUNACHAL PRADESH	1316	1470	6	1996	48	476	0	5312
4	ASSAM	40190	62074	3268	38256	254	115300	22	259364
5	BIHAR	30758	57086	32206	16958	694	69770	904	208376

#	States	Safe/Unsafe
1	A & N ISLANDS	safe
2	ANDHRA PRADESH	unsafe
3	ARUNACHAL PRADESH	safe
4	ASSAM	safe



ID	State	Crime Count
1	A & N ISLANDS	1568.0
3	ARUNACHAL PRADESH	5312.0
6	CHANDIGARH	6052.0
8	D & N HAVELI	566.0
9	DAMAN & DIU	230.0
11	GOA	4124.0
14	HIMACHAL PRADESH	27174.0
19	LAKSHADWEEP	54.0
22	MANIPUR	5586.0
23	MEGHALAYA	5410.0
24	MIZORAM	4306.0

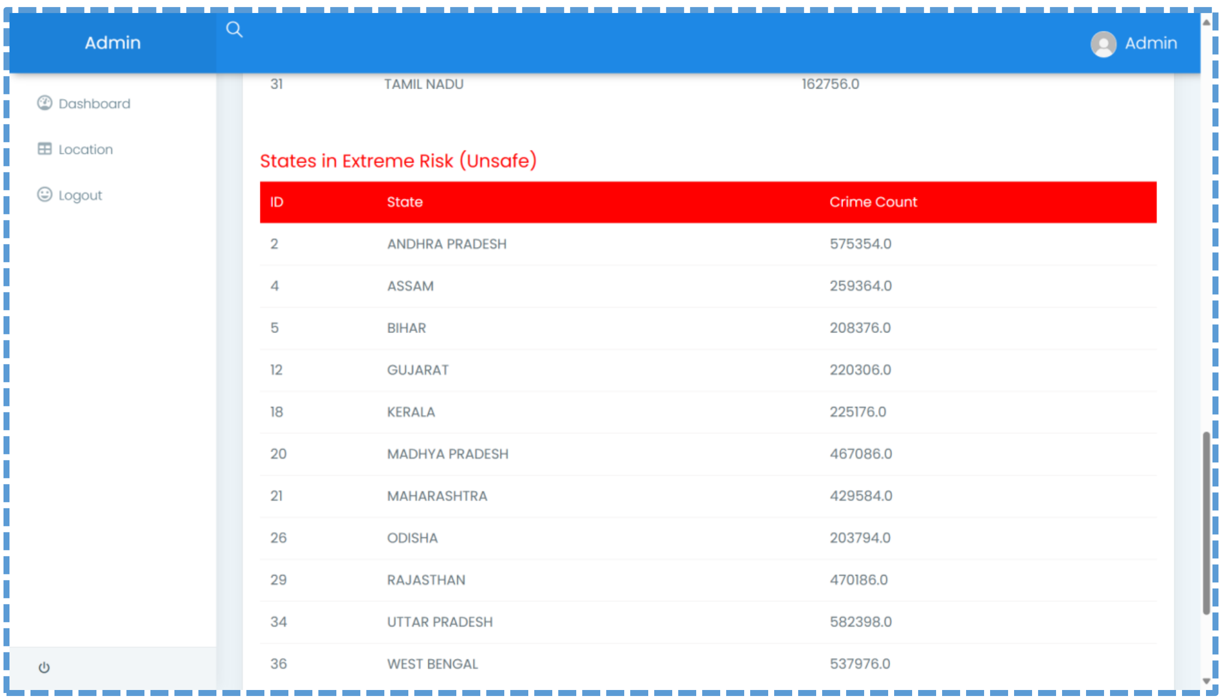
States in Risk (Unsafe)

ID	State	Crime Count
15	JAMMU & KASHMIR	71692.0
16	JHARKHAND	79580.0
28	PUNJAB	78258.0

States in Dangerous (Unsafe)

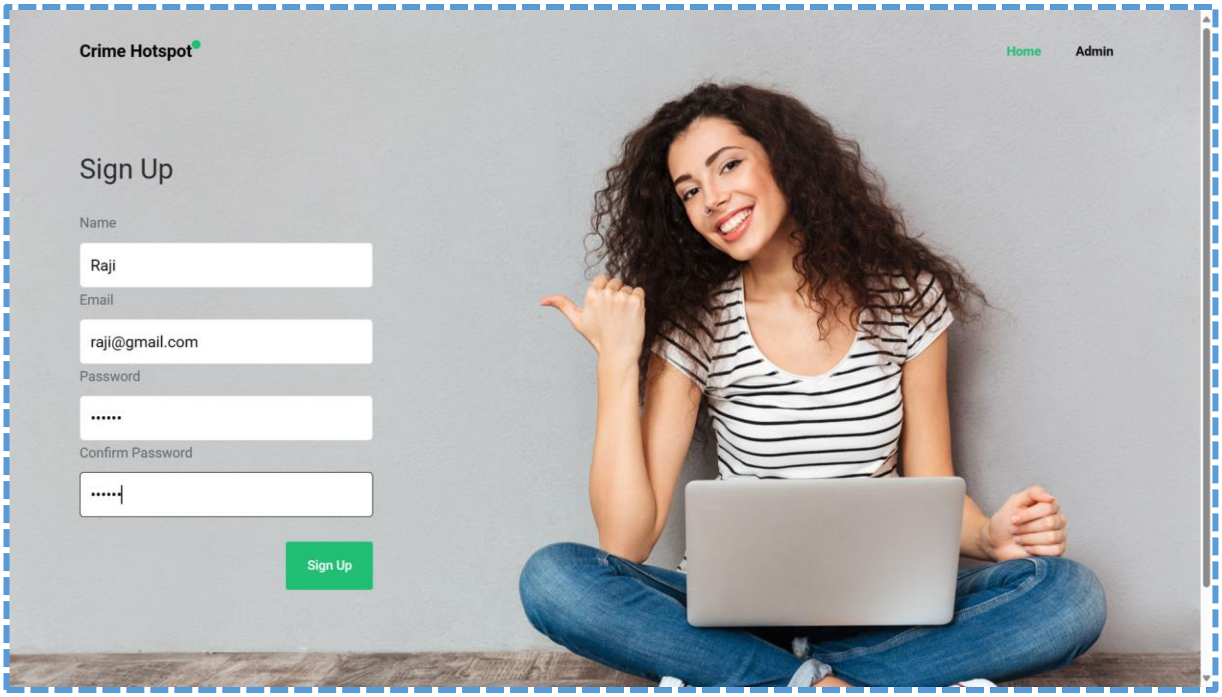
ID	State	Crime Count
7	CHHATTISGARH	118386.0
10	DELHI	152162.0
13	HARYANA	147924.0
17	KARNATAKA	179160.0
31	TAMIL NADU	162756.0

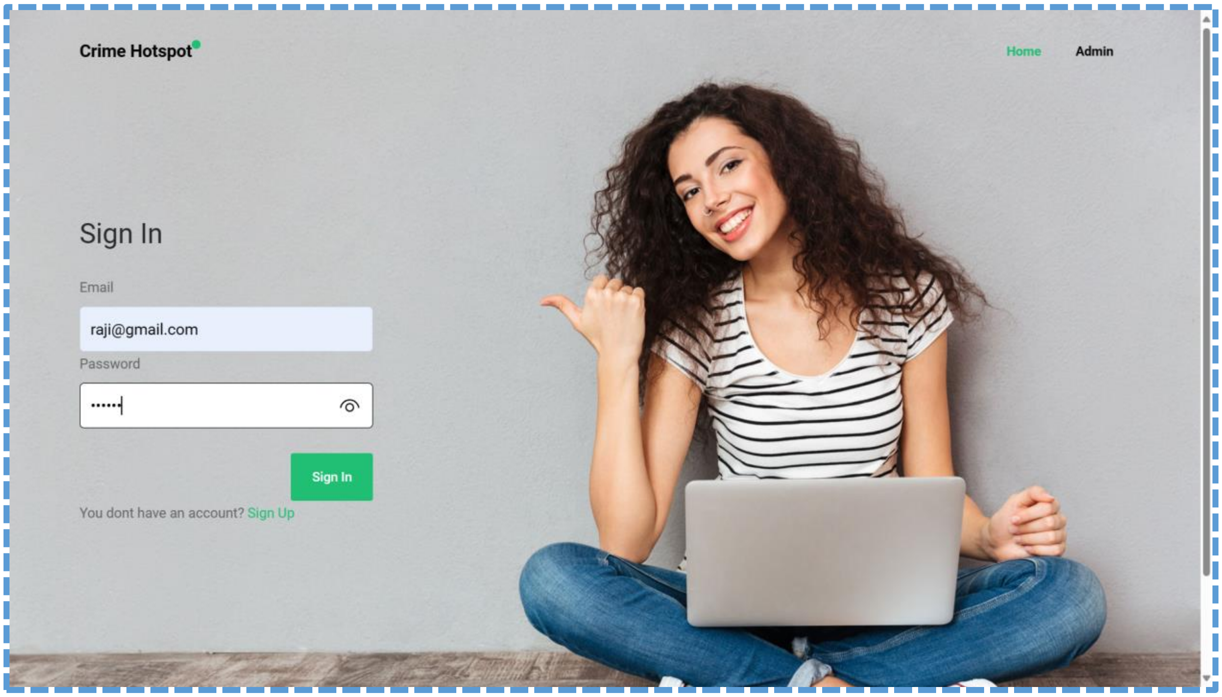
States in Extreme Risk (Unsafe)

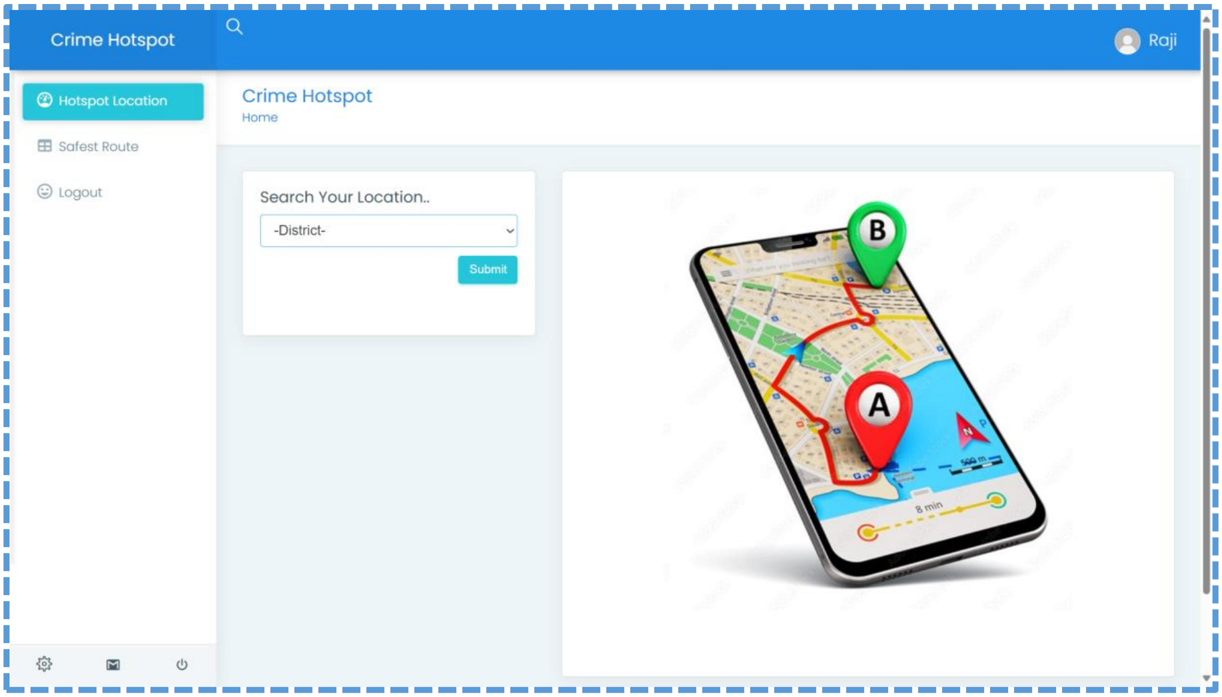


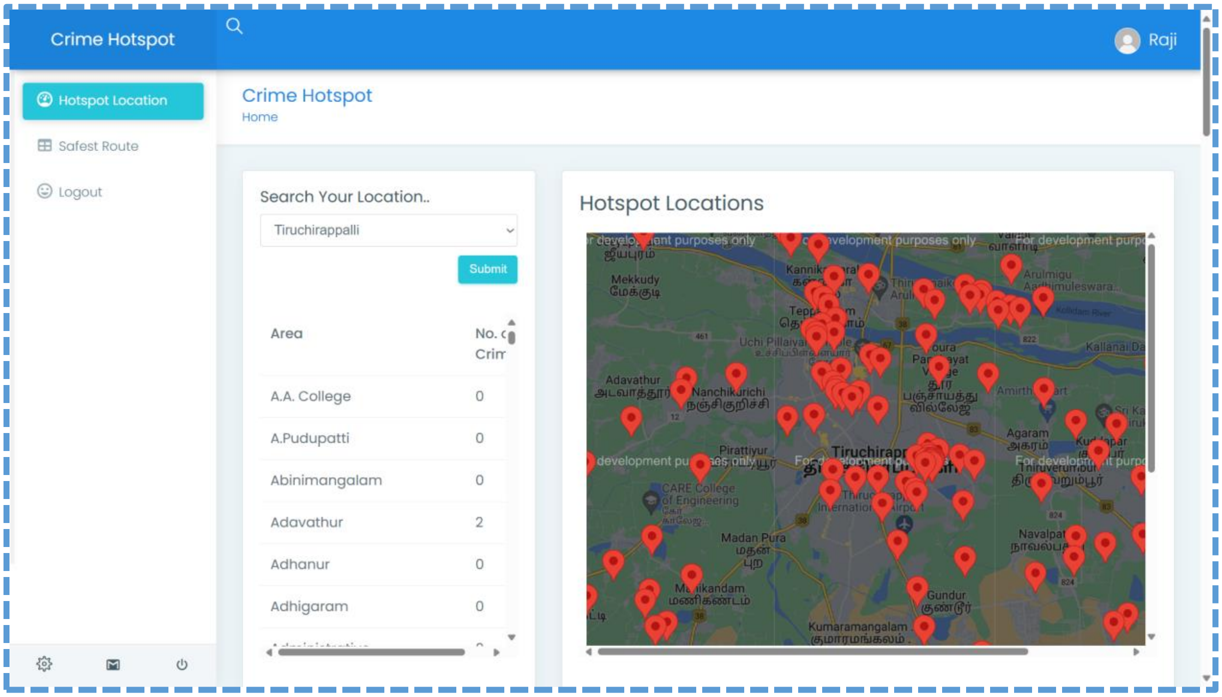
States in Extreme Risk (Unsafe)

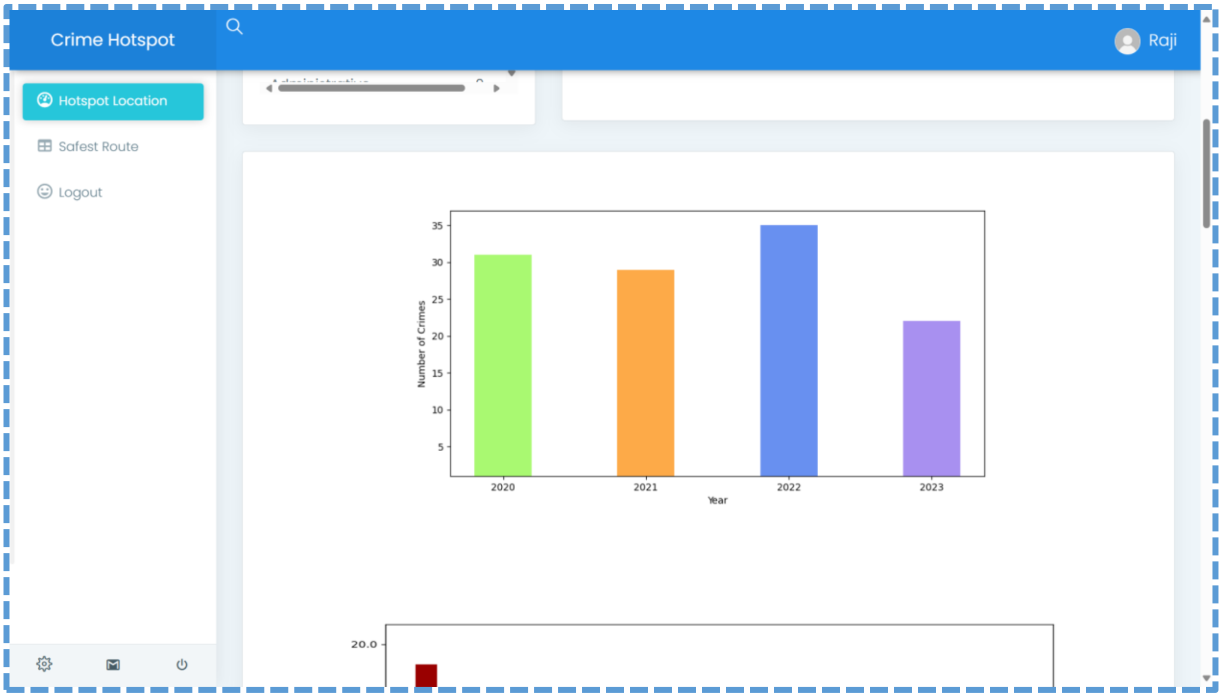
ID	State	Crime Count
2	ANDHRA PRADESH	575354.0
4	ASSAM	259364.0
5	BIHAR	208376.0
12	GUJARAT	220306.0
18	KERALA	225176.0
20	MADHYA PRADESH	467086.0
21	MAHARASHTRA	429584.0
26	ODISHA	203794.0
29	RAJASTHAN	470186.0
34	UTTAR PRADESH	582398.0
36	WEST BENGAL	537976.0

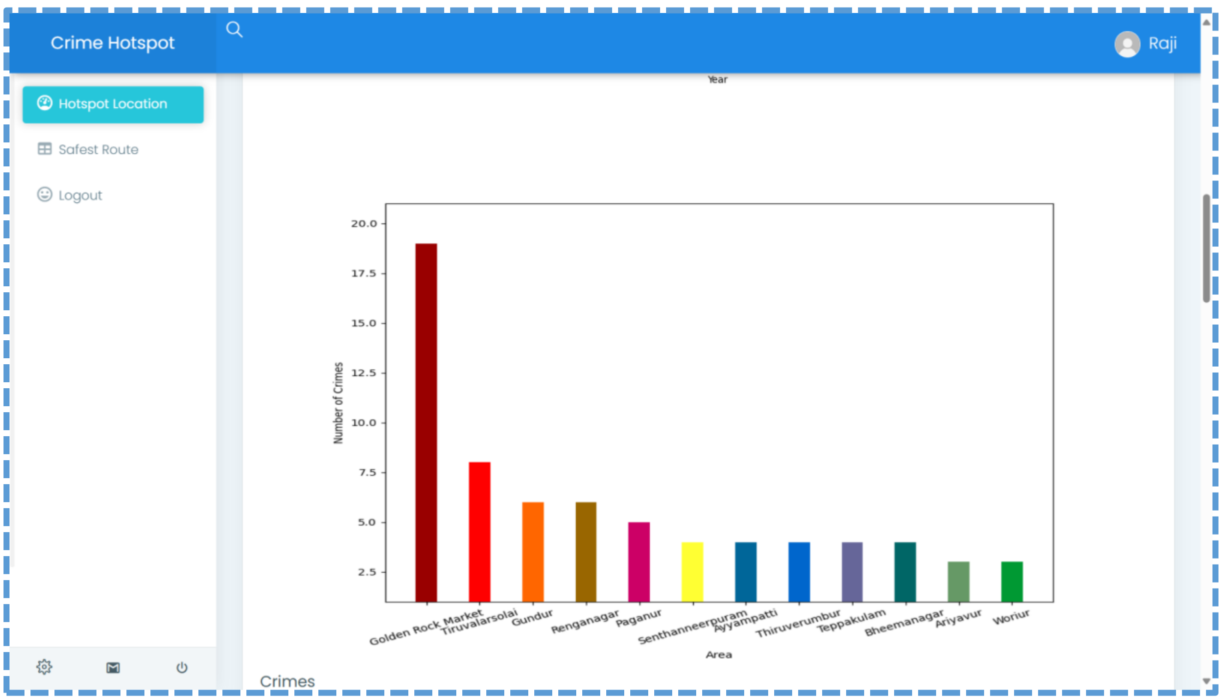

Crime Hotspot
Home
Admin
Sign Up
Name
Raji
Email
raji@gmail.com
Password
Confirm Password
Sign Up

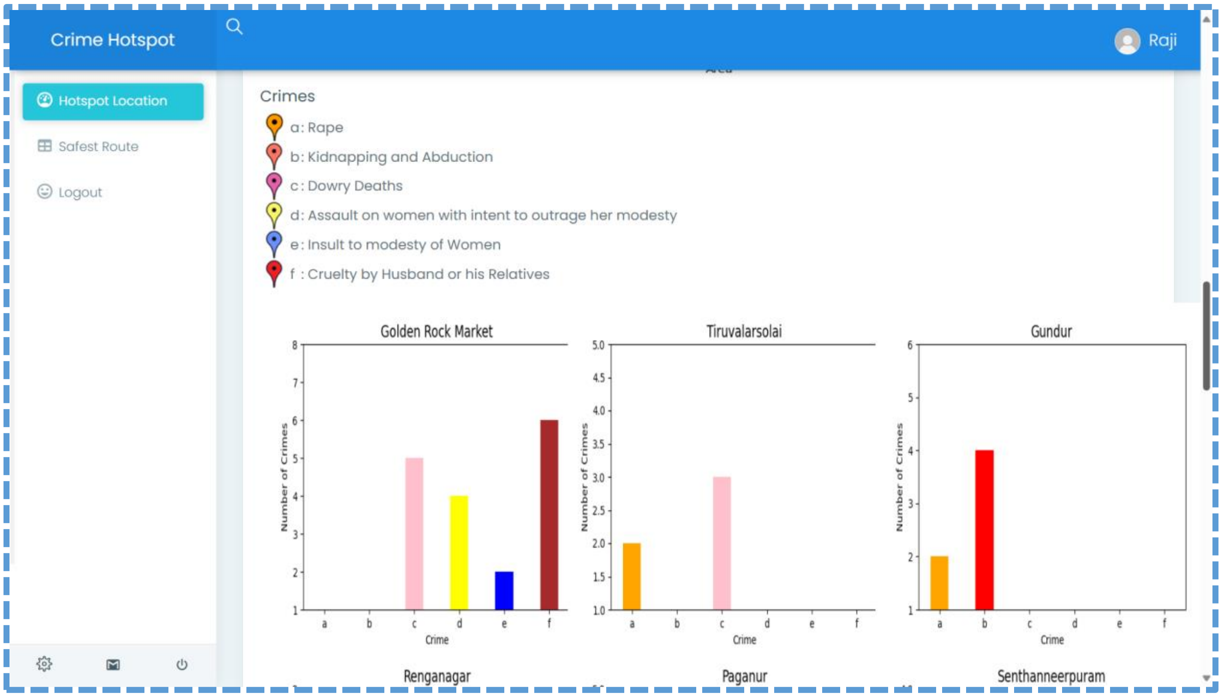

Crime Hotspot
Home
Admin
Sign In
Email
raji@gmail.com
Password
Sign In
You dont have an account? Sign Up

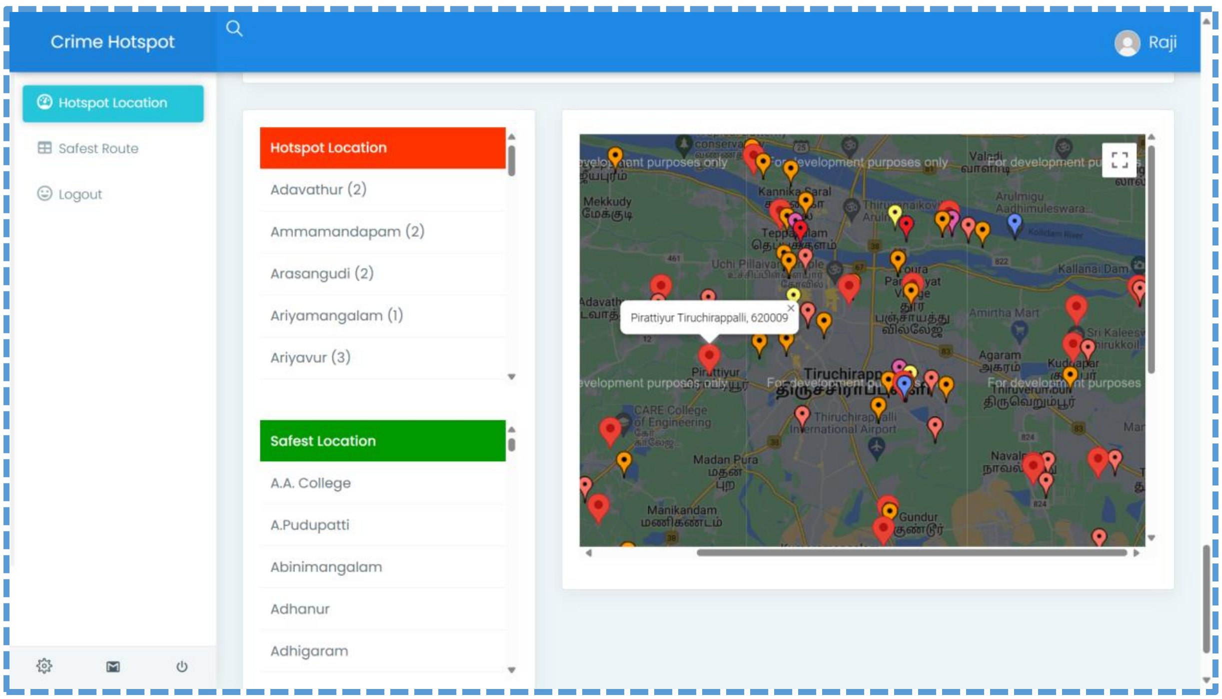

Crime Hotspot
Raji
Hotspot Location
Safest Route
Logout
Crime Hotspot
Home
Search Your Location..
-District-
Submit

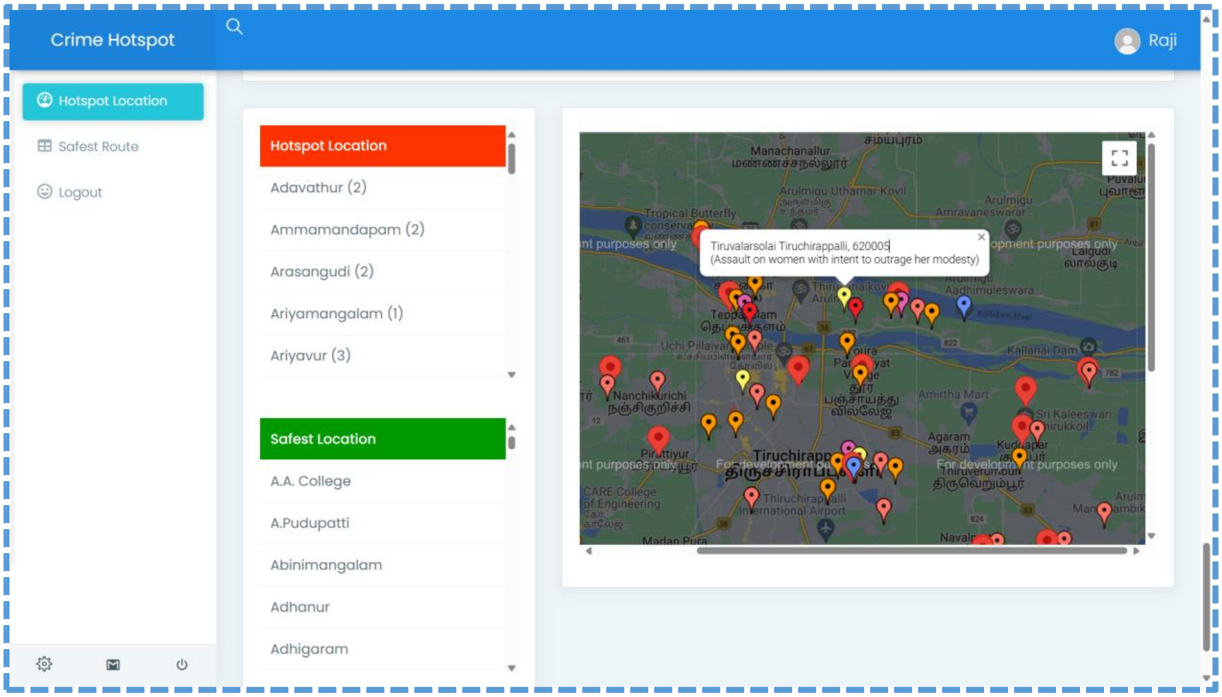

Crime Hotspot
Raji
Hotspot Location
Safest Route
Logout
Crime Hotspot
Home
Search Your Location..
Tiruchirappalli
Submit
Area
A.A. College 0
A.Pudupatti 0
Abinimangalam 0
Adavathur 2
Adhanur 0
Adhigaram 0
Hotspot Locations

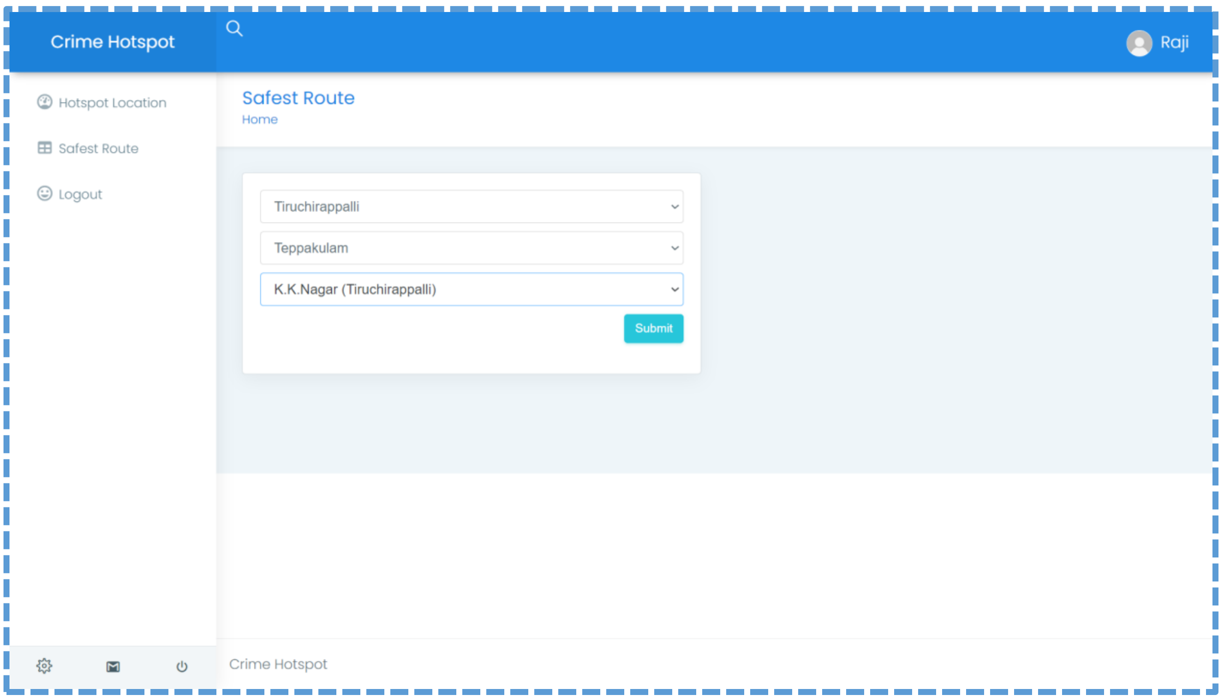

Crime Hotspot
Raji
Hotspot Location
Safest Route
Logout
35
30
25
20
15
10
5
Number of Crimes
2020
2021
2022
2023
Year
20.0

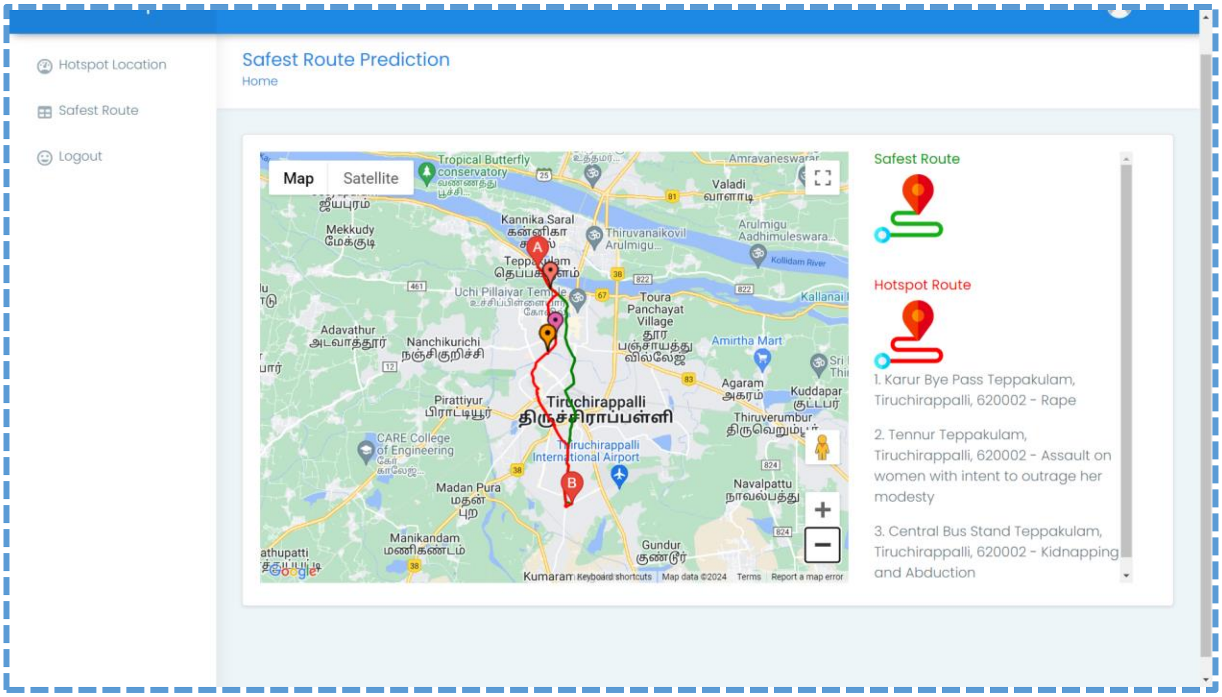

Crime Hotspot
Raji
Hotspot Location
Safest Route
Logout
Year
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
Number of Crimes
Golden Rock Market
Tiruvalarsolai
Gundur
Renganagar
Paganur
Senthanneerpuram
Ayyampatti
Thiruverumbur
Teppakulam
Bheemanagar
Ariyavur
Woriur
Area
Crimes

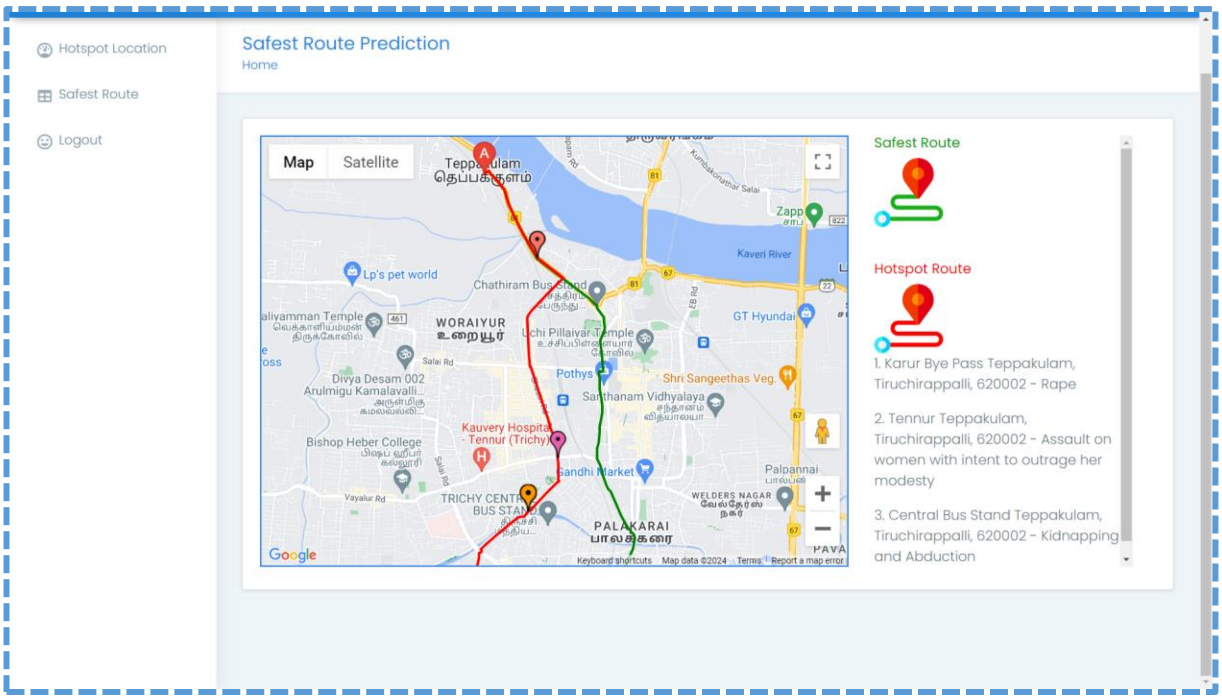

Crime Hotspot
Raji
Hotspot Location
Safest Route
Logout
Crimes
a: Rape
b: Kidnapping and Abduction
c: Dowry Deaths
d: Assault on women with intent to outrage her modesty
e: Insult to modesty of Women
f : Cruelty by Husband or his Relatives
Golden Rock Market
Tiruvalarsolai
Gundur
Number of Crimes
Crime
Renganagar
Paganur
Senthanneerpuram


Crime Hotspot
Raji
Hotspot Location
Safest Route
Logout
Hotspot Location
Adavathur (2)
Ammamandapam (2)
Arasangudi (2)
Ariyamangalam (1)
Ariyavur (3)
Safest Location
A.A. College
A.Pudupatti
Abinimangalam
Adhanur
Adhigaram
Pirattiyur Tiruchirappalli, 620009


Crime Hotspot
Raji
Hotspot Location
Safest Route
Logout
Hotspot Location
Adavathur (2)
Ammamandapam (2)
Arasangudi (2)
Ariyamangalam (1)
Ariyavur (3)
Safest Location
A.A. College
A.Pudupatti
Abinimangalam
Adhanur
Adhigaram
Tiruvalarsolai Tiruchirappalli, 620005
(Assault on women with intent to outrage her modesty)


Crime Hotspot
Raji
Hotspot Location
Safest Route
Logout
Safest Route
Home
Tiruchirappalli
Teppakulam
K.K.Nagar (Tiruchirappalli)
Submit
Crime Hotspot


Hotspot Location
Safest Route
Logout
Safest Route Prediction
Home
Map
Satellite
Tiruchirappalli
Safest Route
Hotspot Route
1. Karur Bye Pass Teppakulam, Tiruchirappalli, 620002 - Rape
2. Tennur Teppakulam, Tiruchirappalli, 620002 - Assault on women with intent to outrage her modesty
3. Central Bus Stand Teppakulam, Tiruchirappalli, 620002 - Kidnapping and Abduction


Hotspot Location
Safest Route
Logout
Safest Route Prediction
Home
Map
Satellite
Safest Route
Hotspot Route
1. Karur Bye Pass Teppakulam, Tiruchirappalli, 620002 - Rape
2. Tennur Teppakulam, Tiruchirappalli, 620002 - Assault on women with intent to outrage her modesty
3. Central Bus Stand Teppakulam, Tiruchirappalli, 620002 - Kidnapping and Abduction

9.2. SOURCE CODE

Packages

```
import math
from flask import Flask, render_template, Response, redirect, request, session, abort,
url_for
import mysql.connector
import datetime
from random import randint
from urllib.request import urlopen
import webbrowser
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

Login

```
def login():
msg=""
if request.method=='POST':
uname=request.form['uname']
pwd=request.form['pass']
cursor = mydb.cursor()
cursor.execute('SELECT * FROM register WHERE uname = %s AND pass = %s',
(uname, pwd))
account = cursor.fetchone()
```

```
if account:
session['username'] = uname
return redirect(url_for('userhome'))
else: msg = 'Incorrect username/password!'
```

User Registration

```
def register():
msg=""
if request.method=='POST':
name=request.form['name']
mobile=request.form['mobile']
email=request.form['email']
uname=request.form['uname']
pass1=request.form['pass']
mycursor = mydb.cursor()
mycursor.execute("SELECT count(*) FROM register where uname=%s",(uname,))
cnt = mycursor.fetchone()[0]
if cnt==0:
mycursor.execute("SELECT max(id)+1 FROM register")
maxid = mycursor.fetchone()[0]
if maxid is None:
maxid=1
sql = "INSERT INTO register(id,name,mobile,email,uname,pass) VALUES (%s, %s, %s, %s, %s, %s)"
val = (maxid,name,mobile,email,uname,pass1)
mycursor.execute(sql, val)
mydb.commit()
```

```
#print(mycursor.rowcount, "Registered Success")
msg="success"
else:
msg='Already Exist'
```

Training

```
#Preprocessing
def process2():
msg=""
mem=0
cnt=0
cols=0
filename = 'static/dataset/crimes.csv'
data1 = pd.read_csv(filename, header=0)
data2 = list(data1.values.flatten())
cname=[]
data=[]
dtype=[]
dtt=[]
nv=[]
i=0
sd=len(data1)
rows=len(data1.values)
col=data1.columns
for ss in data1.values:
cnt=len(ss)
i=0
```

```
while i<cnt:
j=0
x=0
for rr in data1.values:
dt=type(rr[i])
if rr[i]!="":
x+=1
j+=1
dtt.append(dt)
nv.append(str(x))
i+=1
arr1=np.array(col)
arr2=np.array(nv)
data3=np.vstack((arr1, arr2))
arr3=np.array(data3)
arr4=np.array(dtt)
data=np.vstack((arr3, arr4))
print(data)
cols=cnt
mem=float(rows)*0.75
#Data Analysis
df=pd.read_csv('static/dataset/crimes.csv')
df.head()
df.drop('Unnamed: 0',axis=1,inplace=True)
df['STATE/UT'].unique()
df.loc[df['STATE/UT'] == 'A&N Islands', 'STATE/UT'] = 'A & N ISLANDS'
df.loc[df['STATE/UT'] == 'D&N Haveli', 'STATE/UT'] = 'D & N HAVELI'
```

```
df.loc[df['STATE/UT'] == 'Delhi UT', 'STATE/UT'] = 'DELHI'
df['STATE/UT'] = pd.Series(str.upper(i) for i in df['STATE/UT'])
df['DISTRICT'] = pd.Series(str.upper(i) for i in df['DISTRICT'])
#stroring the sum of all crimes comitted within a state statewise
state_all_crimes = df.groupby('STATE/UT').sum()
#droping the sum of year column
state_all_crimes.drop('Year',axis=1,inplace=True)
#adding a column containig the total crime against women in that state
col_list= list(state_all_crimes)
state_all_crimes['Total']=state_all_crimes[col_list].sum(axis=1)
all_crimes = state_all_crimes
#sorting the statewise crime from highest to lowest
state_all_crimes.sort_values('Total',ascending=False)
state_all_crimes=state_all_crimes.reset_index()
total_df=state_all_crimes.sum(axis=0).reset_index()
tf=pd.DataFrame(total_df)
tf=tf.drop([0])
tf=tf.drop([8])
sorted_df = state_all_crimes.sort_values('Total',ascending=False)
#Feature Extraction
state_all_crimes=state_all_crimes.reset_index()
total_df=state_all_crimes.sum(axis=0).reset_index()
tf=pd.DataFrame(total_df)
tf=tf.drop([0])
tf=tf.drop([8])
sorted_df = state_all_crimes.sort_values('Total',ascending=False)
dat=all_crimes = all_crimes.reset_index()
```

```
for ss in dat.values:
data.append(ss)
all_crimes.shape
#finding the mean number of crimes
m=all_crimes['Total'].mean()
print('mean=',m)
#finding the quantiles
q = np.quantile(all_crimes['Total'],[0.25,0.75])
print(q)
l=q[0]
u=q[1]
#copying the state_all_crimes to a new dataframe to normalise values and predict
df_kmeans = all_crimes.loc[:,all_crimes.columns!="STATE/UT"]
#adding an additional column called output
output=[]
for i in df_kmeans['Total']:
if i >= m:
output.append(1)#redzone
elif m > i:
output.append(0)#safe
all_crimes['output']=output
df_kmeans_y=all_crimes['output']
#feature scaling
from sklearn.preprocessing import MinMaxScaler
cols = df_kmeans.columns
ms=MinMaxScaler()
df_kmeans = ms.fit_transform(df_kmeans)
```

```
df_kmeans = pd.DataFrame(df_kmeans,columns=[cols])
df_kmeans.head()
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=2, random_state=0)
kmeans.fit(df_kmeans)
kmeans.inertia_
#checking the accuracy
labels = kmeans.labels_
# check how many of the samples were correctly labeled
correct_labels = sum(df_kmeans_y == labels)
print('labels:',labels)
print('df_kmeans output:',df_kmeans_y)
print("Result: %d out of %d samples were correctly labeled." % (correct_labels,
df_kmeans_y.size))
#based on the prediction of the k means algorithm classifying the states
#as safe or unsafe for women
final=[]
for i in range(len(labels)):
state=all_crimes['STATE/UT'][i]
label = labels[i]
if label == 1:
final.append([state,'unsafe'])
else:
final.append([state,'safe'])
dat2=final_df = pd.DataFrame(final, columns=['STATES/UT', 'SAFE/UNSAFE'])
#final_df
for ss2 in dat2.values:
```

```
data2.append(ss2)
#SVM Classification
class SVM:
def fit(self, X, y):
n_samples, n_features = X.shape# P = X^T X
K = np.zeros((n_samples, n_samples))
for i in range(n_samples):
for j in range(n_samples):
K[i,j] = np.dot(X[i], X[j])
P = cvxopt.matrix(np.outer(y, y) * K)# q = -1 (1xN)
q = cvxopt.matrix(np.ones(n_samples) * -1)# A = y^T
A = cvxopt.matrix(y, (1, n_samples))# b = 0
b = cvxopt.matrix(0.0)# -1 (NxN)
G = cvxopt.matrix(np.diag(np.ones(n_samples) * -1))# 0 (1xN)
h = cvxopt.matrix(np.zeros(n_samples))
solution = cvxopt.solvers.qp(P, q, G, h, A, b)# Lagrange multipliers
a = np.ravel(solution['x'])# Lagrange have non zero lagrange multipliers
sv = a > 1e-5
ind = np.arange(len(a))[sv]
self.a = a[sv]
self.sv = X[sv]
self.sv_y = y[sv]# Intercept
self.b = 0
for n in range(len(self.a)):
self.b += self.sv_y[n]
self.b -= np.sum(self.a * self.sv_y * K[ind[n], sv])
self.b /= len(self.a)# Weights
```

```
self.w = np.zeros(n_features)
for n in range(len(self.a)):
self.w += self.a[n] * self.sv_y[n] * self.sv[n]
def project(self, X):
return np.dot(X, self.w) + self.b
def predict(self, X):
return np.sign(self.project(X))
df['STATE/UT'].unique()
df.loc[df['STATE/UT'] == 'A&N Islands', 'STATE/UT'] = 'A & N ISLANDS'
df.loc[df['STATE/UT'] == 'D&N Haveli', 'STATE/UT'] = 'D & N HAVELI'
df.loc[df['STATE/UT'] == 'Delhi UT', 'STATE/UT'] = 'DELHI'
#converting all the state names to capitals
df['STATE/UT'] = pd.Series(str.upper(i) for i in df['STATE/UT'])
df['DISTRICT'] = pd.Series(str.upper(i) for i in df['DISTRICT'])
#stroring the sum of all crimes comitted within a state statewise
state_all_crimes = df.groupby('STATE/UT').sum()
#droping the sum of year column
state_all_crimes.drop('Year',axis=1,inplace=True)
#adding a column containig the total crime against women in that state
col_list= list(state_all_crimes)
state_all_crimes['Total']=state_all_crimes[col_list].sum(axis=1)
all_crimes = state_all_crimes
#sorting the statewise crime from highest to lowest
state_all_crimes.sort_values('Total',ascending=False)
state_all_crimes=state_all_crimes.reset_index()
total_df=state_all_crimes.sum(axis=0).reset_index()
tf=pd.DataFrame(total_df)
```

```
tf=tf.drop([0])
sorted_df = state_all_crimes.sort_values('Total',ascending=False)
dat=all_crimes = all_crimes.reset_index()
for ss in dat.values:
dt=[]
dt.append(ss[0])
f1=float(ss[8])
if f1>200000:
dt.append("Not Safety")
elif f1>100000:
dt.append("Medium Safety")
elif f1>50000:
dt.append("Low Safety")
else:
dt.append("Safety")
data.append(dt)
```

Search Location

```
def userhome():
msg=""
uname=""
st=""
if 'username' in session:
uname = session['username']
mycursor = mydb.cursor()
mycursor.execute("SELECT * FROM register where uname=%s",(uname,))
usr = mycursor.fetchone()
if request.method=='POST':
```

```
splace=request.form['splace']
dplace=request.form['dplace']
loc1='%'+splace+'%'
loc2='%'+dplace+'%'
mycursor.execute("SELECT count(*) FROM routes where location like %s",(loc1,))
cn = mycursor.fetchone()[0]
if cn>0:
mycursor.execute("SELECT * FROM routes where location like %s",(loc1,))
dd1 = mycursor.fetchall()
v2=""
st="1"
for dd in dd1:
rid=dd[2]
sid=dd[3]
mycursor.execute("SELECT count(*) FROM routes where location like %s && route_id=%s",(loc2,rid))
cn2 = mycursor.fetchone()[0]
if cn2>0:
mycursor.execute("SELECT * FROM routes where location like %s && route_id=%s",(loc2,rid))
dd2 = mycursor.fetchall()
for dd3 in dd2:
rid2=dd3[2]
sid2=dd3[3]
if sid<sid2:
v=str(sid)+","+str(sid2)
```

```
else:
v=str(sid2)+","+str(sid)
v2=str(rid2)+","+v+"|"
ff=open("static/route.txt","w")
ff.write(v2)
ff.close()
msg="Location"
else: st="2
```

CHAPTER 10

BOTTOMLINE

10.1. CONCLUSION

In conclusion, the proposed system is a useful tool to predict crime hotspots and provide recommendations to women for safer routes. The system employs the explainable Decision Tree (xDT) algorithm for crime hotspot prediction and integrates Google Maps API for visualization and location-based recommendations. The system is designed with various modules, including Crime Hotspot Finder Web App, Crime Hotspot Classifier using xDT, Crime Hotspot Pre-processing, and Crime Hotspot Prediction using xDT. Each module serves a specific purpose and contributes to the overall functionality of the system. The proposed system has many advantages over the existing manual systems and other data mining techniques. It provides a more accurate and efficient way of predicting crime hotspots, which will allow law enforcement agencies to take proactive measures to prevent crime against women. The system is also user-friendly, making it easy for end-users to access and utilize. During the feasibility study and testing phase, the system showed promising results in terms of accuracy, precision, and recall. The user-friendly interface and the integration of Google Maps API provided a smooth and interactive experience for users. Furthermore, the system has the potential for further development and integration with other crime prevention measures such as CCTV cameras and police patrols. It can also be expanded to include other types of crimes and demographic groups. Overall, the Crime Hot Spot Prediction and Localization System for Women Safety using xDT and Google Map API is a step towards creating a safer environment for women and empowering them to make informed decisions regarding their safety.

10.2. FUTURE ENHANCEMENT

There are several areas where this can be improved in the future. Some of these areas are:

- **Integration with social media platforms:** The system can be integrated with social media platforms, such as Twitter and Facebook, to gather additional data on crime incidents and public perceptions of safety in certain areas.
- **Integration with emergency services:** The system can be integrated with emergency services, such as police and ambulance services, to enable faster response times to crime incidents in hotspots.
- **Expansion to other regions:** The system can be expanded to cover other regions beyond the current scope. This could involve integrating additional data sources and customizing the system to suit the specific needs of each region.
- **Integration with real-time crime data:** The system can be enhanced by integrating with real-time crime data from police departments or other crime reporting sources. This will help in improving the accuracy of the system and make it more effective in predicting crime hotspots.
- **Integration with mobile devices:** The system can be integrated with mobile devices to provide real-time alerts to women about crime hotspots in their vicinity. This will make the system more user-friendly and convenient to use.
- **Collaboration with law enforcement agencies:** Collaboration with law enforcement agencies can help in improving the accuracy of the system and make it more effective in predicting crime hotspots. Law enforcement agencies can provide valuable insights and feedback on the accuracy of the system.

REFERENCE

1. Mohler, G. O., Short, M. B., Brantingham, P. J., Schoenberg, F. P., & Tita, G. E. (2011). Self-exciting point process modeling of crime. Journal of the American statistical Association, 106(493), 100-108.
2. Malik, O. M., Hussain, I., & Rehman, A. (2019). Predicting crime hotspots using deep learning. In 2019 International Conference on Innovative Computing (ICIC) (pp. 1-6). IEEE.
3. Gerber, M. S., Johnson, M. P., & Berman, M. (2011). Predictive policing and reasonable suspicion. Emory Law Journal, 60, 585.
4. Ahlstrom, C. & Allen, T. (2014). Applying predictive analytics in the public sector: Techniques and examples. Boca Raton: CRC Press.
5. Bhattacharyya, S., Kalita, J. K., & Sarma, K. K. (2014). Crime modeling and mapping using geospatial technologies. Boca Raton: CRC Press.
6. Jadhav, J., & Darade, T. (2021). Women's Safety using Machine Learning: A Review. International Journal of Advanced Research in Computer Science, 12(1), 1-5.
7. Kozievitch, N. P., de Albuquerque, J. P., & de Oliveira, A. L. (2020). A Review of Crime Prediction Based on Environmental Factors. Sensors, 20(10), 2991.