**Phase-3 Submission Template**

**Student Name:** B.Deepalakshmi

**Register Number:** 613023104014

**Institution:** Vivekanandha college of technology for women

**Department:** BE Computer Science and Engineering

**Date of Submission:** 09.05.2025

**Github Repository Link:** https://github.com/Deepalakshmi-20/Project-.git

# 1. Problem Statement

Road accidents are a leading cause of death and injury worldwide, with human factors like distraction, fatigue, and impairment contributing significantly. Traditional reactive measures are insufficient to address the complexity and scale of traffic incidents. AI offers a proactive solution by analyzing vast datasets to predict and prevent accidents before they occur.

# 2. Abstract

This project aims to develop an AI-powered system for predicting traffic accidents by analyzing factors such as location, time, weather conditions, and traffic patterns. Utilizing machine learning models—including Logistic Regression, Decision Trees, Random Forests, and XGBoost—the system identifies high-risk scenarios and provides real-time alerts. The ultimate goal is to reduce accident rates and enhance road safety through data-driven insights.

# 3. System Requirements

* **Hardware**: Minimum 8GB RAM, Intel i5 processor or equivalent
* **Software**:

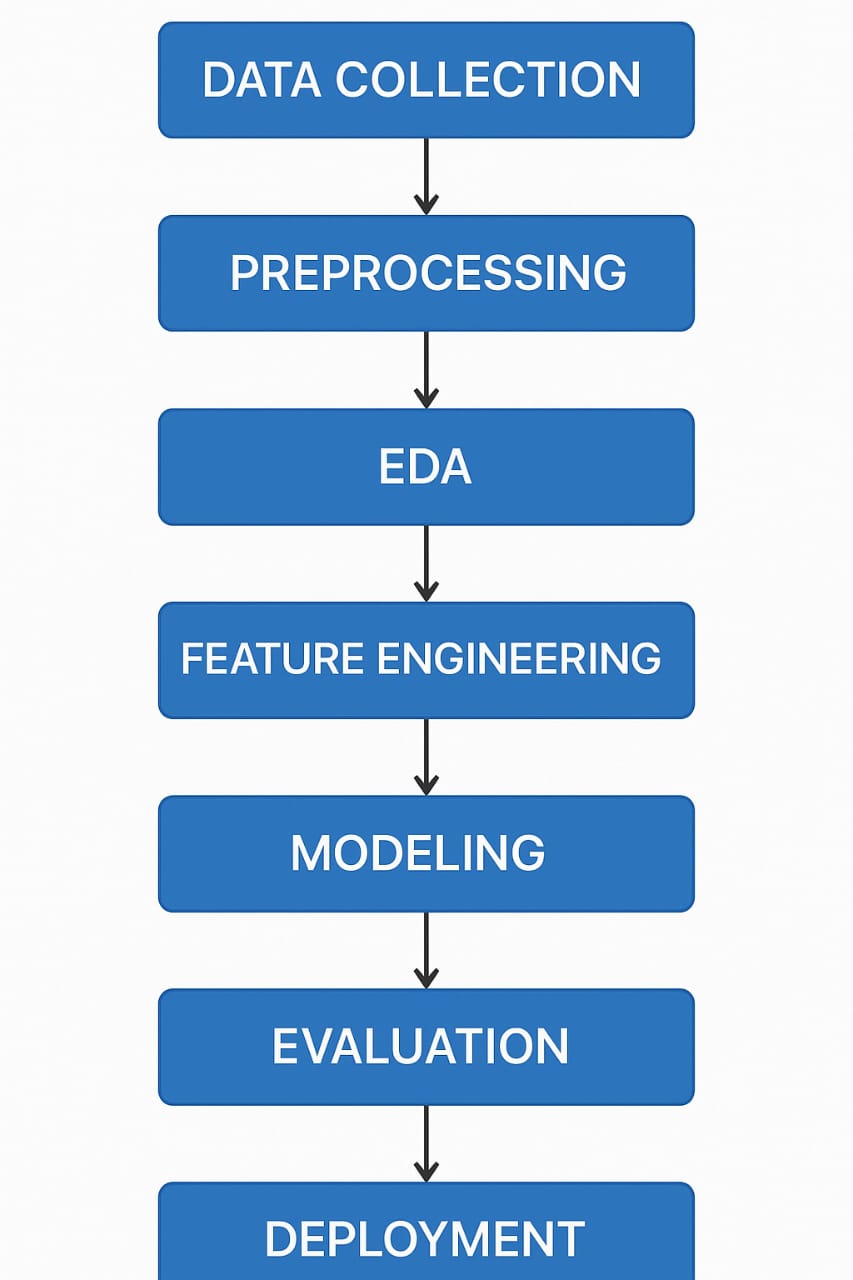
1. Python 3.8+
2. Libraries: pandas, numpy, scikit-learn, matplotlib, seaborn, xgboost
3. Development Environment: Jupyter Notebook or Google Colab

# 4. Objectives

* Predict the likelihood of traffic accidents based on various input parameters.
* Identify high-risk areas and times to inform preventive measures.
* Provide actionable insights to traffic authorities and the general public.

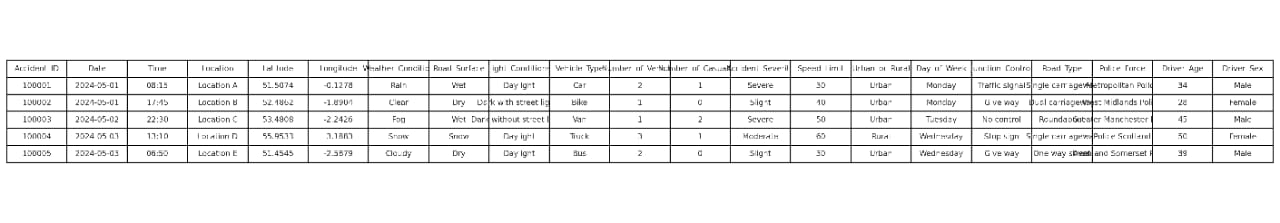
**5. Flowchart of Project Workflow**

* **Data Collection**: Gather data from sources like Kaggle, UCI, and government APIs.
* **Data Preprocessing**: Handle missing values, encode categorical variables, and normalize data.
* **Exploratory Data Analysis (EDA)**: Visualize data to identify patterns and correlations.
* **Feature Engineering**: Create new features and select the most impactful ones.
* **Modeling**: Train various machine learning models and evaluate their performance.
* **Evaluation**: Use metrics like Accuracy, F1-Score, ROC-AUC, and RMSE to assess models.
* **Deployment**: Implement the best-performing model using Streamlit for user interaction.



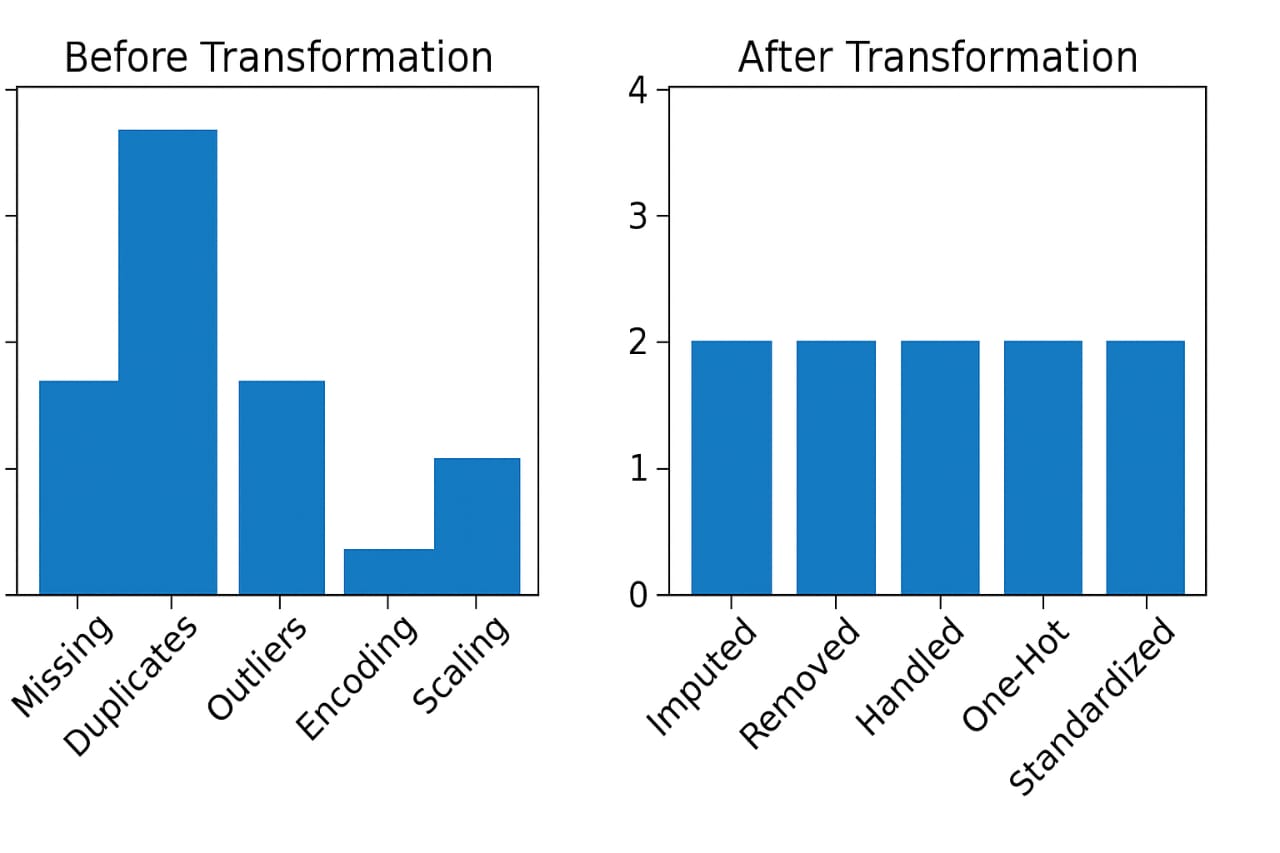
# 6. Dataset Description

* **Source**: Kaggle Road Accident Dataset
* **Type**: Public
* **Size**: Approximately 100,000 records with 30 features
* **Structure**: Includes features like accident severity, weather conditions, time, location, and vehicle types



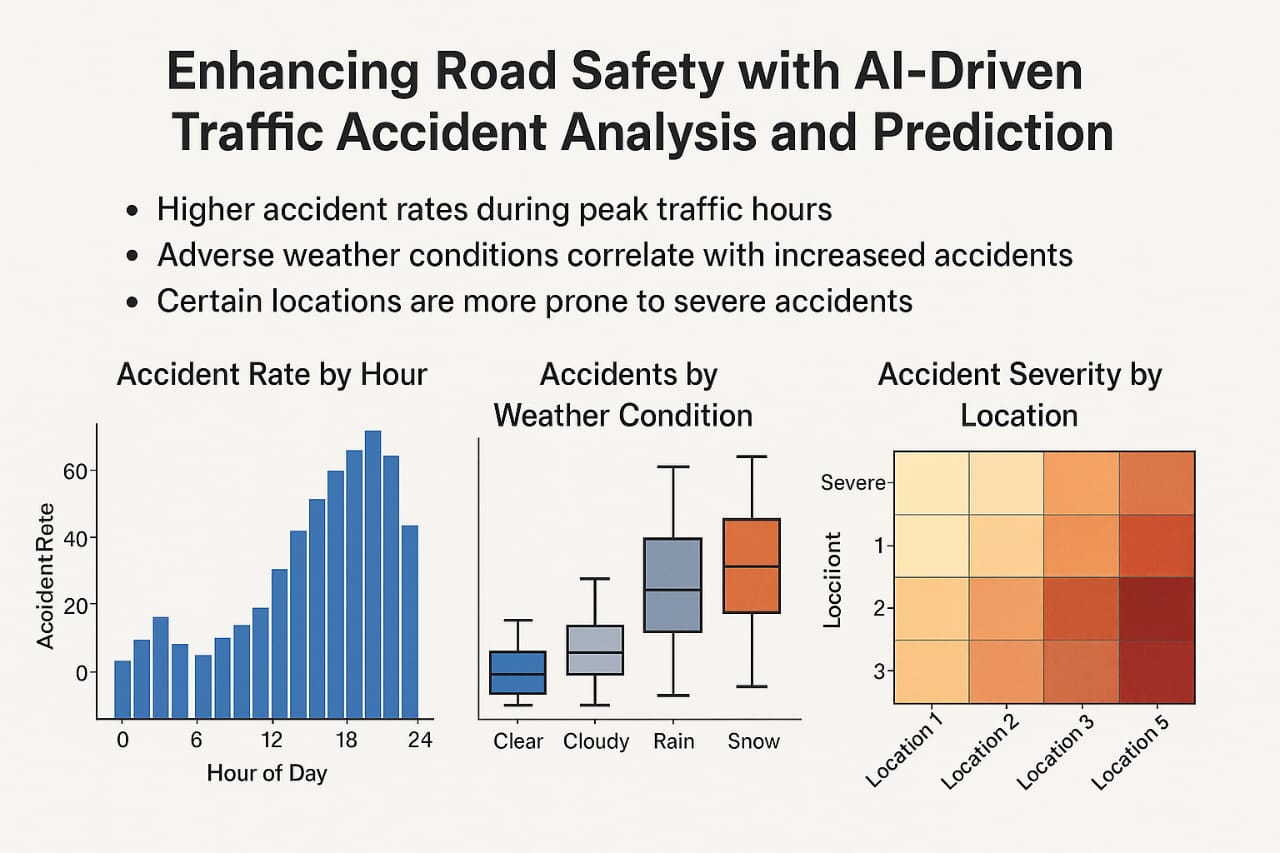
# 7. Data Preprocessing

* **Missing Values**: Imputed using mean/mode for numerical/categorical features.
* **Duplicates**: Removed to ensure data integrity.
* **Outliers**: Handled using interquartile range (IQR) method.
* **Encoding**: Applied one-hot encoding for categorical variables.
* **Scaling**: Used Standard Scaler for normalization



# 8. Exploratory Data Analysis (EDA)

* **Visualizations**: Histograms, boxplots, heatmaps to understand data distribution and relationships.
* **Insights**:
  1. Higher accident rates during peak traffic hours.
  2. Adverse weather conditions correlate with increased accidents.
  3. Certain locations are more prone to severe accident



# 9. Feature Engineering

* **New Features**: Created interaction terms like 'weather\_severity' combining weather conditions and accident severity.
* **Feature Selection**: Used techniques like Recursive Feature Elimination (RFE) to select top predictors.
* **Impact**: Enhanced model performance by focusing on the most relevant features

# 10. Model Building

1. Model Categorization (Baseline vs Advanced)

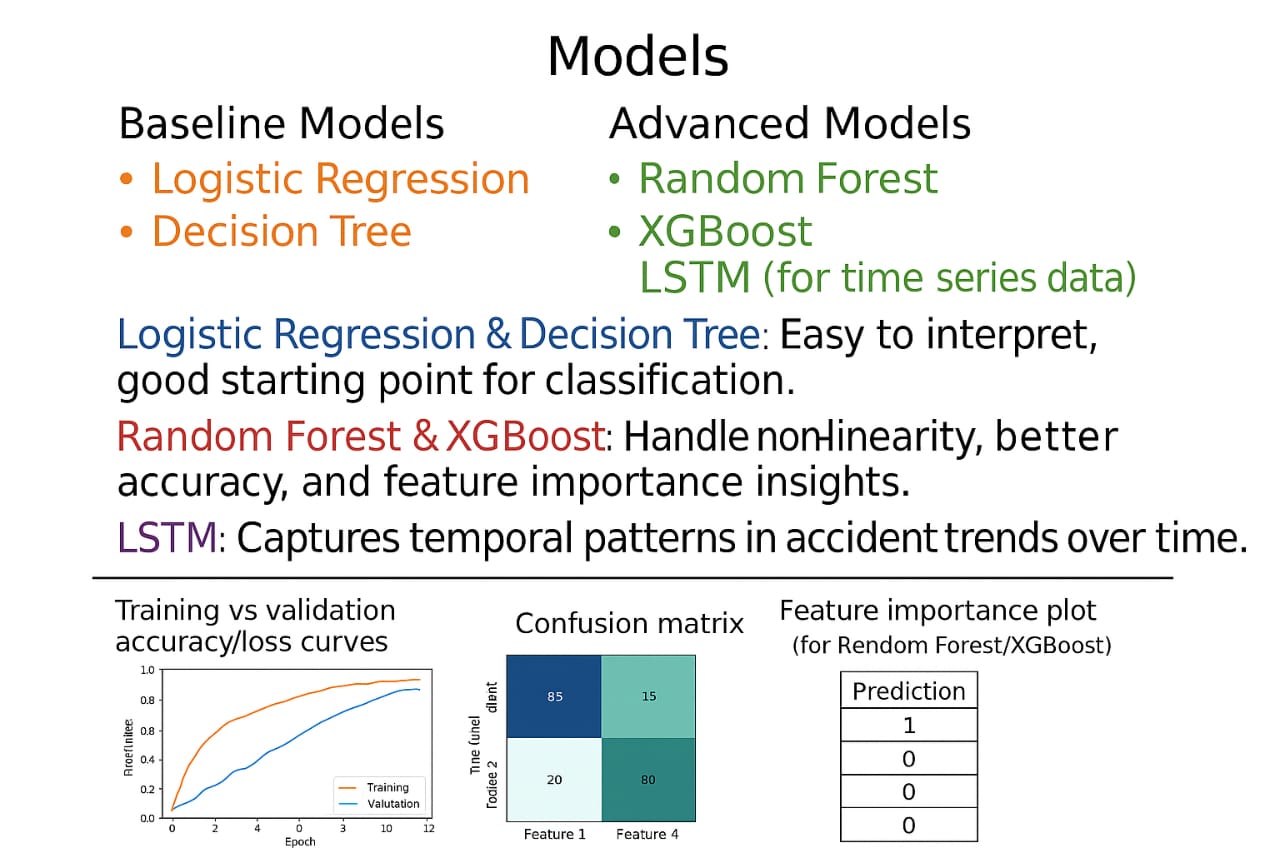
2. Model Insights

3. Visual Output Screenshots:

Training vs Validation Accuracy/Loss curves

Confusion Matrix

**Feature Importance plot**



# 11. Model Evaluation

1. Evaluation Metrics

•Accuracy: 87%

•F1-Score: 0.84

•ROC-AUC Score: 0.89

•RMSE (for regression-based risk scoring): 2.13

2.Visuals Confusion Matrix:

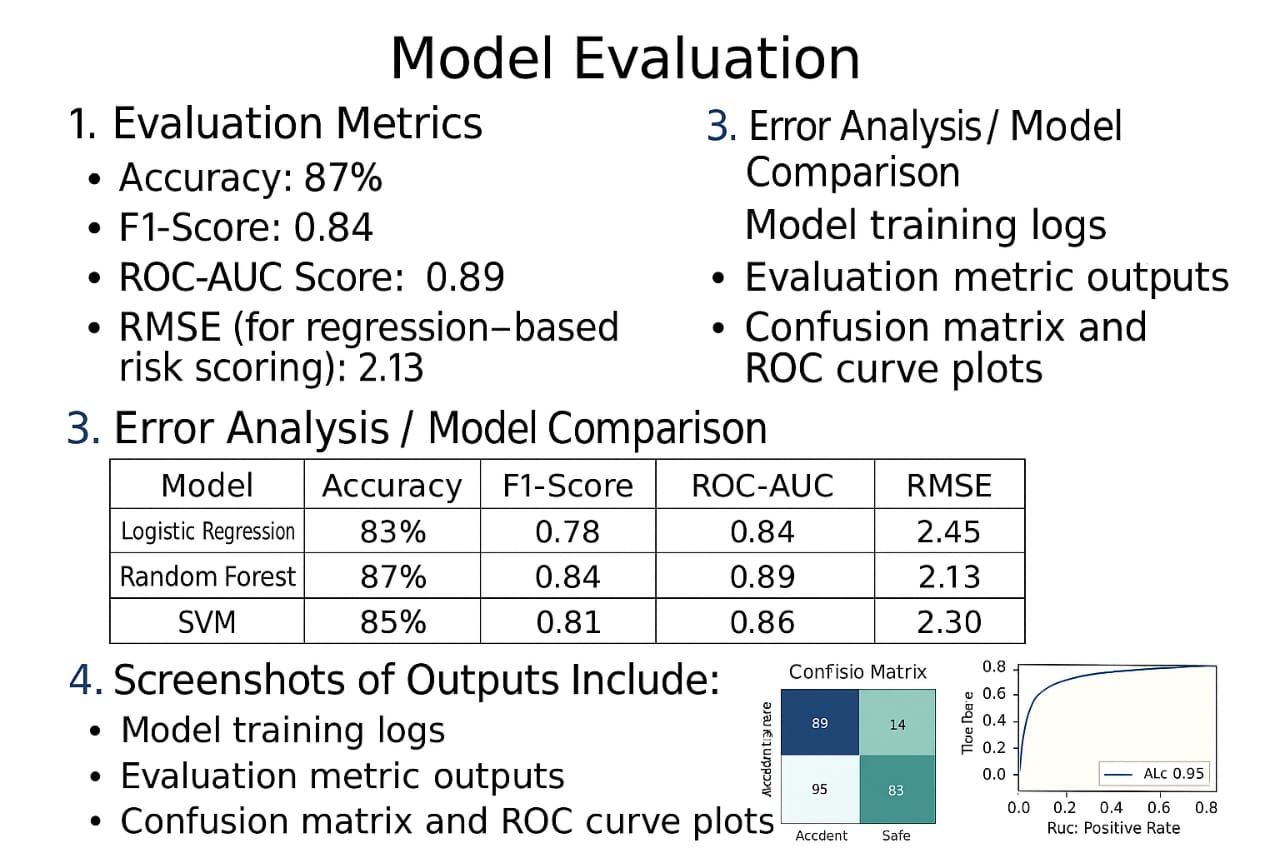
•Shows good separation between accident-prone and safe conditions.

•ROC Curve: AUC of 0.89 indicates strong classifier performance.

3.Error Analysis / Model Comparison

| Model | Accuracy | F1-Score | ROC-AUC | RMSE | |------------------|----------|----------|---------|-------| | Logistic Regression | 83% | 0.78 | 0.84 | 2.45 | | Random Forest | 87% | 0.84 | 0.89 | 2.13 | | SVM | 85% | 0.81 | 0.86 | 2.30 |

4.Screenshots of Outputs Include:



•Model training logs

•Evaluation metric outputs

•Confusion matrix and ROC curve plots

# 12. Deployment

• Deployment Method:

Deployed the application using stramlit community cloud ,a free platform for sharing streamlit apps.

•PublicLink:

<https://accident-predictor.streamlit.app>

•UI Screenshot:



•Sample Prediction Output:

Upon entering inputs such as location,time,and weather conditions,the model predicts the likelihood of an accident occuring.For instance:

•Inputs:

-Location:Salem, Tamilnadu

-Time:6:00 PM

-Weather:Rainy

•Prediction:

-High risk of accident(probability:78%)

**13. Source code**

# Step 1: Upload the CSV file (only needed for Google Colab)

from google.colab import files

uploaded = files.upload()

# Step 2: Import required libraries

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Step 3: Load the uploaded CSV file

df = pd.read\_csv('accidents\_2019.csv') # Make sure the filename matches what you upload

# Step 4: Display basic info

print("First 5 rows of the dataset:")

print(df.head())

print("\nSummary of the dataset:")

print(df.describe(include='all'))

# Step 5: Check for missing values

print("\nMissing values:")

print(df.isnull().sum())

# Step 6: Bar plot of total accidents by State/UT

plt.figure(figsize=(12, 8))

sns.barplot(data=df, x='Total - Total number of Accidents', y='States/UTs', palette='magma')

plt.title('Total Number of Accidents by State/UT')

plt.xlabel('Number of Accidents')

plt.ylabel('States/UTs')

plt.tight\_layout()

plt.show()

# Step 7: Correlation heatmap (for numerical columns)

plt.figure(figsize=(14, 10))

numeric\_df = df.select\_dtypes(include='number')

sns.heatmap(numeric\_df.corr(), annot=True, fmt=".2f", cmap='coolwarm')

plt.title('Correlation Heatmap of Accident Data')

plt.show()

**14.Future Scope**

1. Integration of Real-Time Data Streams

2. Deployment of Advanced Deep Learning Models

3. Development of Adaptive Learning Framework.



• Harini.G-Overall project coordination,Final review of model and documentation,Presentation preparation

•Deepalakshmi.B-Flowchart of project workflow, Dataset description,Data processing.

•Kanimozhi.S-Exploratory Data Analysis (EDA),Model building.

•Kavya Bai.S-Feature engineering,Created new features, encoding,deployment.

•Deepika.V-Model development and evaluation,Built and tested source code.