# Project Based Learning

*on*

**“Women’s Safety App”**

By

**Dishant Das (202300601)**
**Roland Singh(202300279)**
**Richa Telang(202300544)**
**Tryphena Anet Chettri(202300523)**
**Priyajit Chakraborty(202300398)**

*In partial fulfilment of requirements for the award of degree in*
Bachelor of Technology in Computer Science and Engineering
(2025)

*Under the Project Guidance of*

**Ms. Sayantani Roy**
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**
**SIKKIM MANIPAL INSTITUTE OF TECHNOLOGY**
**(A constituent college of Sikkim Manipal University)**
MAJITAR, RANGPO, EAST SIKKIM – 737136

# Content



| SL No | Contents | Slides no. |
|---|---|---|
| 1 | Abstract | 3 |
| 2 | Introduction | 4 |
| 3 | Literature Survey | 5 |
| 4 | Problem identification and formalization | 10 |
| 5 | Planning on critical thinking | 11 |
| 6 | Design Solution | 13 |
| 7 | Necessary Tools for implementation | 18 |
| 8 | Development of various identified modules | 20 |
| 9 | Validation of developed modules | 28 |
| 10 | Literature Survey: Key Findings Content | 32 |
| 11 | Seminars and Workshops | 33 |
| 12 | Gant chart | 34 |
| 13 | References | 35 |

# ABSTRACT


- This project presents a dual-component safety system: a reactive Android application and a proactive ML prediction engine.
- The Android app provides immediate emergency help, featuring a one-touch SOS button, shake-to-alert, and a map to find nearby police stations & hospitals.
- The ML engine uses an NLP pipeline to extract crime data from news and a XGBoost model to predict safety scores for different areas. This model powers a "Safest Route" simulation, demonstrating a system that can intelligently navigate users around high-risk zones.

# INTRODUCTION

- Women's safety remains a major concern, with many facing threats in both public and private spaces.
- Often, incidents go unreported due to fear, stigma, or lack of immediate support.
- With the rise of smartphones and connectivity, technology can play a powerful role in enhancing personal safety.
- This women’s safety app is designed to empower women by providing real-time safety tools like SOS alerts and live location sharing, making help more accessible.

# INFORMATION GATHERING THROUGH LITERATURE SURVEY

| S. no | Author name | Journal name, vol., page, year | Title of the Paper | Inference | Research Gap | Relevance to the work |
|---|---|---|---|---|---|---|
| 1. | Dhruv Chand, Sunil Nayak, Karthik S. Bhat, Shivani Parikh, Yuvraj Singh, Amita Ajith Kamath. | TENCON 2015 - 2015 IEEE Region 10 Conference, Date: 01-04 November 2015 | **A mobile application for Women's Safety: WoSApp** | ● WoSApp is a mobile application that helps women quickly alert police during emergencies through phone shaking or a PANIC button.<br>● The app automatically sends the user's real-time location and emergency contact information to authorities. | ● The app relies heavily on the availability of mobile internet and GPS, which may not function reliably in remote or low-signal areas.<br>● The paper does not address scenarios where the phone is inaccessible to the victim.<br>● There is no mention of integration with law enforcement systems to confirm whether alerts are received. | ● Highlights the importance of emergency features, which can be integrated into the app for quick response.<br>● Emphasizes the need for location tracking to ensure accurate, timely assistance from authorities.<br>● Raises the issue of network and battery dependency. |

# INFORMATION GATHERING THROUGH LITERATURE SURVEY

| S. no | Author name | Journal name, vol., page, year | Title of the Paper | Inference | Research Gap | Relevance to the work |
|---|---|---|---|---|---|---|
| 2. | K. Srinivas, Suwarna Gothane, C. Saisha Krithika; Anshika; T.Susmitha | International Journal for Research in Applied Science & Engineering Technology (IJRASET), Vol. 9 Issue VI, 2021 pp. 232–236 | **Android App for Women Safety** | ● The paper presents an Android-based safety application that sends continuous SOS alerts with GPS location until manually stopped.<br><br>● The system is focused on user-friendliness and quick emergency response.<br><br>● Demonstrates a simple yet reliable approach for personal safety communication. | ● Lacks AI/ML intelligence and predictive features.<br><br>● Limited only to SMS-based tracking, with no real-time mapping.<br><br>● No backend database integration for storing or analyzing unsafe locations. | ● Formed the base for SOS module.<br><br>● Inspired the inclusion of shake detection and live GPS sharing.<br><br>● Helped define app's simplicity and reliability principle. |

# INFORMATION GATHERING THROUGH LITERATURE SURVEY

| S. no | Author name | Journal name, vol., page, year | Title of the Paper | Inference | Research Gap | Relevance to the work |
|---|---|---|---|---|---|---|
| 3. | Suriya Prakash S., Nithya S., Rithika S., Sangavi S. | International Journal for Research Trends and Innovation (IJRTI), Vol. 10, Issue 5, May 2025, pp. 180–184 | **Smartphone-Based Women Safety Application with Location Tracking and Emergency Response** | ● Proposed a GPS-based unsafe zone alert system that triggers warnings within a 3 km radius.<br>● Integrated Google Maps API for navigation and location display.<br>● Sent automatic alerts to nearby police and emergency centers. | ● Unsafe zones were static and not updated dynamically.<br>● Lacked machine learning or predictive scoring for safety analysis.<br>● Did not include user feedback or adaptive route planning. | ● Influenced the creation of our real-time unsafe zone alert system.<br>● Motivated us to include ML-based route scoring for adaptability.<br>● Strengthened the app's location awareness and proactive alerting design. |

# INFORMATION GATHERING THROUGH LITERATURE SURVEY

| S. no | Author name | Journal name, vol., page, year | Title of the Paper | Inference | Research Gap | Relevance to the work |
|---|---|---|---|---|---|---|
| 4. | Geetanjali Institute of Technology , Avishi Sharma; Lakshika Sarupria; Siddhika Dhabhai; Viral Jain; Yashveer Singh Deora; Charu Kavadia. | International Advanced Research Journal in Science, Engineering and Technology (IARJSET – ICMART 2023 Conference) pp. 45–48 | **Mobile Application on Women Safety** | ● Developed a power-button-triggered SOS function for quick emergency response.<br>● Designed a minimalist, user-friendly interface suitable for panic situations.<br>● Incorporated Google Maps integration to locate nearby police stations. | ● Focused only on reactive measures, no preventive modules<br>.<br>● Did not include data-driven prediction or ML models.<br>● Missing visualization or scoring of unsafe areas. | ● Guided our one-tap emergency interface and UI/UX layout.<br>● Inspired integration of map-based navigation with emergency routes.<br>● Reinforced the need for fast and accessible SOS features. |

# INFORMATION GATHERING THROUGH LITERATURE SURVEY

| S. no | Author name | Journal name, vol., page, year | Title of the Paper | Inference | Research Gap | Relevance to the work |
|---|---|---|---|---|---|---|
| 5. | Sharon Levy, Wenhan Xiong, Elizabeth Belding, William Yang Wang | ACM / arXiv preprint arXiv:1811.01147, 2018 pp. 1–10 | **SafeRoute: Learning to Navigate Streets Safely in an Urban Environment** | ● Proposed a Reinforcement Learning-based safe route planning model using real crime data.<br><br>● The algorithm achieved a 17% reduction in exposure to unsafe areas.<br><br>● Demonstrated that ML models can effectively enhance safety-aware navigation. | ● Focused on general pedestrian safety, not gender-specific needs.<br><br>● Required localized datasets for regional accuracy.<br><br>● Implementation complexity limits its use on mobile apps. | ● Served as the foundation for our ML-based Safe Route Recommendation model.<br><br>● Provided insights into safety vs distance cost optimization.<br><br>● Helped in structuring our data pipeline and route scoring mechanism. |

# PROBLEM IDENTIFICATION AND FORMALIZATION

- **Problem 1 (Reactive):**
  In emergencies, it’s slow and difficult to call for help or alert contacts, especially if hands aren’t free.

- **Problem 2 (Proactive):**
  People lack real-time safety insights for specific areas, often entering unsafe zones unknowingly.

- **Formalization:**
  To design a complete safety system with:

  A **Reactive Component** (SOS App for emergencies)

  A **Proactive Component** (ML model predicting unsafe areas)

# PLANNING ON CRITICAL THINKING AND MODULE DEFINITION

**Module 1 – SheGuard Android App (Reactive)**

- SOS button, Shake Detection, SMS Alerts
- Helplines and Live Safety Bulletin
- Map-based navigation showing nearby police/hospitals

# PLANNING ON CRITICAL THINKING AND MODULE DEFINITION

## Module 2 – Safety Prediction Engine (Proactive)

- NLP (NER + LSTM) extracting incidents from news
- User report collection
- XGBoost prediction model
- Safest Route algorithm

# DESIGN SOLUTION FOR VARIOUS IDENTIFIED MODULES/UNITS

**Module 1: SheGuard Android App (Client)**
The mobile application serves as the user-facing component of the project. It is designed for accessibility, reliability, and real-time emergency support.

- **Unit 1.1 – Core SOS System:** Includes the primary SOS button, shake detection feature, and automated SMS alert system that sends the user’s live location to emergency contacts.

- **Unit 1.2 – Emergency Navigation:** Displays nearby police stations and hospitals using Google Maps API, allowing users to navigate directly to safe locations during emergencies.

- **Unit 1.3 – Information Hub:** Provides one-tap access to important helpline numbers and displays a "Safety Bulletin" with real-time safety information and alerts.

# DESIGN SOLUTION FOR VARIOUS IDENTIFIED MODULES/UNITS

**App Flow:**

- **User Setup & Registration:** The user registers and provides emergency contact details.
- **Home Screen:** Displays the main dashboard with SOS, Helplines, and Maps features.
- **Maps API Integration:** Fetches current GPS coordinates and plots nearby safe zones.
- **Emergency Assistance:** On trigger (tap or shake), the app sends alerts and displays the nearest police stations or hospitals.

# DESIGN SOLUTION FOR VARIOUS IDENTIFIED MODULES/UNITS

**Module 2: Safety Prediction Engine (Backend)**
The backend module applies machine learning and data analysis to proactively identify unsafe regions and recommend safer routes.

- **Unit 2.1 – NLP Data Pipeline (News Extraction):** Collects and processes news articles using Natural Language Processing to extract location-based crime data.

- **Unit 2.2 – Prediction Model (XGBoost):** Uses structured datasets to train an ML model that predicts a safety score (1 = Unsafe, 3 = Safe) for specific areas.

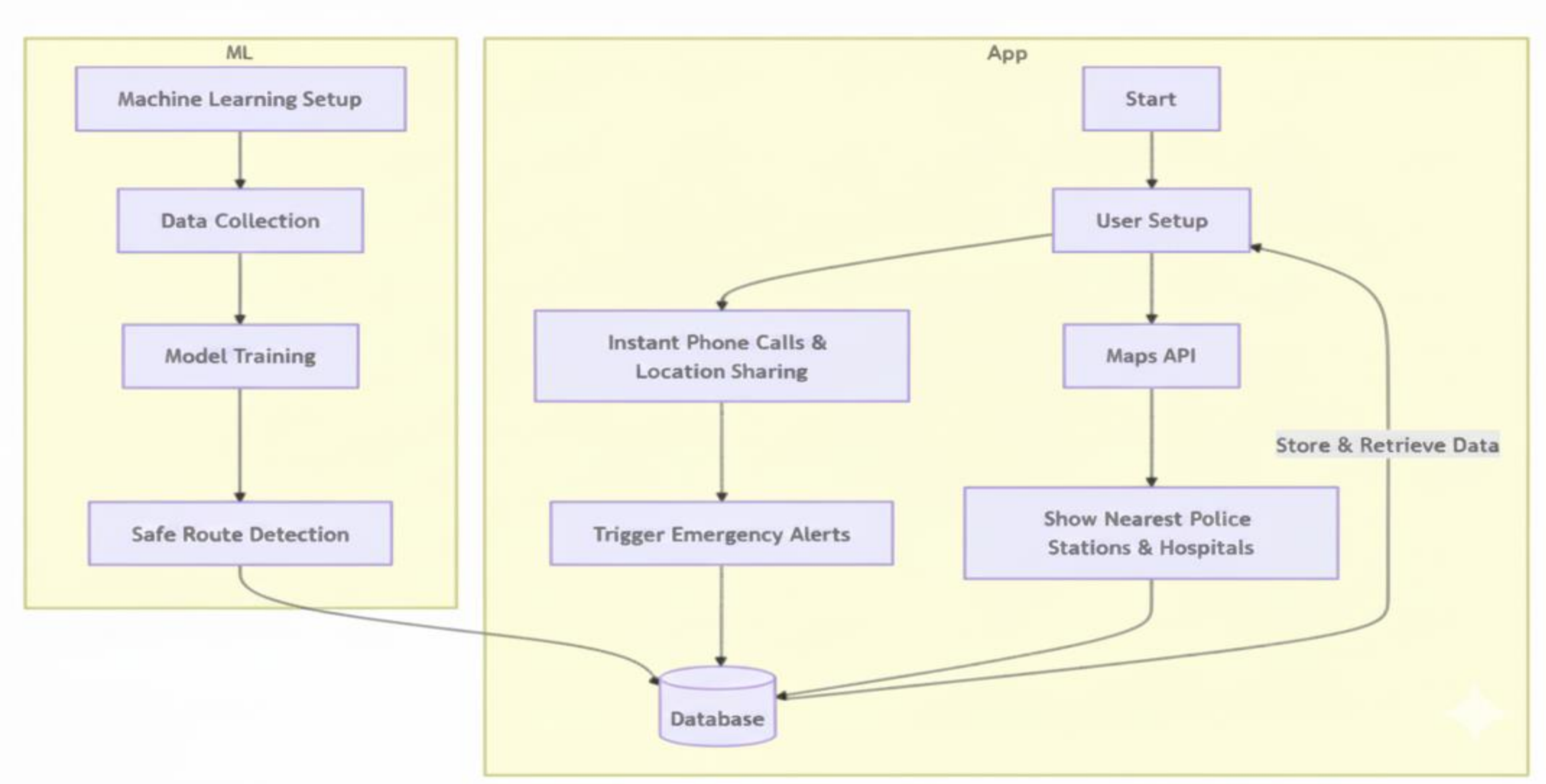
- **Unit 2.3 –"Safest Route" Algorithm (Simulation):** Utilizes the model’s output to simulate and suggest the safest possible route between two points, balancing distance and safety metrics.

# DESIGN SOLUTION FOR VARIOUS IDENTIFIED MODULES/UNITS

**ML Pipeline Flow:**

- **Data Collection (News):** Gathers incident reports from news articles and user-submitted data.
- **Data Processing:** NLP techniques (NER + tokenization) extract meaningful entities like location, time, and crime type.
- **Model Training:** XGBoost model trained on the processed dataset to predict the safety level of an area.
- **Safe Route Detection:** The algorithm compares different paths and highlights the safest one using safety scores and map overlays.

# DESIGN SOLUTION FOR VARIOUS IDENTIFIED MODULES/UNITS




BLOCK DIAGRAM

# NECESSARY TOOLS FOR IMPLEMENTATION

**1. Android App Development**

IDE: Android Studio

Language: Java

APIs: Google Maps SDK, Google Places API

Networking: Volley / Retrofit

# NECESSARY TOOLS FOR IMPLEMENTATION

**2. Machine Learning Engine (Colab Notebook)**

Language: Python

Core ML: Scikit-learn, XGBoost

NLP: spaCy (for NER), TensorFlow/Keras (for LSTM)

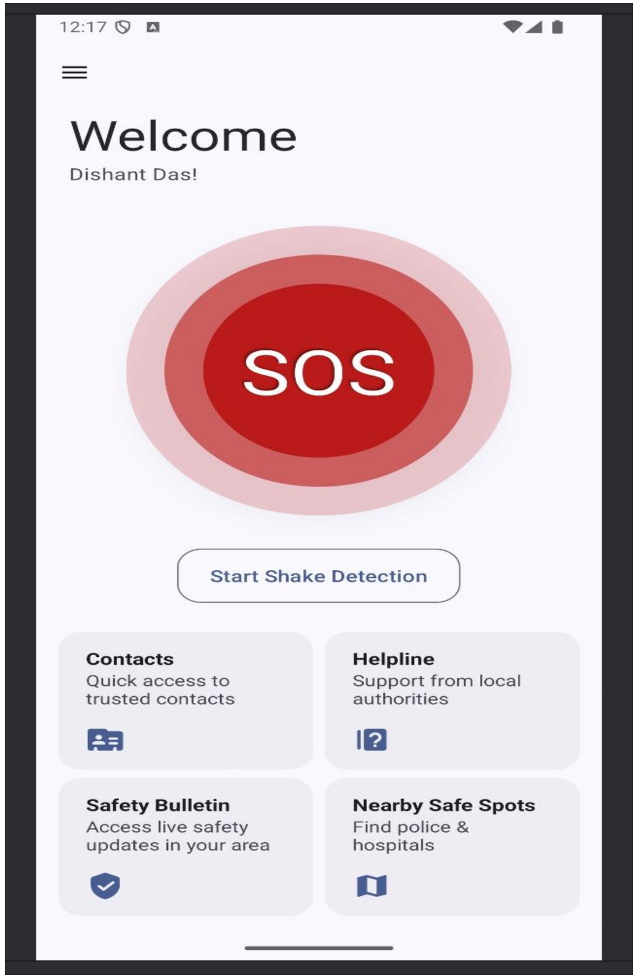
Data Viz: Plotly, Folium

Routing: OpenRouteService (ORS)

# DEVELOPMENT OF VARIOUS IDENTIFIED MODULES/UNITS

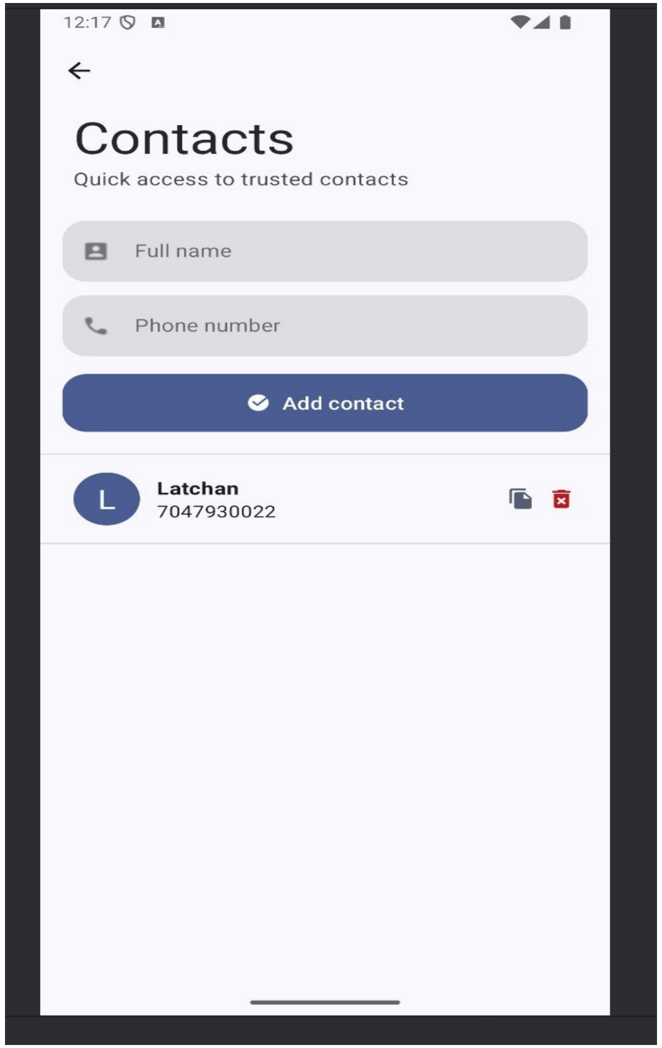
- **DEVELOPMENT**:
  ANDROID APP
  (CORE UI)

- Built a modern, card-based dashboard for all key features.



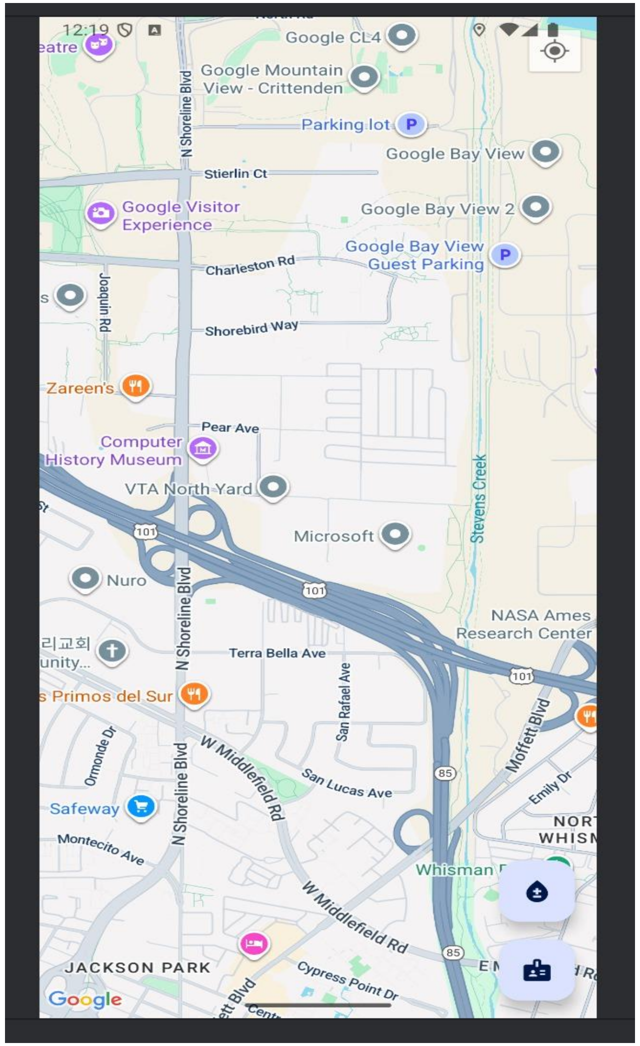
# DEVELOPMENT OF VARIOUS IDENTIFIED MODULES/UNITS

- Implement a trusted contacts, module for Managing emergency contacts.



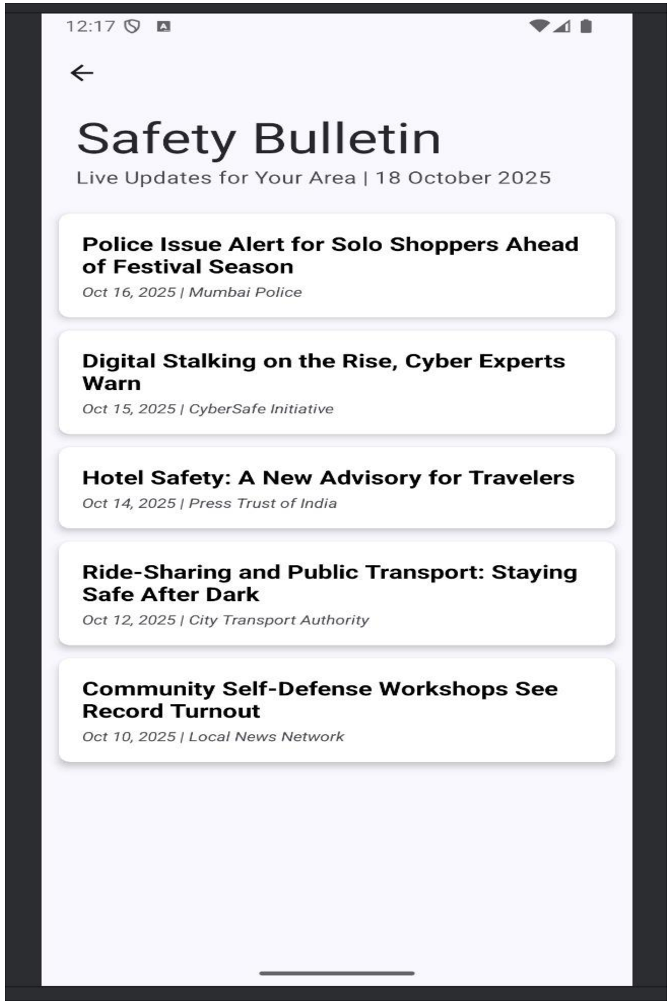
# DEVELOPMENT OF VARIOUS IDENTIFIED MODULES/UNITS

- Developed the "Nearby Safe Spots" module using the Google Maps SDK to show the user's location and find police/hospitals.



# DEVELOPMENT OF VARIOUS IDENTIFIED MODULES/UNITS

- Created an interactive "Safety Bulletin" to display live updates/news.



# Development: NLP Pipeline (Data)

- Loaded a custom dataset of 5000 news articles.

Load the data

**Reasoning**: Import the pandas library and load the CSV file into a DataFrame, then display the head and info of the DataFrame to understand its structure and content.

```
import pandas as pd

df = pd.read_csv('delhi_crime_news_final_2015_2024.csv')
display(df.head())
display(df.info())
```

| | crime_type | article | crime_severity |
|---|---|---|---|
| 0 | Theft | On the morning of 2019-07-14, a brazen theft u... | 0 |
| 1 | Assault | The late-night assault in Dwarka on 2017-11-03... | 1 |
| 2 | Robbery | On 2020-02-19, a high-profile robbery occurred... | 1 |
| 3 | Burglary | A quiet residential burglary on 2018-08-22 in ... | 1 |
| 4 | Vandalism | In 2016-05-12, a spree of vandalism hit Chandn... | 0 |

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 3 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   crime_type      1000 non-null   object
 1   article         1000 non-null   object
 2   crime_severity  1000 non-null   int64
dtypes: int64(1), object(2)
memory usage: 23.6+ KB
None
```

# Development: NLP Pipeline (Model)

- Built a Bidirectional LSTM model with TensorFlow/ Keras to understand the context of articles and classify them by crime type.

Build the LSTM model

**Reasoning**: Import necessary libraries, create a sequential model, add layers with appropriate parameters, compile the model, and display the summary.

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, SpatialDropout1D, Bidirectional
from tensorflow.keras.callbacks import EarlyStopping

model = Sequential()
model.add(Embedding(input_dim=vocabulary_size, output_dim=128, input_length=max_sequence_length))
model.add(SpatialDropout1D(0.3))
model.add(Bidirectional(LSTM(128, dropout=0.3, recurrent_dropout=0.3)))
model.add(Dense(len(y_train.unique()), activation='softmax'))

model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])

model.summary()
```

```
/usr/local/lib/python3.12/dist-packages/keras/src/layers/core/embedding.py:97: UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(
```

**Model: "sequential_1"**

| Layer (type) | Output Shape | Param # |
|---|---|---|
| embedding (Embedding) | ? | 0 (unbuilt) |
| spatial_dropout1d (SpatialDropout1D) | ? | 0 |
| bidirectional (Bidirectional) | ? | 0 (unbuilt) |
| dense (Dense) | ? | 0 (unbuilt) |

**Total params:** 0 (0.00 B)
**Trainable params:** 0 (0.00 B)
**Non-trainable params:** 0 (0.00 B)

# Development: NLP Pipeline (Training)

## Train the model

### Subtask:

Train the LSTM model on the training data.

+ Code + Text

**Reasoning**: Train the compiled LSTM model using the training data for a specified number of epochs and batch size, and store the training history.

```
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)

history = model.fit(X_train, y_train, epochs=10, batch_size=32, validation_split=0.2, callbacks=[early_stopping])
```

```
Epoch 1/10
19/19 ━━━━━━━━━━━━━━━━━━━━ 51s 3s/step - accuracy: 0.9526 - loss: 0.1668 - val_accuracy: 0.8000 - val_loss: 0.5327
Epoch 2/10
19/19 ━━━━━━━━━━━━━━━━━━━━ 51s 3s/step - accuracy: 0.9667 - loss: 0.1354 - val_accuracy: 0.8133 - val_loss: 0.6401
Epoch 3/10
19/19 ━━━━━━━━━━━━━━━━━━━━ 82s 3s/step - accuracy: 0.9758 - loss: 0.0920 - val_accuracy: 0.8067 - val_loss: 0.6207
Epoch 4/10
19/19 ━━━━━━━━━━━━━━━━━━━━ 51s 3s/step - accuracy: 0.9848 - loss: 0.0565 - val_accuracy: 0.8000 - val_loss: 0.6536
Epoch 5/10
19/19 ━━━━━━━━━━━━━━━━━━━━ 51s 3s/step - accuracy: 0.9804 - loss: 0.0674 - val_accuracy: 0.7600 - val_loss: 0.6557
Epoch 6/10
19/19 ━━━━━━━━━━━━━━━━━━━━ 82s 3s/step - accuracy: 0.9793 - loss: 0.0766 - val_accuracy: 0.7800 - val_loss: 0.9531
```

# Development: Geospatial Prediction

- Created a ColumnTransformer to preprocess a synthetic Delhi dataset.
- Implemented an XGBoost classifier, a highly effective model for this type of structured data.
- The default model achieved a strong baseline accuracy of 97%.

```python
print("\n--- Training XGBoost Model ---")

# Create the XGBoost pipeline
xgb_pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                               ('classifier', xgb.XGBClassifier(
                                   objective='multi:softmax',
                                   num_class=3,
                                   use_label_encoder=False,
                                   eval_metric='mlogloss',
                                   random_state=42
                               ))])

# Train the model
xgb_pipeline.fit(X_train, y_train)

# Evaluate the model
y_pred_xgb = xgb_pipeline.predict(X_test)
accuracy_xgb = accuracy_score(y_test, y_pred_xgb)
model_results['XGBoost'] = accuracy_xgb

print(f"--- Results for XGBoost ---")
print(f"Accuracy: {accuracy_xgb:.4f}")
print(classification_report(y_test, y_pred_xgb, target_names=['1 (Safe)', '2 (Moderate)', '3 (Unsafe)']))
print("-" * 40 + "\n")
```

```
--- Training XGBoost Model ---
--- Results for XGBoost ---
Accuracy: 0.9700
              precision    recall  f1-score   support

    1 (Safe)       1.00      0.97      0.99        37
2 (Moderate)       0.91      1.00      0.95        31
  3 (Unsafe)       1.00      0.94      0.97        32

    accuracy                           0.97       100
   macro avg       0.97      0.97      0.97       100
weighted avg       0.97      0.97      0.97       100

----------------------------------------
```

# Validation: NLP Pipeline

- The trained LSTM model was evaluated on the unseen test data.
- Achieved a final accuracy of 84.8% in classifying the crime type/severity from raw text.
- This validates our NLP pipeline as an effective tool for automated data gathering.

## Evaluate the model

**Reasoning**: Evaluate the trained model on the testing data and print the results.

```
loss, accuracy = model.evaluate(X_test, y_test)
print(f'Test Loss: {loss:.4f}')
print(f'Test Accuracy: {accuracy:.4f}')
```

```
8/8 ━━━━━━━━━━━━━━━━━━━━ 4s 461ms/step - accuracy: 0.8317 - loss: 0.4865
Test Loss: 0.4409
Test Accuracy: 0.8480
```

## Make predictions

**Reasoning**: Use the trained model to predict the crime severity on the test data and convert the predictions to class labels.

```
predictions = model.predict(X_test)
predicted_classes = tf.argmax(predictions, axis=1).numpy()
```

```
8/8 ━━━━━━━━━━━━━━━━━━━━ 6s 653ms/step
```

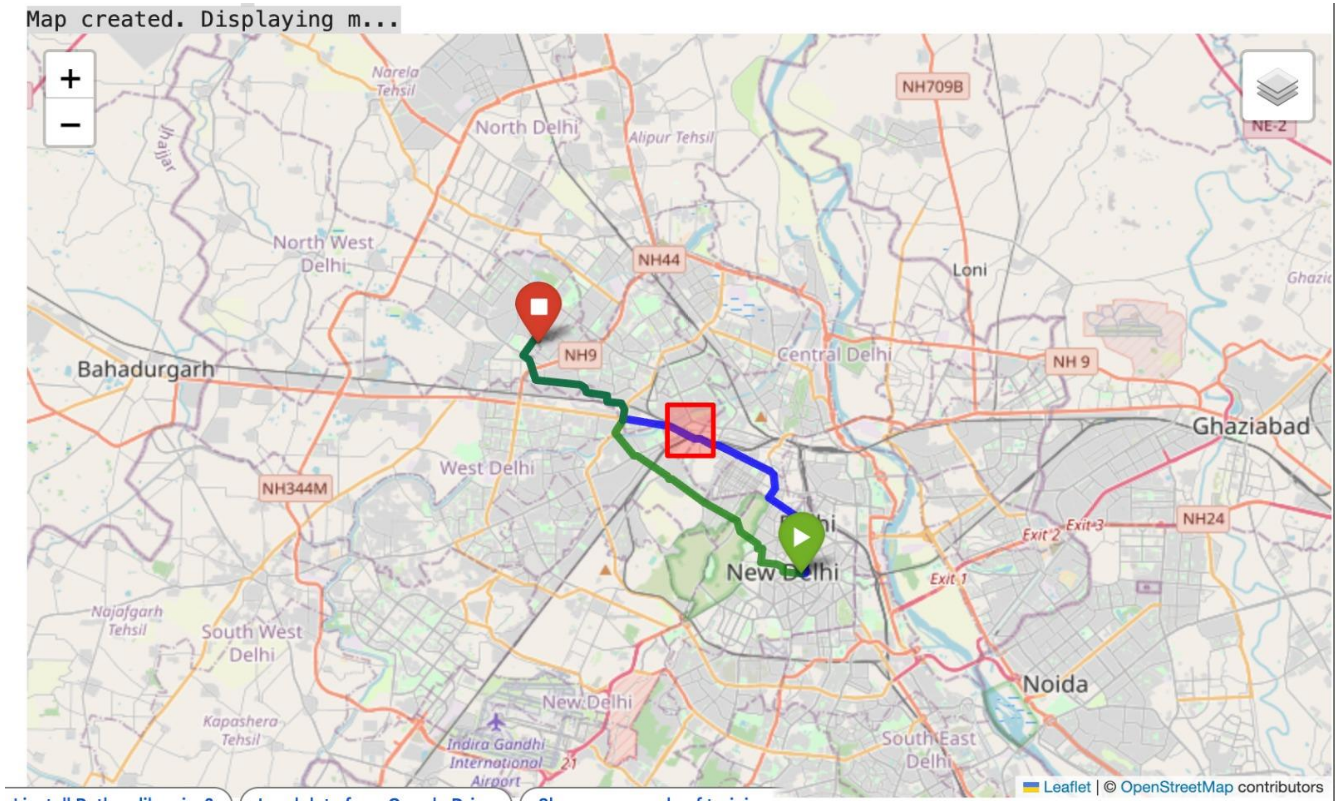
# Validation: Geospatial Prediction

- The XGBoost model was hyperparameter-tuned using RandomizedSearchCV for maximum performance.
- The final tuned model achieved a 99% accuracy on the unseen test data.
- The classification report shows the model is extremely reliable at identifying all three safety categories.

```
--- 📈 Starting Hyperparameter Tuning for XGBoost... ---
Fitting 3 folds for each of 25 candidates, totalling 75 fits

--- Best Parameters for XGBoost ---
{'classifier__subsample': 0.7, 'classifier__n_estimators': 500, 'classifier__max_depth': 10, 'classifier__learning_rate': 0.1, 'classifier__colsam

Tuned XGBoost Accuracy: 0.9900

              precision    recall  f1-score   support

    1 (Safe)       1.00      1.00      1.00        37
2 (Moderate)       0.97      1.00      0.98        31
  3 (Unsafe)       1.00      0.97      0.98        32

    accuracy                           0.99       100
   macro avg       0.99      0.99      0.99       100
weighted avg       0.99      0.99      0.99       100

/usr/local/lib/python3.12/dist-packages/xgboost/training.py:183: UserWarning: [17:58:05] WARNING: /workspace/src/learner.cc:738:
Parameters: { "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
```

# Validation: "Safest Route" Simulation

- This simulation demonstrates the practical value of our ML model.
- Blue Line (Fastest Route): The default route, which passes through a high-risk (red) zone.
- Green Line (Safest Route): The custom route that uses our model's data to navigate around the danger.



# VALIDATION OF DEVELOPED MODULES

## App Validation:

- "Nearby Safe Spots" map with blue (police) and red (hospital) pins
- "Safety Bulletin" displaying news list

# Literature Survey: Key Findings Content

Finding 1: Most Apps are Only Reactive: The majority of existing apps are simple SOS/SMS buttons. They lack the "proactive" (preventative) features that our project introduces.

Finding 2: "Unsafe Zones" are Static: Some apps have "unsafe zones", but they are pre-defined and not dynamic. They don't use real-time data or machine learning.

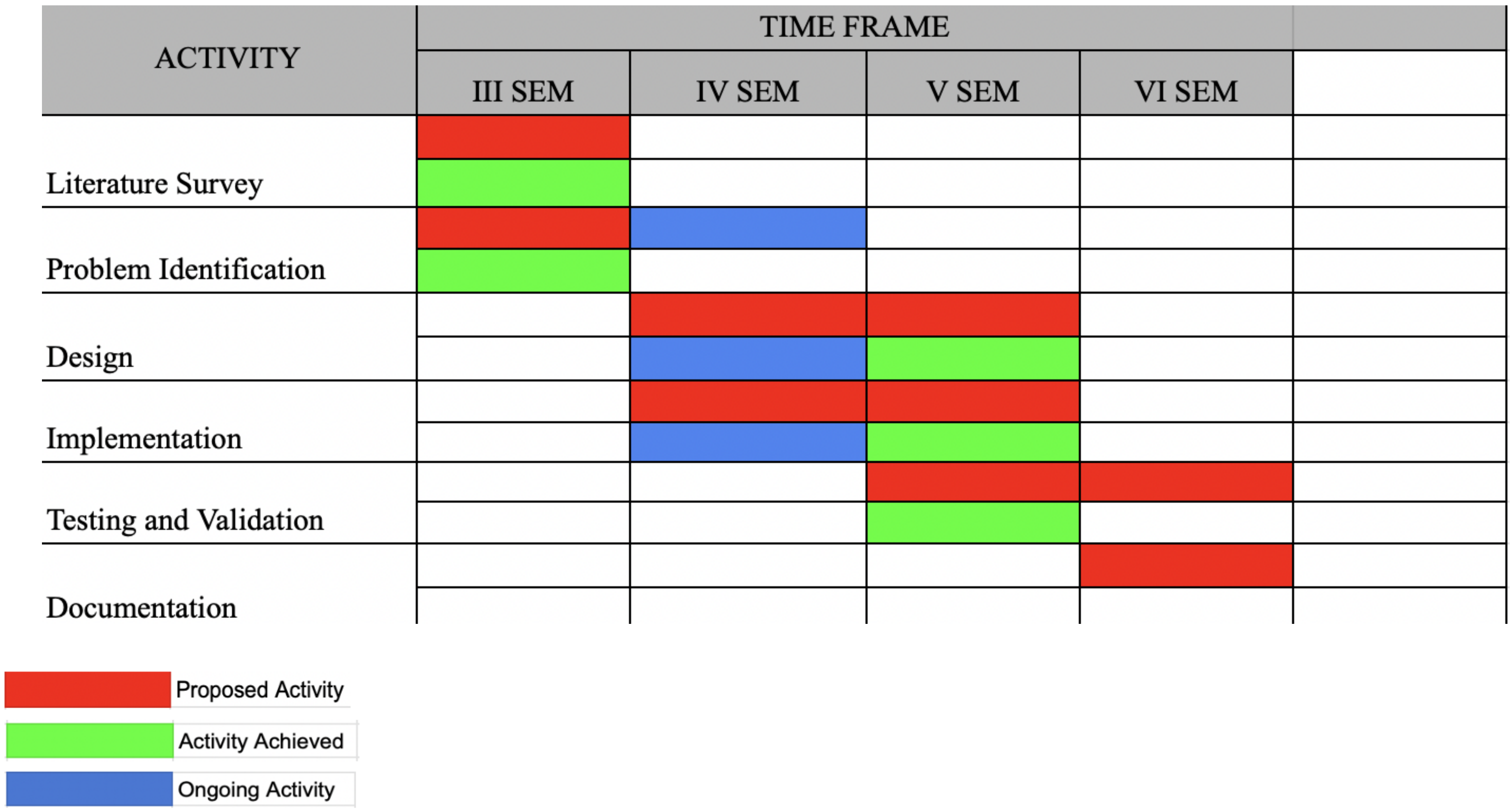
Finding 3: "Safe Routing" is Theoretical: Advanced research on "safe routing" exists, but it's often a theoretical framework, not a practical feature integrated into a complete, working app

Our Project's Contribution: Our project fills these gaps by combining a best-in-class reactive app with a proactive ML engine that predicts safety scores and simulates a dynamic "safest route."

# SEMINARS AND WORKSHOPS

- The Complete Web Development bootcamp with Dart by Angela Yu (60 hours)
- Machine Learning Specialization by Andrew Ng (10 hours)
- IoT Application Development (30 hours)
- Applied Machine Learning by NIT Kurukshetra (30 hours)
- Hands-on Python Training by Encoders (7 hours)
- Web dev bootcamp by Encoders (8 hours)
- Data Science from Geeks For Geeks(1 month)

# GANTT CHART


ACTIVITY
TIME FRAME
III SEM
IV SEM
V SEM
VI SEM
Literature Survey
Problem Identification
Design
Implementation
Testing and Validation
Documentation
Proposed Activity
Activity Achieved
Ongoing Activity

# REFERENCES


[1] K. Srinivas, S. Gothane, C. S. Krithika, Anshika, and T. Susmitha, "Android App for Women Safety," *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, vol. 9, issue VI, pp. 232–236, 2021.

[2] Suriya Prakash S., Nithya S., Rithika S., and Sangavi S., "Smartphone-Based Women Safety Application with Location Tracking and Emergency Response," *International Journal for Research Trends and Innovation (IJRTI)*, vol. 10, issue 5, pp. 180–184, May 2025.

[3] Avishi Sharma, Lakshika Sarupria, Siddhika Dhabhai, Viral Jain, Yashveer Singh Deora, and Charu Kavadia, "Mobile Application on Women Safety," *International Advanced Research Journal in Science, Engineering and Technology (IARJSET – ICMART 2023 Conference)*, pp. 45–48, 2023.

[4] Sharon Levy, Wenhan Xiong, Elizabeth Belding, and William Yang Wang, "SafeRoute: Learning to Navigate Streets Safely in an Urban Environment," *ACM / arXiv preprint arXiv:1811.01147*, pp. 1–10, 2018.

[5] Cashify Editorial Team, "Women's Day: Top Safety Apps," *Cashify India*, available at: https://www.cashify.in/womens-day-top-safety-apps, accessed March 8, 2022.

THANK YOU.