A PROJECT REPORT ON

CHAT APPLICATION WITH SPAM DETECTION

SUBMITTED TO THE
CUMMINS COLLEGE OF ENGINEERING FOR WOMEN,
KARVENAGAR, PUNE
(An autonomous institute affiliated to Savitribai
Phule Pune university),

IN THE PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE AWARD OF THE
DEGREE

OF

BACHELOR OF TECHNOLOGY

(COMPUTER ENGINEERING)

SUBMITTED BY

PRAJAKTA DEOKULE	**C22019221332**	**4329**
MAITREYEE JOSHI	**C22019221355**	**4351**
TANVI KATAKKAR	**C22020222305**	**4376**
LAVANNYA PATIL	**C22020222306**	**4377**

DEPARTMENT OF COMPUTER ENGINEERING

MKSSS'S CUMMINS COLLEGE OF ENGINEERING FOR WOMEN

KARVENAGAR, PUNE 411052
SAVITRIBAI PHULE PUNE UNIVERSITY

A.Y. 2022 -2023

CERTIFICATE

This is to certify that the project report entitles

“ CHAT APPLICATION WITH SPAM DETECTION”

Submitted By

PRAJAKTA DEOKULE	**C22019221332**	**4329**
MAITREYEE JOSHI	**C22019221355**	**4351**
TANVI KATAKKAR	**C22020222305**	**4376**
LAVANNYA PATIL	**C22020222306**	**4377**

is a bonafide student of this institute and the work has been carried out by her under the supervision of **Prof. Rakhi A. Dongaonkar** and it is approved for the partial fulfillment of the requirement ofcummins college of engineering for women, karvenagar, pune
(An autonomous institute affiliated to Savitribai Phule Pune University.),

for the award of the degree of **Bachelor of Technology** (Computer Engineering).

(Prof. Rakhi Dongaonkar)
Guide,
Department of Computer Engineering

(Dr. S. M. Kelkar)
Head,
Department of Computer Engineering

(Dr. M.B. Khambete)
Director,
Cummins College of Engineering for Women

Place : Pune

Date :

BLINK ADS

25 May 2023

To,
Head of Department,
Department of Computer Engineering,
MKSSS's Cummins College of Engineering For Women,
Karve Nagar Pune 411052

Subject : Regarding B. Tech Project Completion

This is certify that the project title **Chat Application with Spam Detection** is successfully completed during the academic year 2022-23, by the following mentioned students of MKSSS's Cummins College of Engineering For Women from Computer engineering , Name of the students :-

Name	**Registration Number**	**Roll Number**
Lavannya Patil	C22020222306	4377
Prajakta Deokule	C22019221332	4329
Tanvi Katakkar	C22020222305	4376
Maitreyee Joshi	C22019221355	4351

Regards

Prabhat Mishra
Founder & CEO
Blink Ads
www.blinkads.in
9175615193

ACKNOWLEDGMENT

Our profound gratitude goes to our wonderful guide, **Prof. Rakhi Dongaonkar** for her invaluable support, patience, time and guidance in seeing us to the completion of this project work. Also our gratitude goes to **Dr. Supriya Kelkar**, Head, Department of Computer Engineering for top of the state labs for project work.

We also extend gratitude and appreciation to **Dr. M.B. Khambete**, Principal, Cummins College of Engineering for Women, Pune.
We also wish to acknowledge the great support and advice given to us by our **BlinkAds** mentor **Mr. Prabhat Mishra**.

We also wish to acknowledge the great support of our parents, siblings, who have been a source of inspiration towards our academic pursuit.

PRAJAKTA DEOKULE
MAITREYEE JOSHI
TANVI KATAKKAR
LAVANNYA PATIL

ABSTRACT

SMS messaging is now more common than ever thanks to recent advancements in mobile technology. As a result, there are now more spam communications are happening via mobile devices. Even though emails are the primary source of spam worldwide, SMS services are now not far behind in their contribution to the problem. No one wants their mobile devices to be inundated with spam messages. Numerous methods have been developed to identify and minimize spam messages, and research on the subject is constantly ongoing.

Spam classification in SMS messages is a difficult task.A lot of research has been done in this area using machine learning approaches including Naive Bayes (NB), Random Forest (RF), and Support Vector Machine (SVM). These techniques, however, perform only to a limited extent, thus failing to classify a diverse variety of spam messages correctly. So a thorough investigation is required to discover a more reliable and accurate way. To solve this, we put forth a technique called Long Short-Term Memory (LSTMs), an enhanced Recurrent Neural Network (RNN) structure with memory cells as part of its gating mechanism. In this project, we're creating a web and mobile application that will weed out spam messages.This report covers all aspects of system design, project implemenatation and the technology used along with the functional and non functional requirements of the project.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

1. AI - Artificial Intelligence
2. ML - Machine Learning
3. SRS - Software Requirements Specification
4. LAN - Local Area Network
5. OS - Operating System
6. HTML - Hypertext Markup Language
7. LSTM - Long SHort Term Memory
8. DFD - Data Flow Diagram
9. UI - User Interface
10. RN - React Native
11. DRY - Don’t Repeat Yourself
12. CRUD - Create, Read, Update, Delete
13. API - Application Programming Interface
14. NLP - Natural Language Processing

LIST OF FIGURES

LIST OF TABLES

01. INTRODUCTION

Messaging is a form of interpersonal communication, and there are billions of individuals that use mobile devices every day. However, the absence of adequate message filtering methods makes this sort of communication insecure. Spam contributes to this risk by making mobile SMS communication insecure.

One of the major issues with instant message systems is regarded to be spam. Spam is a junk message. Spam messages are distributed to users without their consent, and receivers find them annoying. It includes a variety of information, including phishing messages, advertisements for goods and services. These days, more mobile devices are being used in the environment for message communication, which has led to an increase in spam messages.

The users may occasionally suffer financial loss as a result of these spam messages. The cost of sending mail and messages is very low for senders but highly expensive for the recipients. Spam's cost can be calculated as the loss of human time and the loss of vital messages or communications. The important communications are impacted by these spam messages.

1.1 MOTIVATION

The emergence of technology has transformed the way we interact in profound ways. We communicate seamlessly while being remote using various devices and platforms like chat application that keep coming out. Spam messages have dramatically increased since chat applications first appeared. Spam is electronic messaging that is unsolicited, unwelcomed, and may contain malicious content.

The risks associated with spam communications for users are numerous: unwanted advertising, disclosure of the user's personal information, falling prey to fraud or financial schemes, being seduced into malware and phishing websites, unintentional exposure to offensive content, etc. Therefore, spam identification is urgently needed in order to enhance user experience, message quality, and to protect users from fraud.1

1.2 PROBLEM DEFINITION

Chat Application with Spam detection – the problem statement deals with creation of real-time chat application with spam detection capabilities that uses machine learning algorithms.

The chat application offers features including group and instant messaging, profile personalization, emoji support, slash commands, and presence and status tracking for users. The chat application has a spam detection feature driven by AI. The messages can be filtered out as spam or ham using the machine learning technique.

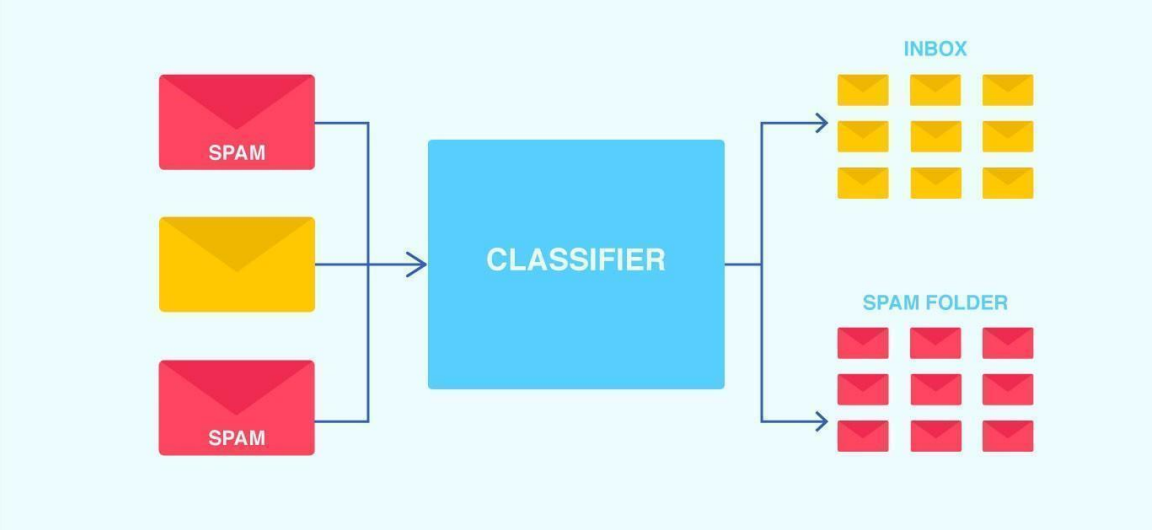


Fig 1.1 :- Classification of Spam and Ham Messages

1.3 OBJECTIVES

a) To develop a real time chat application:

   It consists of offering a group chat and instant messaging function that facilitates mass communication.

b) To integrate spam detection algorithm:

   To incorporate the most effective spam detection method.

c) To provide user friendly interface:

   Providing users easy to access, easy to operate and easy to understand interface.

d) To classify the message as spam or ham:

Using machine learning algorithms to classify the messages as spam or not spam (ham).

1.4 PROJECT SCOPE AND LIMITATIONS

The scope of the project is limited to the features mentioned above like group and instant messaging, profile personalization, emoji support, sending images and videos as media in messages and spam detection for text messages .

Some of the limitations of the project are-

1.Tracking user's last seen time is not implemented.

2.There is spam detection available for text messages but not for imges,video or other forms of messages.

3.There is no block feature to block certain users.

4.There is no functionality to delete sent messages or delete received messages for the receiver.

5.There is no feature to forward messages or share messages to multiple users at once.

1.5 METHODOLOGIES OF PROBLEM SOLVING

For classifiying a message as ham or spam, we need an algorithm.

- **Naive-Bayes Classifier**

  Simple classifier construction method. Assumes that, given a class variable, the value of one feature is unrelated to the value of any other feature.

- **Logistic Regression**

  For classification tasks, logistic regression employs the sigmoid and logit functions. S-shaped curves are used to anticipate the output variable.

- **K-nearest Neighbors**

Based on distance calculations, this classifier is easy to use and effective. The new data point is categorized based on the count of neighbors class and the k-nearest neighbors that are identified.

- **Decision Tree Classifiers**

  Uses a system of rules in the same way as humans do to make judgments. The idea is to build yes/no questions using the dataset's attributes, split the dataset repeatedly, and then isolate all the data points that correspond to each class.

- **Random Forest**

  To determine the class of a new data point, the Random Forest classifier consults with several different decision trees. It is an ensemble approach, which means that it creates a final output by combining the results of multiple trees.

- **SVM (Support Vector Machine)**

  A hyperplane is built in SVM based on the classification process. To locate the hyperplane, some data points are used as support vectors.

- **LSTM (Long Short Term Memory)**

  Artificial neural networks include recurrent neural networks. The previous state output in RNNs is used to create the current state input. But vanishing gradient descent is a concern for conventional RNNs. The gradients of the loss function in neural networks approach zero when more layers with specific activation functions are added, making the network challenging to train. LSTM addresses the issues with conventional RNNs. Text mining issues are best suited for LSTMs.

02.LITERATURE SURVEY

2.1 BACKGROUND OF DOMAIN

The increase in smart phone users has led to a huge increase in SMS spam messages. Despite the fact that a variety of information channels are currently seen as "spotless" and reliable in many parts of the world, ongoing data clearly demonstrate that there is rise in spam messages in phones oer time. It is a big issue all over the world.

Separating messages spam is a similarly late task to solve this problem. It gains several concerns and practical fixes from SMS spam separation. In any case, it brings up its own unique problems. By including Indian messages in the available meesages dataset, this research aims to address the challenge of classifying flexible messages as Ham or Spam for the Indian Users. The research examines various machine learning classifiers on a sizable dataset of individual SMS texts.

The entire research of Julis et al [1] was broken up into various iterations. Each iteration was finished by working through the following four stages: inception, where the project's idea was discovered; elaboration, where the system's architecture was designed; construction, where the existing code was put into use; and transition, where the developed portion of the project was validated. There are still certain areas that can be improved, such as by incorporating more filtering methods or altering certain features of the ones that already exist. Changes like increasing or decreasing the message's intriguing word count and rearranging the formula for determining interesting rate can be made afterwards.

This research however does not solve the problem of analysis and management of reports in spam sms filter storing, which is far more challenging.

Spam has developed into a significant problem for computer security since it serves as a primary channel for the spread of dangers including viruses, worms, and phishing scams. Presently, a significant portion of received emails and messages are spam. Different strategies are in place to address this issue, including challenge response models, whitelisting, blacklisting, email signatures, and various machine learning techniques.

These options are available to consumers, however owing to the dynamic nature of the Web, there aren't any worldwide 100% secure methods that can address this issue. Machine learning techniques are typically used by spam detectors to filter web traffic. Muhammed Iqbal et al's[3] research focuses on methodically examining the benefits and drawbacks of existing technology for spam detection, and a taxonomy of accepted methods is presented.

The research claims that there is no way to completely eradicate the spam problem and proposes a solution of merely combining various techniques of finding a probability score to determine whether the given text is spam or not. However, they don't say "How to combine those techniques".

Spam detection is crucial for protecting email and message communication. A significant problem is the accurate identification of spam, and numerous detection techniques have been put forth by various researchers. However, these techniques fall short in their ability to correctly and effectively detect spam. GuangJun et al [4] have suggested a technique for spam identification using machine learning predictive models to address this problem. The technique is used in order to identify spam. The testing findings demonstrate the great capability of the suggested strategy to detect spam. 99% accuracy was attained using the suggested strategy, which is high compared to other systems already in use. Thus, the findings imply that the suggested approach is more trustworthy for precise and prompt identification of spam and will secure messaging and email systems.

The spammers target those people who are unaware of these frauds and target them by easily creating phony profiles and email accounts. In their spam emails, they pose as a real person. Therefore, it is necessary to identify spam emails that are fraudulent. Nikhil Kumar et al's [5] project will do this by using machine learning techniques. Their paper will discuss machine learning algorithms and show how to apply them to their data sets. The best algorithm for email spam detection is then chosen because it has the highest precision and accuracy.

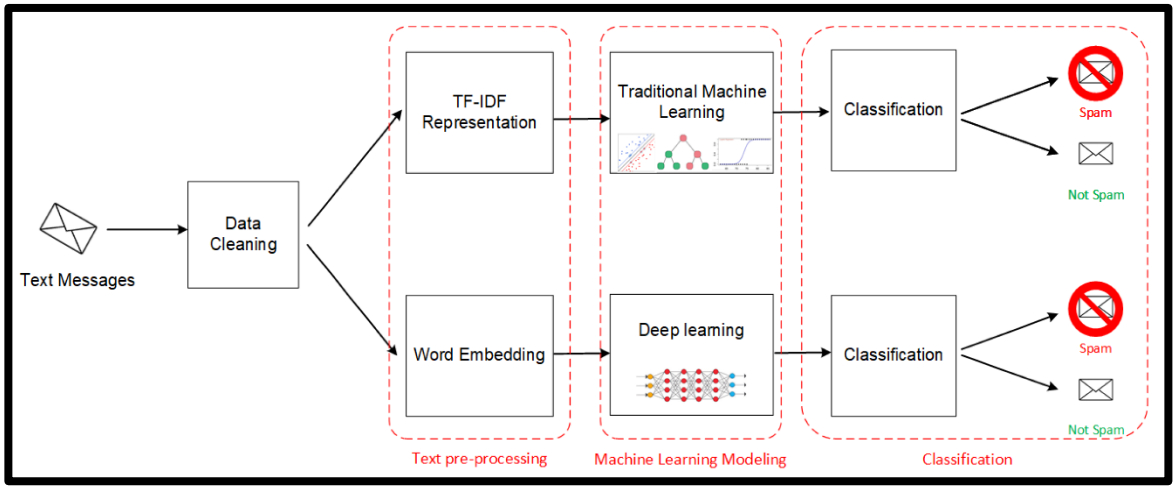


Fig 2.1: Process to Classify the Messages

2.2 COMPARISON OF RESEARCH PAPERS STUDIED

Sr. No	Research Paper	Description	Merits	Demerits
1	Sridevi Gadde,A .Lakshmanarao,S.S atyanarayana, "SMS Spam Detection using Machine Learning and Deep Learning Techniques", 2021	In this paper, authors proposed a deep learning model for SMS spam detections. The accuracy is compared with the proposed model with previous work.	Author applied various classification algorithms and achieved an accuracy of 99% with the LSTM model. Experimental results showed that LSTM outperforms previous models for spam detection.	The dataset which is being used is UCI with labels, So they applied various supervised learning algorithms for SMS spam detection. The model is tested on only this dataset
2	M.Rubin Julis, S.Alagesan, "Spam Detection In Sms Using Machine	The research examines various machine learning classifiers on a	The research work involved a careful study on the different filtering algorithms and existing anti-spam tools.	This research however does not solve the problem of analysis and management of report in spam sms

	Learning Through Text Mining", 2020	sizable dataset of individual SMS texts		filter storing, which is far more challenging.
3	Muhammad Iqbal, Malik Muneeb Abid, Mushtaq Ahmad, Faisal Khurshid, "Study on the Effectiveness of Spam Detection Technologies", 2016	This paper states different strategies are in place to address spam detection issues, including challenge response models, whitelisting, blacklisting, pattern detection, and various machine learning techniques.	Research focuses on methodically examining the benefits and drawbacks of existing technology for spam detection, and a taxonomy of accepted method is presented.	They proposed a solution of merely combining various techniques of finding a probability score to determine whether the given text is spam or not. However, they don't say "How to combine those techniques".
4	Luo GuangJun, Shah Nazir, Habib Ullah Khan, Amin Ul Haq, "Spam Detection Approach for Secure Mobile Message Communication Using Machine Learning Algorithms", 2020	GuangJun et have suggested a technique for spam identification using machine learning predictive models to address this problem. The technique is used in order to identify spam	99% accuracy was attained using the suggested strategy, which is high compared to other systems already in use	The model is tested on only one dataset.

03.REQUIREMENTS

3.1 DESCRIPTION OF REQUIREMENT

Developing a real time chat-application with spam detection capabilities using machine learning methods. The chat application has a spam detection feature driven by AI. The messages can be filtered out as spam or ham using the machine learning technique.

3.2 SRS(System Requirements Specification)

3.2.1 Scope

The scope of this project includes group chat, emoji support, push notifications, AI powered spam detection, profile customization and slash commands. Using the chat application, the user would also be able to react to messages, preview the URL's and videos, and monitor presence and status indications while utilizing the chat application.

The scope of this project does not include features like video calling, voice calling, audio recording, or disappearing messaging.

3.2.2 Features

- Register user
- Login for web app
- Group chat
- Emoji support
- Push notifications for messages
- AI powered spam detection
- User settings and profile customization
- User role configuration
- Typing indicators
- Message status indicators

- Message reactions
- URL Preview
- File upload and preview
- Video playback
- Auto complete enabled search
- Slash commands

3.2.3 Functional Requirements

Requirement 1: User registration

Function	User Registration
Pre-condition	User should not be registered already
Steps	User fills all the relevant details
Post-condition	The user successfully registers for the application

Requirement 2: Convey a Message

Function	Users should be able to send and receive messages instantly.
Pre-condition	User should be already registered
Steps	1) User access the application 2) To send a message, the user clicks on the contact. 3) After entering the message, the user clicks the Send button.
Post-condition	The message is successfully sent.

Requirement 3: User's message should be sent to Backend Server.

Function	User's message should be delivered to the backend to check for spam/ham.
Pre-condition	User should have already sent a message.
Steps	User types in the message and hits the "send" button.

	User's message is then sent to the backend server via the frontend server.
Post-condition	The user's message is received at the backend server.

Requirement 4: Backend server must send the user message for Spam detection.

Function	Backend sends the user message for spam detection.
Pre-condition	User's message should be delivered at the backend server.
Steps	User's message is passed as an argument to the function that uses the trained model to make predictions on spam detection.
Post-condition	The user's message has been classified as spam/ham.

Requirement 5: Backend Server sends the prediction back to the user

Function	Backend provides spam/ham response back to the frontend once prediction is complete.
Pre-condition	Spam Detection Model has already made a prediction on the received user messages.
Steps	Prediction from trained model is sent to the backend server. Backend server gives a response to frontend server with spam/ham classification on the current user message. Frontend Server works appropriately depending on the spam/ham classification of the user message
Post-condition	The user gets to know if his message was spam or not.

3.2.4 External Interface Requirements

3.2.4.1 User Interfaces

- Front-End Software: React-Native for Android/iOS and ReactJS for WebApp
- Back-End Software: PostgreSQL

- Framework: Django

3.2.4.2 Hardware Interfaces

- Android/ iOS Mobile Phone
- 128 Minimum Ram Required
- Internet/LAN
- Laptop/PC with any operating system

3.2.4.3 Software Interfaces

- Operating System:
  For Mobile Application: Android, iOS
  For Web Application: Windows/MacOS
- Browser: Browser that supports Javascript, CGI and HTML.

3.2.5 Non-Functional Requirements

- **Performance:**

The following performance requirements are being taken into account:

a) Instant message sending

b) Simultaneous access

c) Instant application loading

d) Registration procedure time of less than one minute

- **Security:**

The user will have access to his personal profile. To protect privacy, the user will be able to manage the ways that people can read his profile and his status indications.

- **Software quality attributes:**
  - **Availability:** The application needs to be accessible to users around-the-clock.

- **Portability:** The application can be accessed through any web application and mobile application (Android and iOS).
- **Scalability:** The application should be able to provide service to over 1000 users simultaneously.
- **Usability:**The application should be user-friendly, easy to operate and should provide a pleasant experience for the users.

3.2.6 System Requirements

- **Database Requirement**

PostgreSQL-With over 35 years of continuous development, the durable, open-source PostgreSQL object-relational database system has built a solid reputation for dependability, feature robustness, and performance.

The official documentation has a lot of information outlining how to set up and use PostgreSQL. There are several of helpful resources in the open source community to learn about PostgreSQL, understand how it functions, and locate job possibilities. Learn more about ways to interact with the neighbourhood.

- **Software Requirements**

Sr. No	Tools/Technology	Use
1	Python	Backend Development
2	Python Libraries (Scikit-learn, Pandas, etc.)	Spam detection Model
3	ReactJS	Frontend Development(Web)
4	ReactNative	Frontend Development(mobile)
5	Selenium	Web testing
6	Postman	API Testing

- **Hardware Requirements**

1. Android/ iOS Mobile Phone

2. 128 Minimum Ram Required
3. Internet/LAN
4. Laptop/PC with any operating system

3.3 Analysis Model: SDLC Model to be applied-

The **Agile methodology** is used for implementation of this project.It is a way to manage a project by breaking it up into several phases and involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams cycle through a process of planning, executing, and evaluating. Continuous collaboration is vital, both with team members and project stakeholders. In Agile processes, there is constant feedback, allowing team members to adjust to challenges as they arise and it also gives the stakeholders an opportunity to oversee the development of the product and offer continuous feedback. This approach is useful as the requirements change with time and the development is done under the stakeholder's supervision.Unlike traditional approaches for development, this leaves very little room for major errors in the project and is of great benefit to all parties involved.

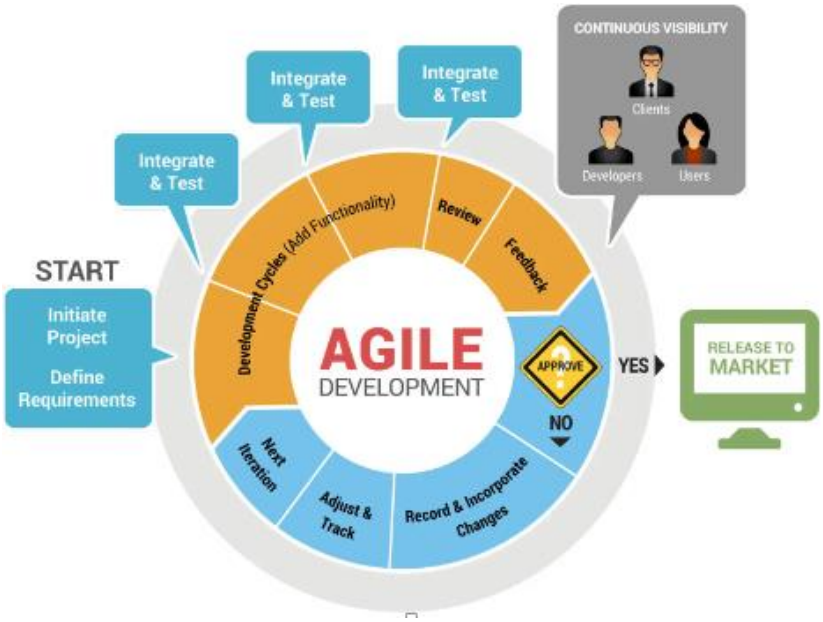


Fig 3.3: Agile Development Methodology

04.SYSTEM DESIGN

4.1 System Architecture

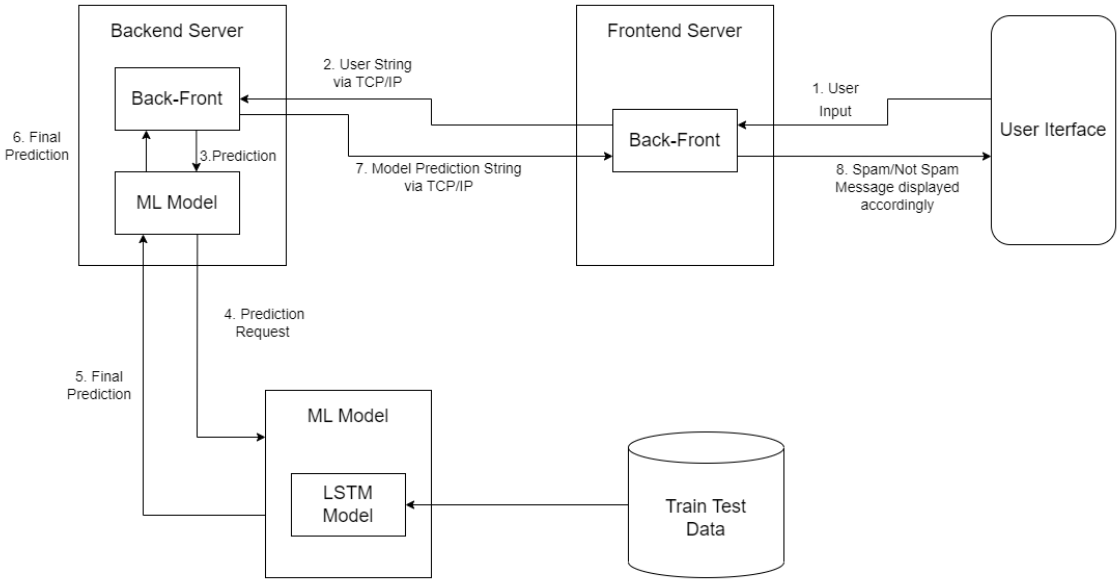


Fig 4.1 :System Architecture

4.2 Algorithms

Create a Backend Server:

1. Host the SMS Spam-Detection trained model on the server.
2. Open a port on which the frontend of the chatbot can communicate on.
3. Accept string/user messages from the frontend and give it as an input to the trained SMS Spam-Detection Model.
4. The model will return its predictions to the backend server, which will then send the predictions back to the frontend.

Create a Frontend Server:

1. Connect the frontend server to the frontend of the Chatbot (Mobile App/Web App)
2. For each client, create a thread to keep each client's request isolated from the other clients.
3. Accept user messages when the user hits the "send" button.

4. Send the messages to the backend server via the communicable port established at the backend server and wait for a response.
5. If the response is “spam”, exit.
6. If the response is “ham”, send an appropriate response to the user.

4.3 Entity Relationship Diagram

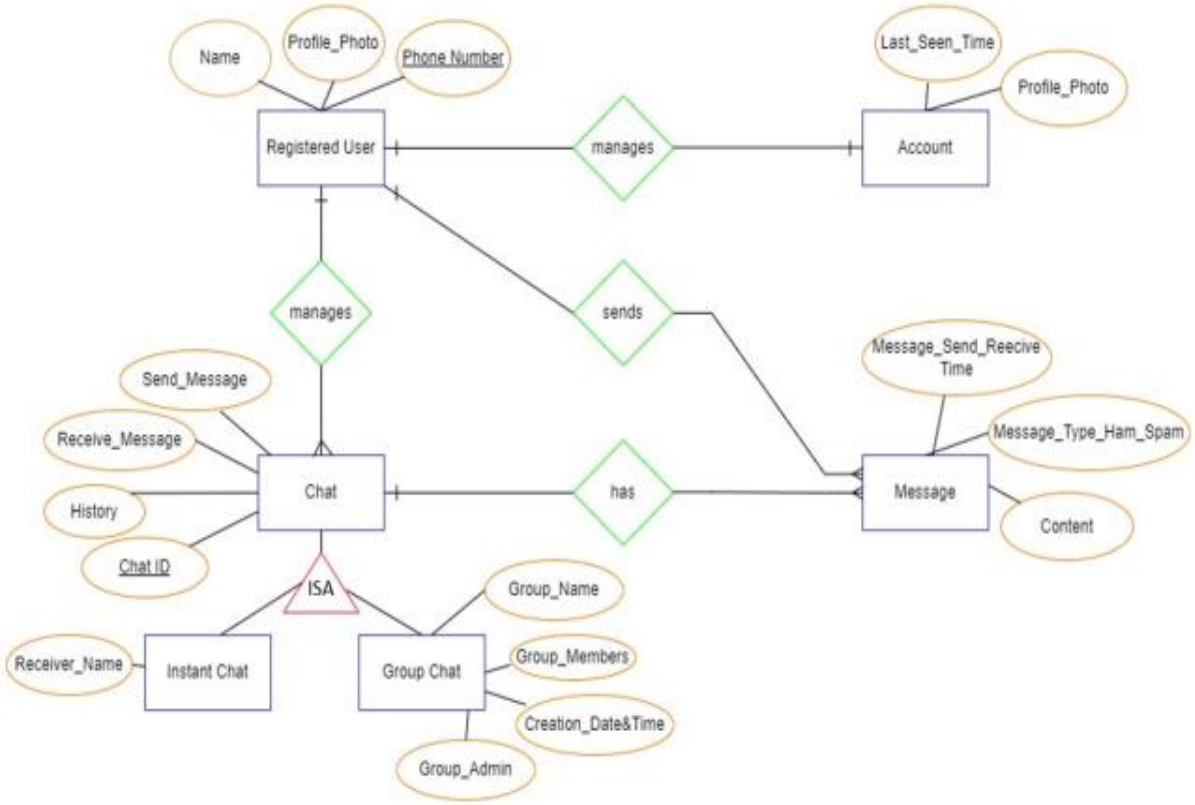


Fig 4.2: ER Diagram

4.4 UML Diagrams

- **Use Case Diagram**

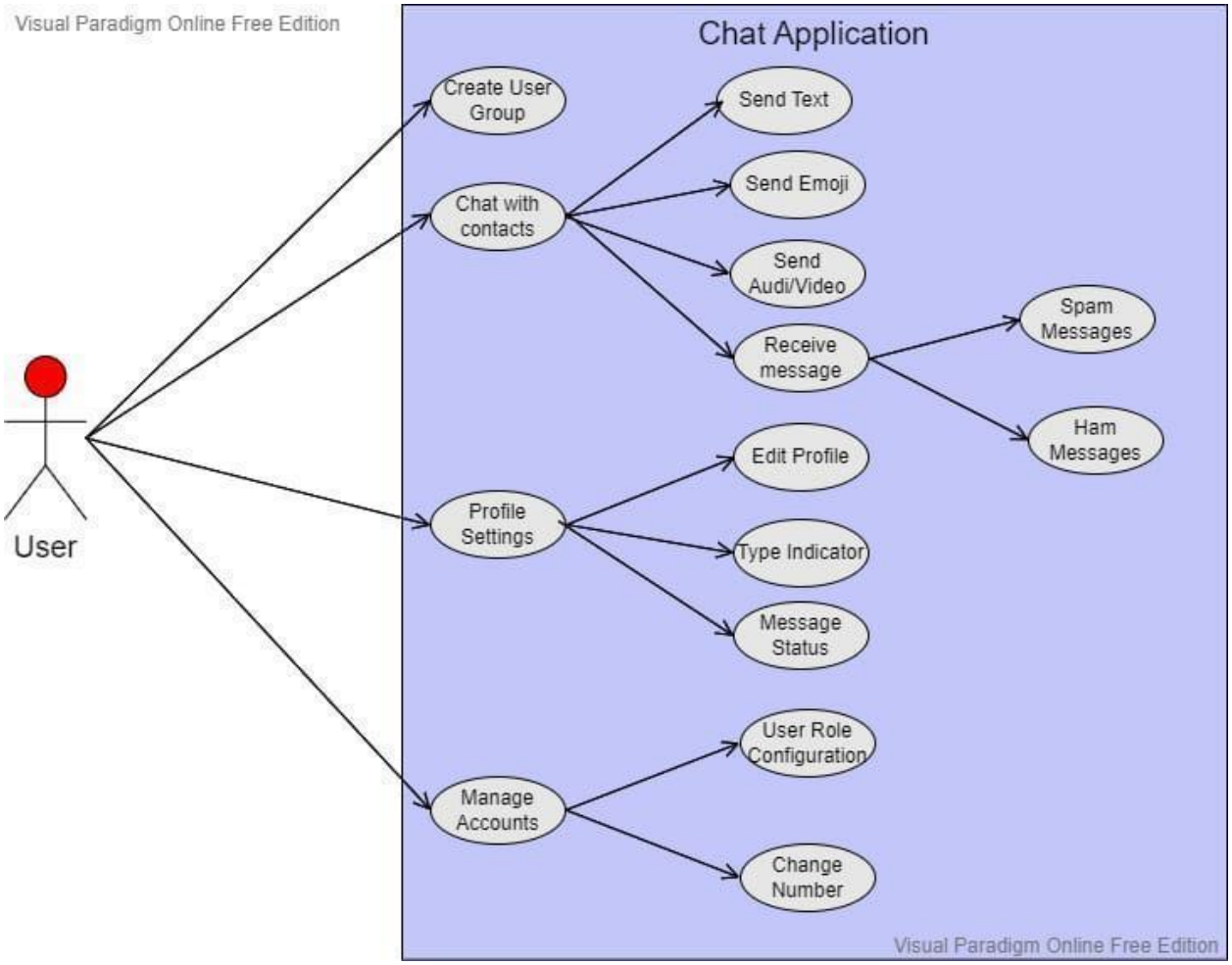


Fig 4.3: Use Case Diagram

- **Activity Diagram**

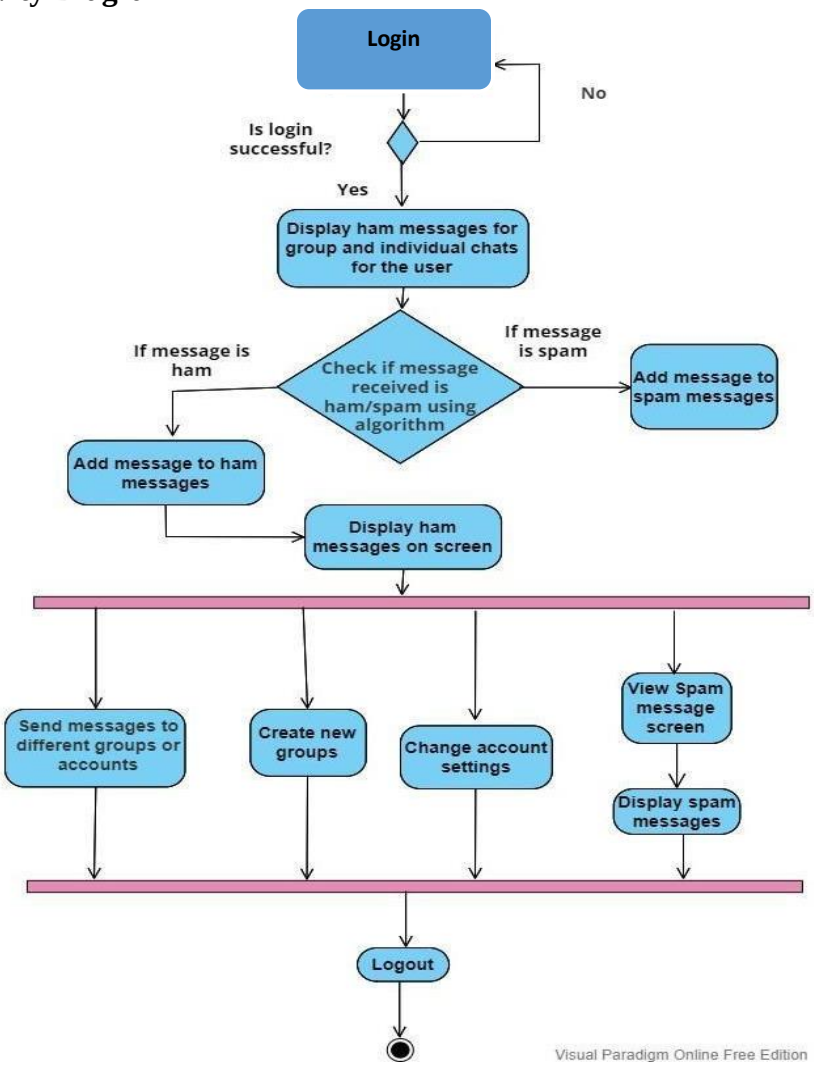


Fig 4.4: Activity Diagram

- **Class Diagram**

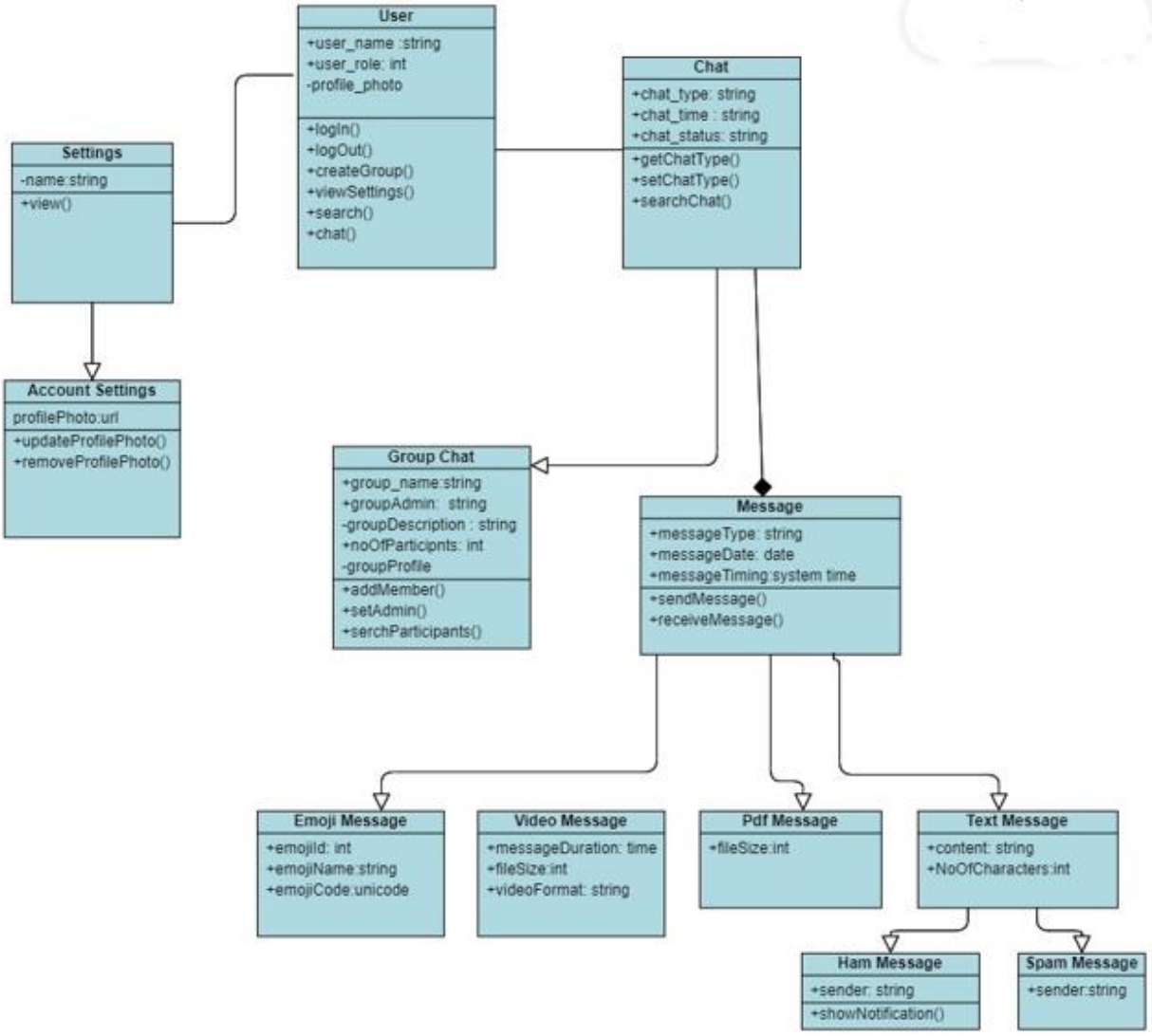


Fig 4.5 Class Diagram

4.5 Data Flow Diagrams

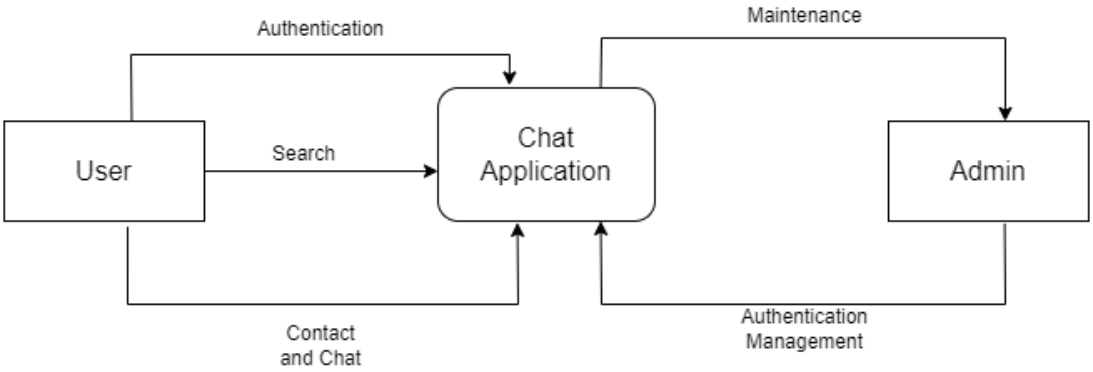


Fig 4.8: Level 0 DFD

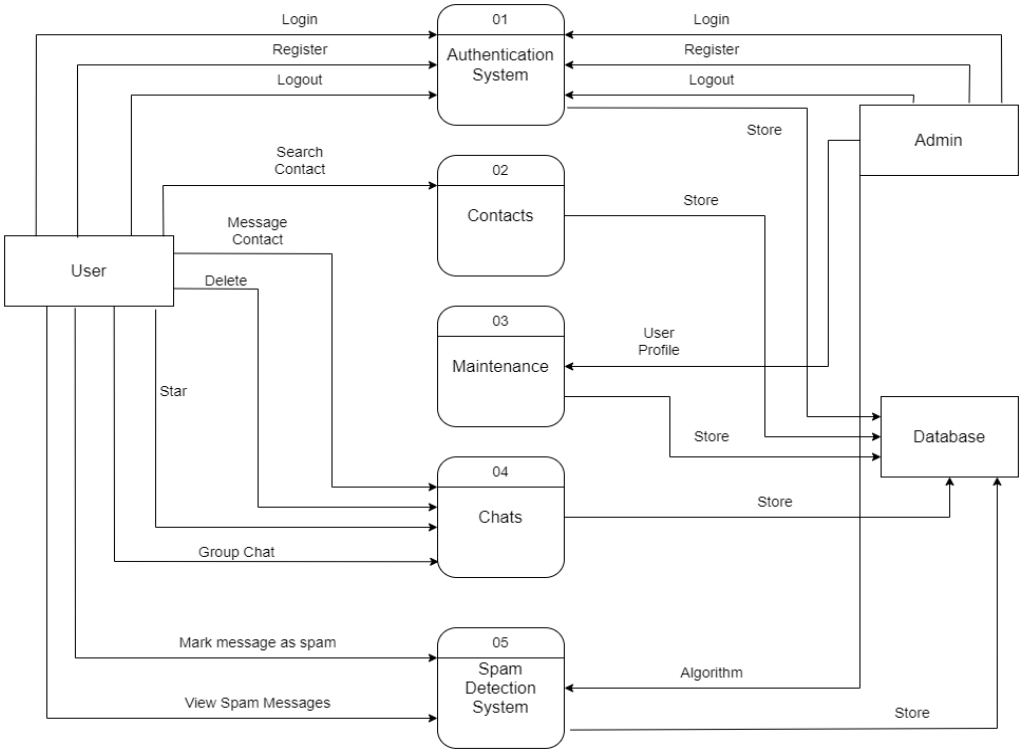


Fig 4.9: Level 1 DFD

05. TECHNOLOGY

5.1 TECHNOLOGY STACK

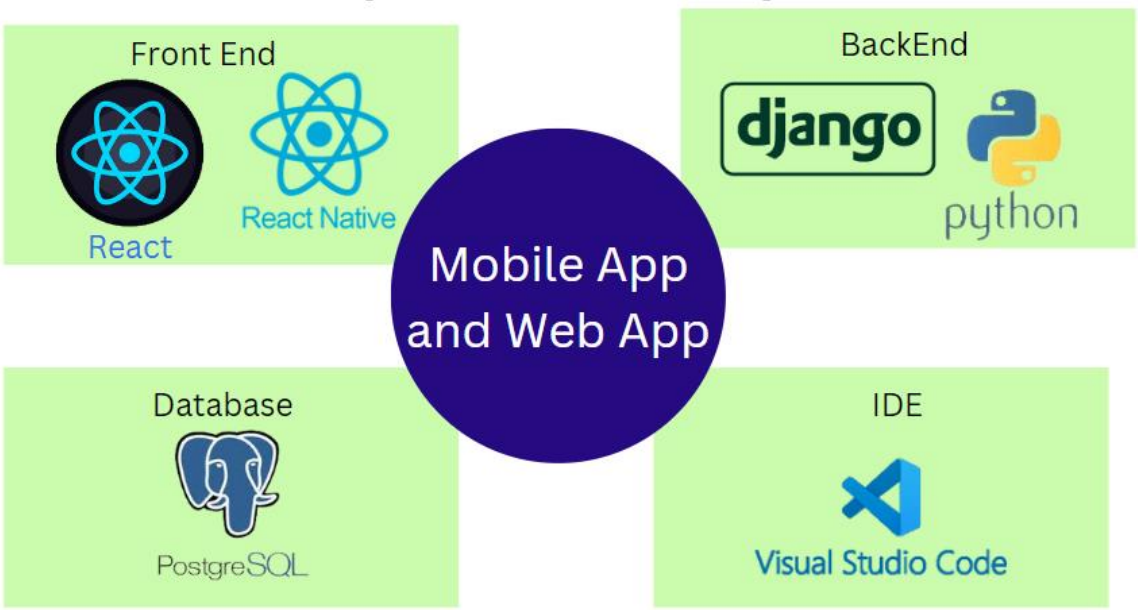


Fig 5.1:Technology Stack

React:

React (React.js/ReactJS) is a JavaScript frontend library which is free and open-source.It is used for creating user interfaces based on UI components. It is managed by Meta and a some other independent programmers and groups.React JS is useful forr single-page, mobile, or server-rendered applications. Beyond only UI, React has certain extensions for supporting Flux and React Native.

React Native

You may create natively rendered mobile apps for iOS and Android using the mobile app framework known as React Native (RN).It enables us to develop applications for numerous platforms which includes the most popular mobile applications in the world-Instagram, Facebook, and Skypewhich are powered by React Native development. Significant time and resource savings result from this. React, a JavaScript library that was already quite well-liked when the mobile framework was announced, served as the

foundation for React Native. The framework allowed frontend developers, who had previously been restricted to working with web-based technologies, to produce comprehensive programmes for mobile platforms.

Postgresql

An open-source relational database system is PostgreSQL.Postgresql supports both SQL (relational) and JSON (non-relational) querying. A highly stable database, PostgreSQL has been developed by the open-source community for more than 20 years. Numerous web, mobile, and analytics applications all use PostgreSQL as their main database.

Django

A Python framework called Django makes it simpler to develop websites using Python. It handles the challenging aspects so you can focus on creating your web applications. Don't Repeat Yourself (DRY) principles are emphasized, and it includes login system, database connection, and CRUD operations.

Python

Python is a popular high-level programming language invented by Guido van Rossum. Python requires less code due to its syntax and was designed with code readability in mind. Python is enables quick work and more effective system integration.

Standard tools, APIs and Libraries used:

1. Pandas
2. Numpy
3. NLTK
4. Tensorflow
5. Keras

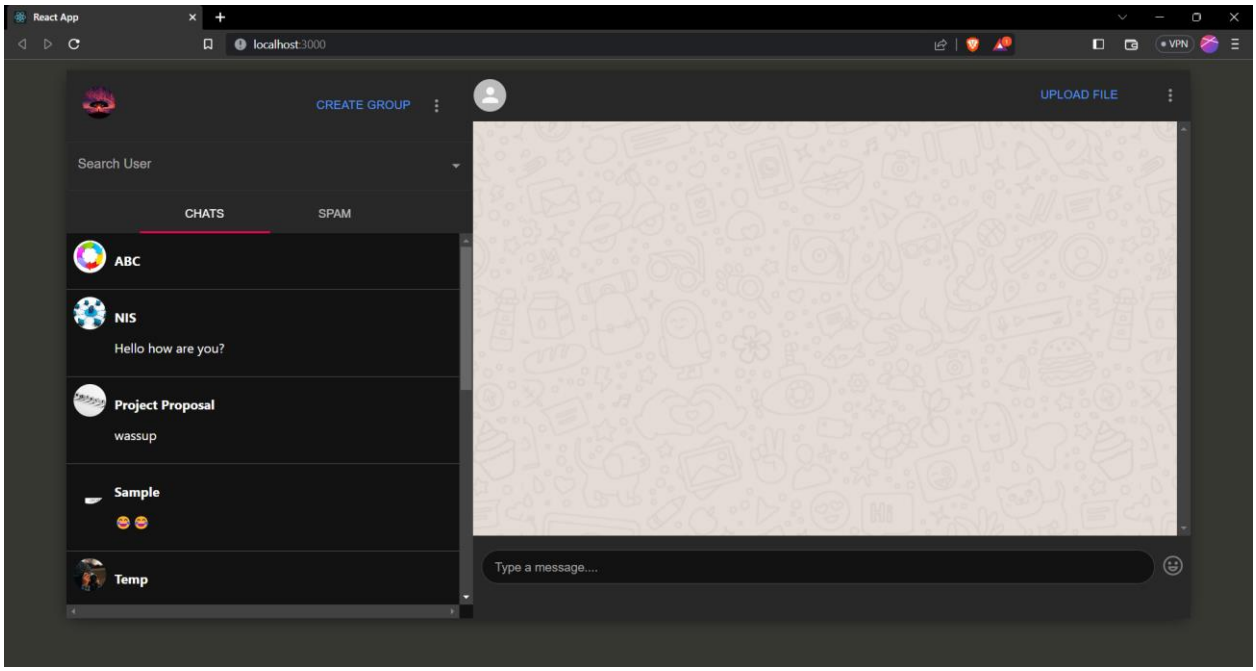
5.2 TEST PLANS

Test Scenario Description	Test Case ID	Test Case Description	Testing Steps	Pre-requisites	Test Data	Expected Result	Actual Result	Status
Verify Login Functionality in WebApp	TC_01	Login with valid email and valid password	1. Launch the Web app 2. Enter valid email 3. Enter valid password 4. Click on Login	User must have a network connection	email: finalprojectgr11@gmail.com password:xxxxxxxxxx	Successful Login	Successful Login	Pass
	TC_02	Login with valid email and invalid password	1. Launch the Web app 2. Enter valid email 3. Enter valid password 4. Click on Login	User must have a network connection	email: finalprojectgr11@gmail.com password:123456	Pop-up message stating "Invalid Username or Password"	Pop-up message stating "Invalid Username or Password"	Pass
	TC_03	Login with invalid email and valid password	1. Launch the Web app 2. Enter valid email 3. Enter valid password 4. Click on Login	User must have a network connection	email: finalprojectgr@gmail.com password:xxxxxxxxxx	Pop-up message stating "Invalid Username or Password"	Pop-up message stating "Invalid Username or Password"	Pass
	TC_04	Login with invalid email and invalid password	1. Launch the Web app 2. Enter valid email 3. Enter valid password 4. Click on Login	User must have a network connection	email: finalprojectgr@gmail.com password:123456	Pop-up message stating "Invalid Username or Password"	Pop-up message stating "Invalid Username or Password"	Pass
Verify Login Functionality in MobileApp	TC_05	Login with valid email and valid password	1. Launch the Mobile app 2. Enter valid email 3. Enter valid password 4. Click on Login	User must have a network connection	email: finalprojectgr11@gmail.com password:xxxxxxxxxx	Successful Login	Successful Login	Pass
	TC_06	Login with valid email and invalid password	1. Launch the Mobile app 2. Enter valid email 3. Enter valid password 4. Click on Login	User must have a network connection	email: finalprojectgr11@gmail.com password:123456	Pop-up message stating "Invalid Username or Password"	Pop-up message stating "Invalid Username or Password"	Pass
	TC_07	Login with invalid email and valid password	1. Launch the Mobile app 2. Enter valid email 3. Enter valid password 4. Click on Login	User must have a network connection	email: finalprojectgr@gmail.com password:xxxxxxxxxx	Pop-up message stating "Invalid Username or Password"	Pop-up message stating "Invalid Username or Password"	Pass
	TC_08	Login with invalid email and invalid password	1. Launch the Mobile app 2. Enter valid email 3. Enter valid password 4. Click on Login	User must have a network connection	email: finalprojectgr@gmail.com password:123456	Pop-up message stating "Invalid Username or Password"	Pop-up message stating "Invalid Username or Password"	Pass

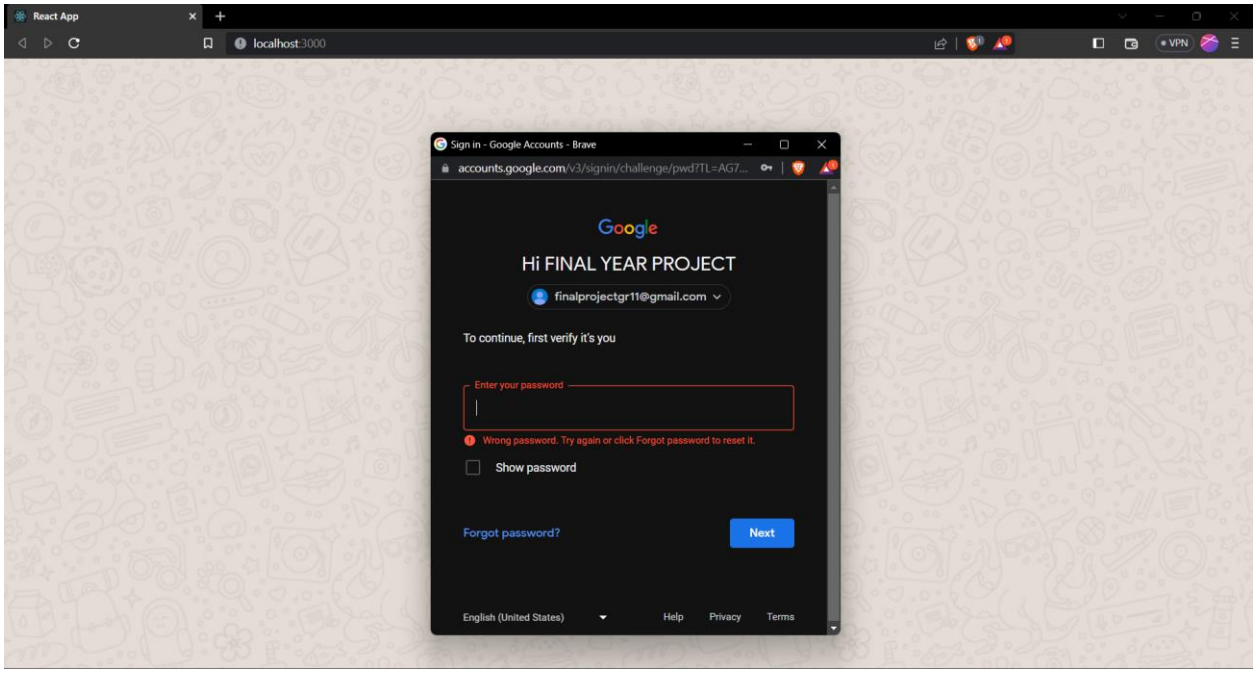
Create New Group in WebApp	TC_09	Create new group for users	1. Launch the Web App 2. Login Successfully 3. Open menu and click on "Create new group" icon 4. Give a group name 5. Select a group picture 6. Select users by searching via email id	User must have a network connection and chat database must have at least 3 registered users	Group name "WebApp" with any picture and 2 selected users	Group Created Successfully	Group Created Successfully	Pass
Create New Group in MobileApp	TC_10	Create new group for users	1. Launch the Mobile App 2. Login Successfully 3. Open menu and click on "Create new group" icon 4. Give a group name 5. Select a group picture 6. Select users by searching via email id	User must have a network connection and chat database must have at least 3 registered users	Group name "MobileApp" with any picture and 2 selected users	Group Created Successfully	Group Created Successfully	Pass
Create Instant Chat with another user in WebApp	TC_11	Start an instant chat with another user	1. Launch the WebApp 2. Login Successfully 3. Type the user's email id in Search Bar 4. Select the user if found	User must have a network connection and chat database must have at least 2 registered users	user: boilearly@gmail.com	Instant Chat created Successfully	Instant Chat created Successfully	Pass
Create Instant Chat with another user in MobileApp	TC_12	Start an instant chat with another user	1. Launch the Mobile App 2. Login Successfully 3. Type the user's email id in Search Bar 4. Select the user if found	User must have a network connection and chat database must have at least 2 registered users	user: boilearly@gmail.com	Instant Chat created Successfully	Instant Chat created Successfully	Pass
Test the classification of spam-ham messages in WebApp	TC_13	Check if ham messages are being displayed in group chat	1. Launch the Web App 2. Login Successfully 3. Select any group 4. Type in a ham message 5. Hit send	User must have a network connection and must have at least 1 group	message: Good morning, How are you doing?	Message is displayed	Message is displayed	Pass
	TC_14	Check if spam messages are being displayed in spam bar	1. Launch the Web App 2. Login Successfully 3. Select any group 4. Type in a spam message 5. Hit send	User must have a network connection and must have at least 1 group	messages: Good morning!	Message is displayed in spam bar	Message is displayed in spam bar	Pass
	TC_15	Check if ham messages are being displayed in Instant chat	1. Launch the Web App 2. Login Successfully 3. Select any instant chat 4. Type in a ham message 5. Hit send	User must have a network connection and must have at least 1 instant chat	message: Good morning, How are you doing?	Message is displayed	Message is displayed	Pass
	TC_16	Check if spam messages are being displayed in spam bar	1. Launch the Mobile App 2. Login Successfully 3. Select any instant chat 4. Type in a spam message 5. Hit send	User must have a network connection and must have at least 1 instant chat	messages: Good morning!	Message is displayed in spam bar	Message is displayed in spam bar	Pass

Test the classification of spam-ham messages in MobileApp	TC_17	Check if ham messages are being displayed in group chat	1. Launch the Mobile App 2. Login Successfully 3. Select any group 4. Type in a ham message 5. Hit send	User must have a network connection and must have at least 1 group	message: Good morning, How are you doing?	Message is displayed	Message is displayed	Pass
	TC_18	Check if spam messages are being displayed in spam bar	1. Launch the Mobile App 2. Login Successfully 3. Select any group 4. Type in a spam message 5. Hit send	User must have a network connection and must have at least 1 group	messages: Good morning!	Message is displayed in spam bar	Message is displayed in spam bar	Pass
	TC_19	Check if ham messages are being displayed in Instant chat	1. Launch the Mobile App 2. Login Successfully 3. Select any instant chat 4. Type in a ham message 5. Hit send	User must have a network connection and must have at least 1 instant chat	message: Good morning, How are you doing?	Message is displayed	Message is displayed	Pass
	TC_20	Check if spam messages are being displayed in spam bar	1. Launch the Mobil App 2. Login Successfully 3. Select any instant chat 4. Type in a spam message 5. Hit send	User must have a network connection and must have at least 1 instant chat	messages: Good morning!	Message is displayed in spam bar	Message is displayed in spam bar	Pass

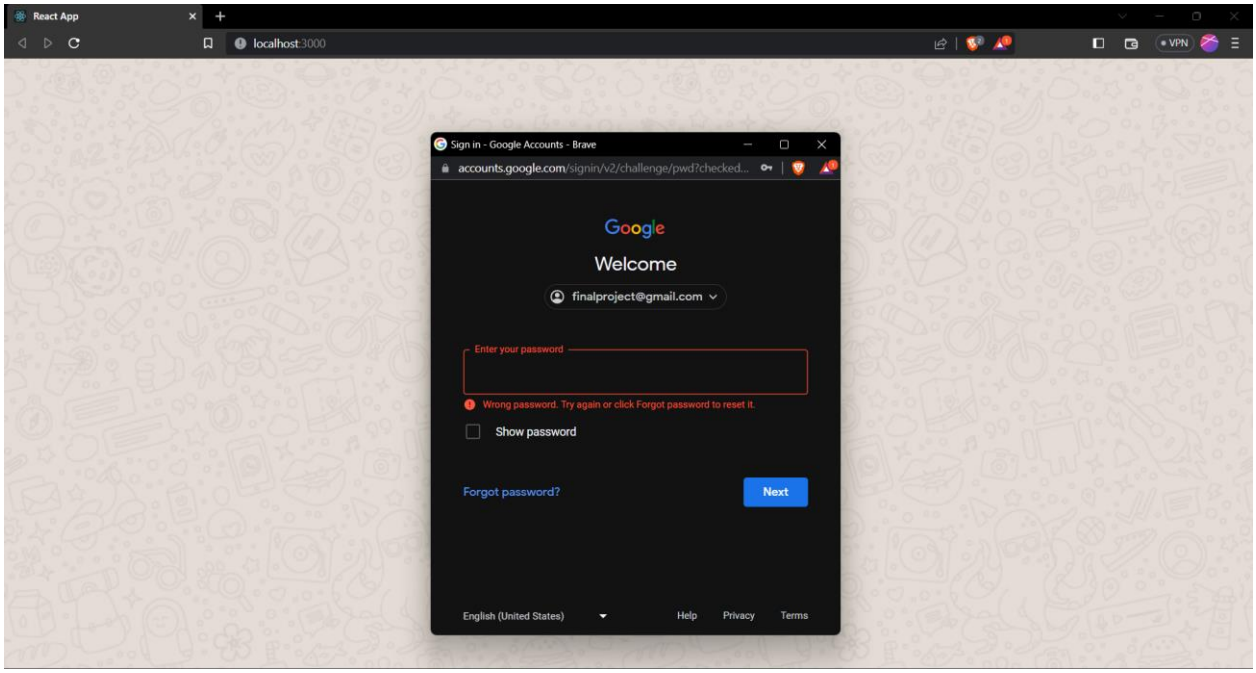
5.2.1 TC_01 - Login with valid email and valid password



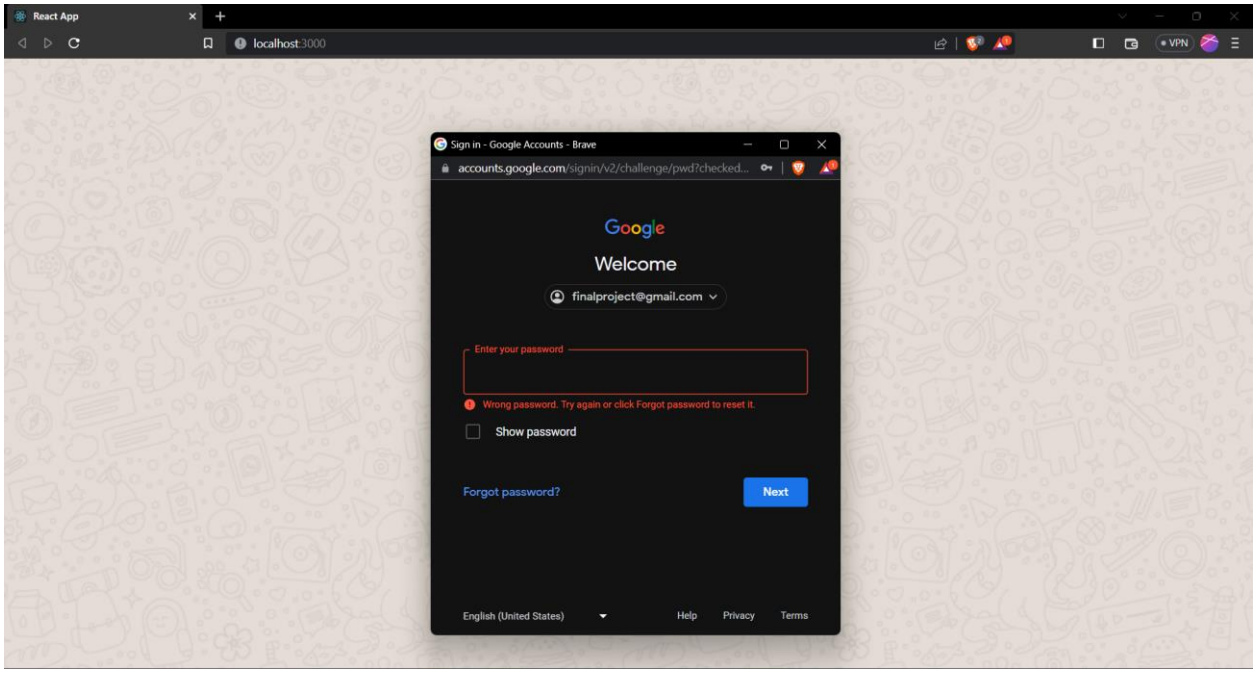
5.2.2 TC_02 - Login with valid email and invalid password



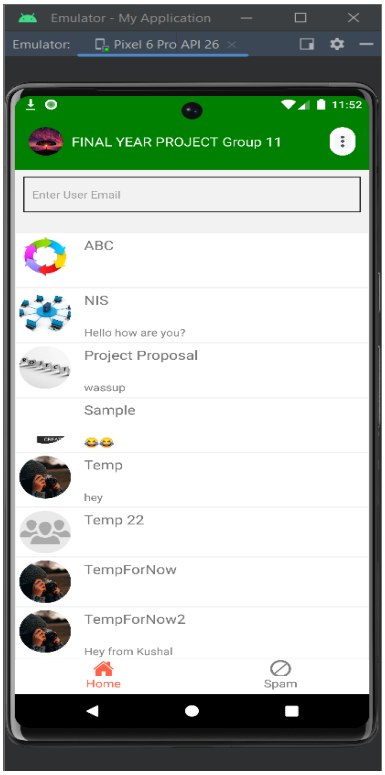
5.2.3 TC_03 - Login with invalid email and valid password



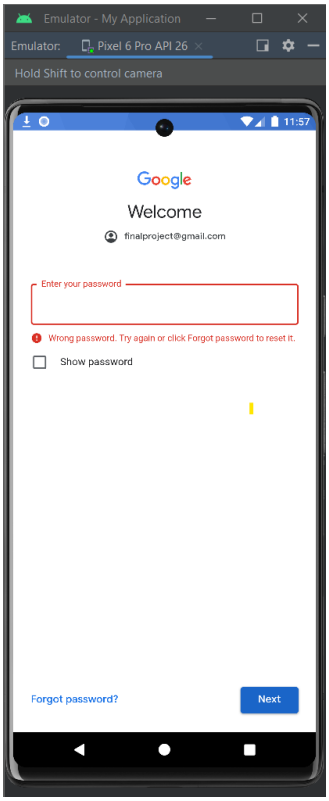
5.2.4 TC_04 - Login with invalid email and invalid password



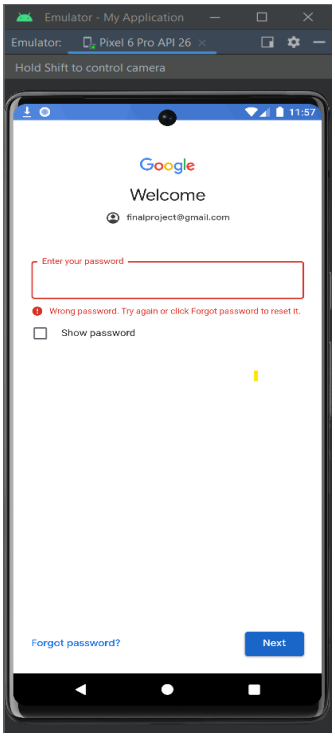
5.2.5 TC_05 - Login with valid email and valid password



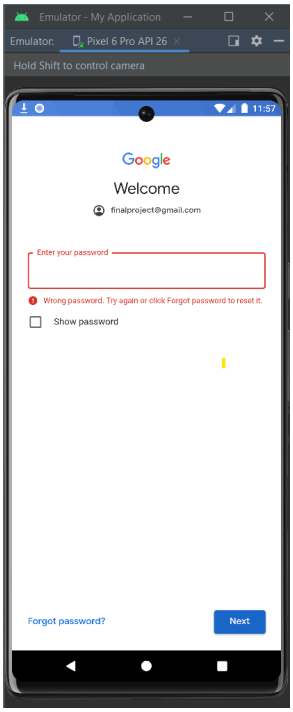
5.2.6 TC_06 - Login with valid email and invalid password



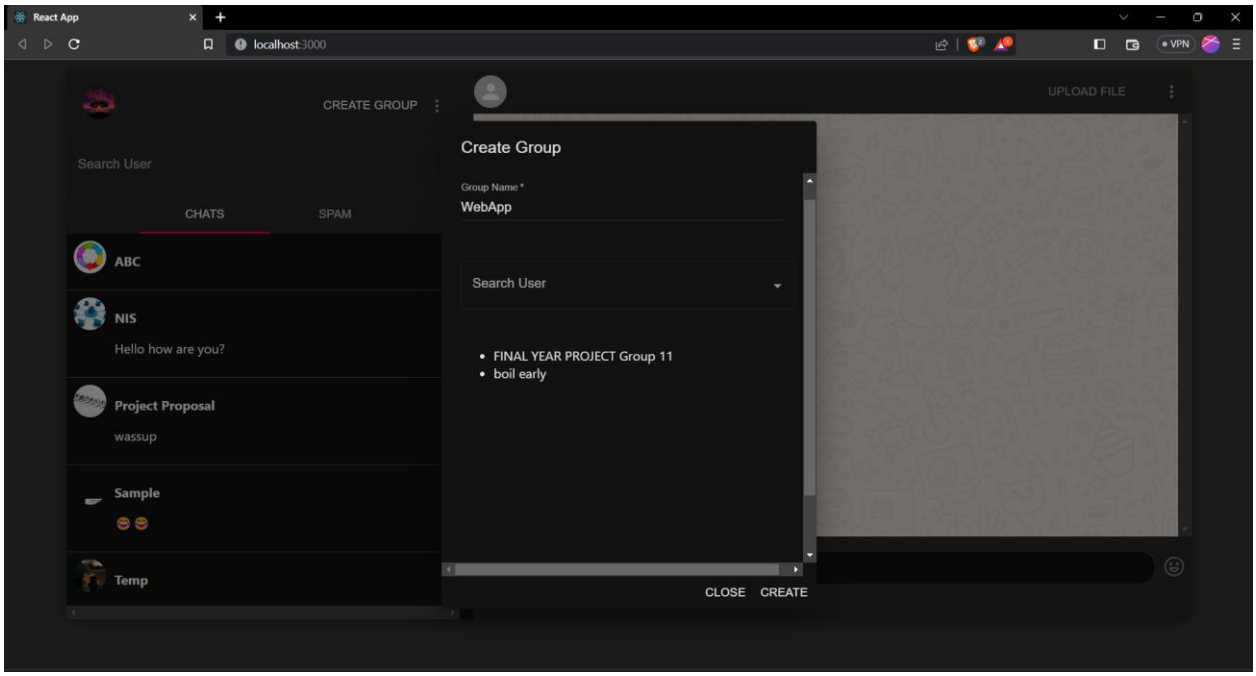
5.2.7 TC_07 - Login with invalid email and valid password

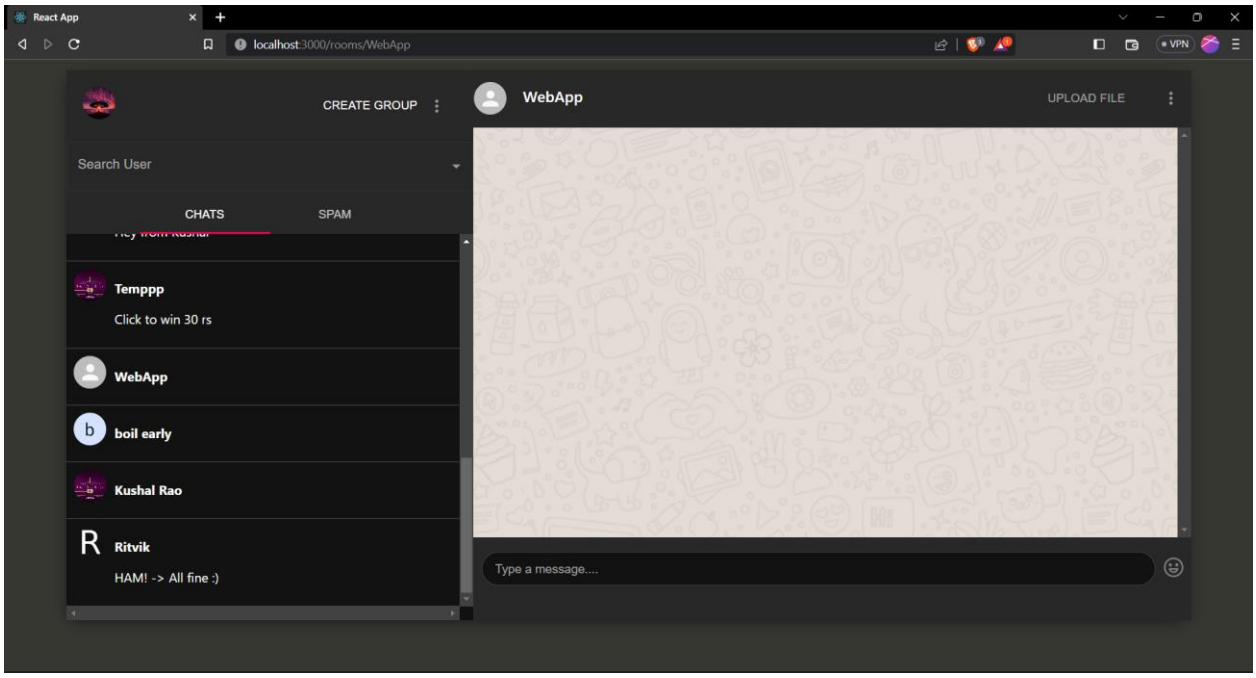


5.2.8 TC_08 - Login with invalid email and invalid password

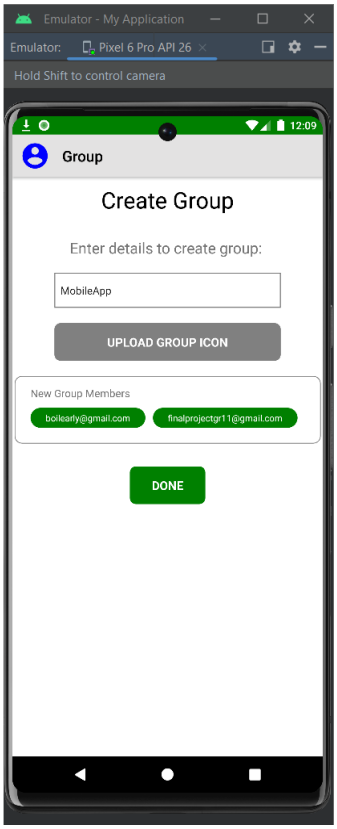


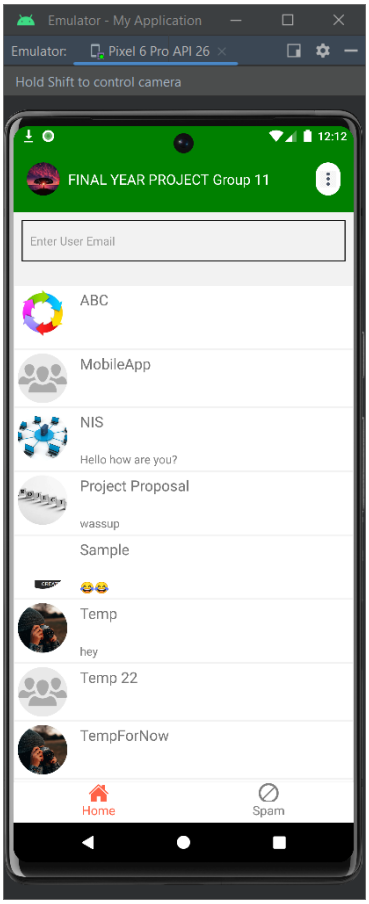
5.2.9 TC_09 - Create New Group in WebApp



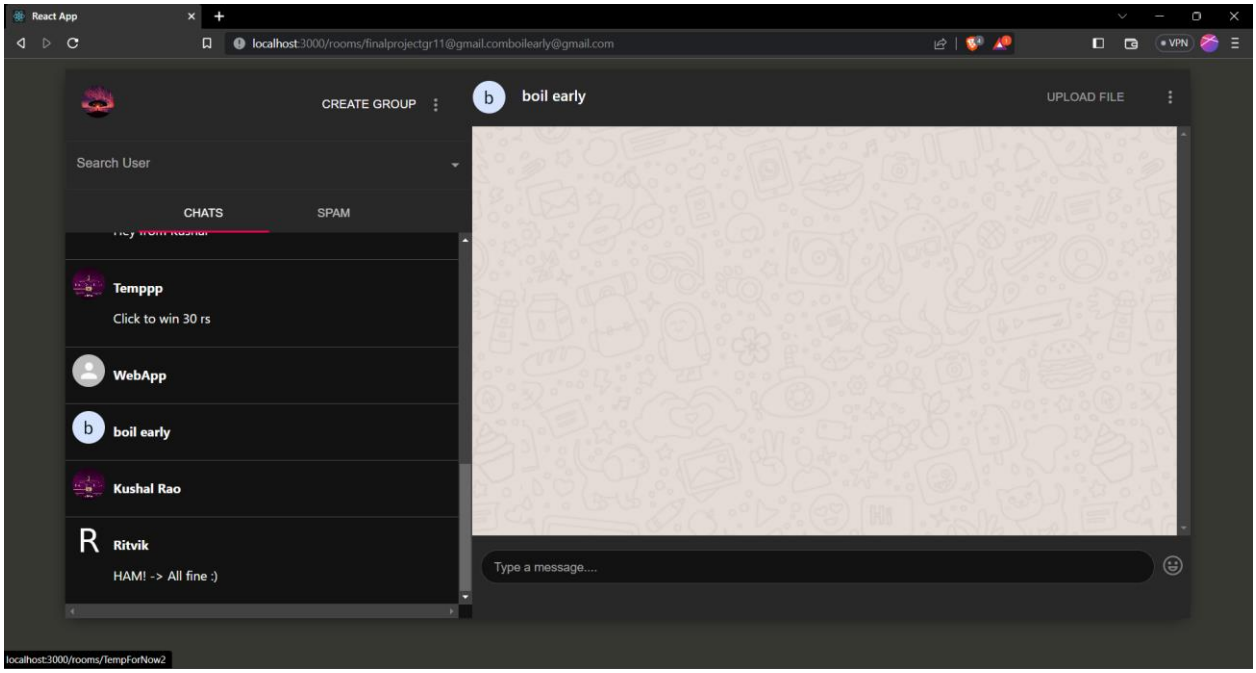


5.2.10 TC_10 - Create New Group in MobileApp



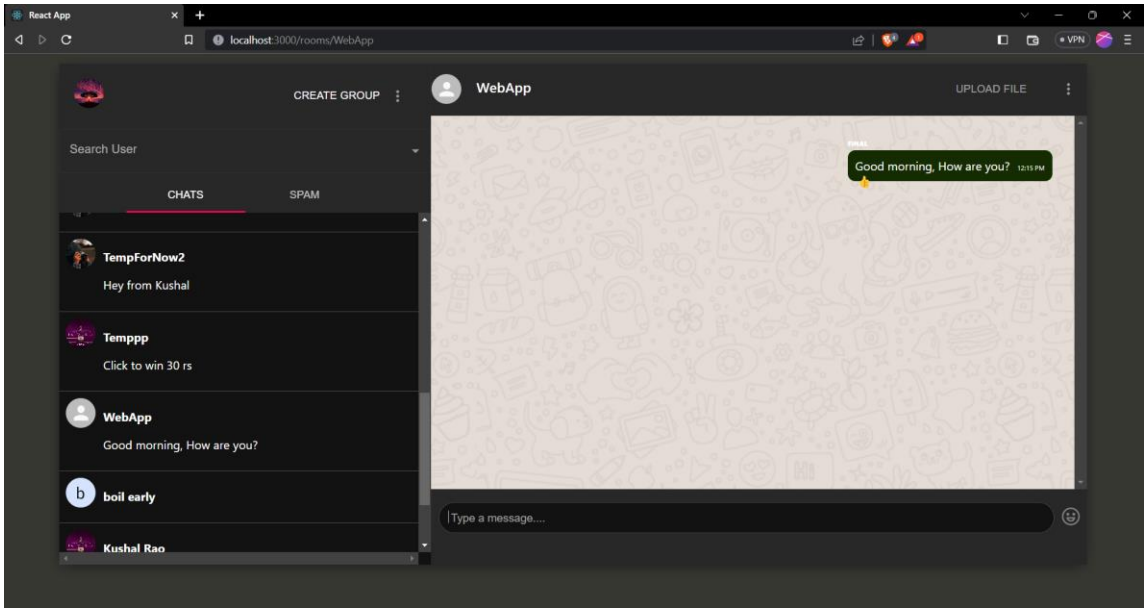


5.2.11 TC_11 - Start an Instant Chat on Webapp

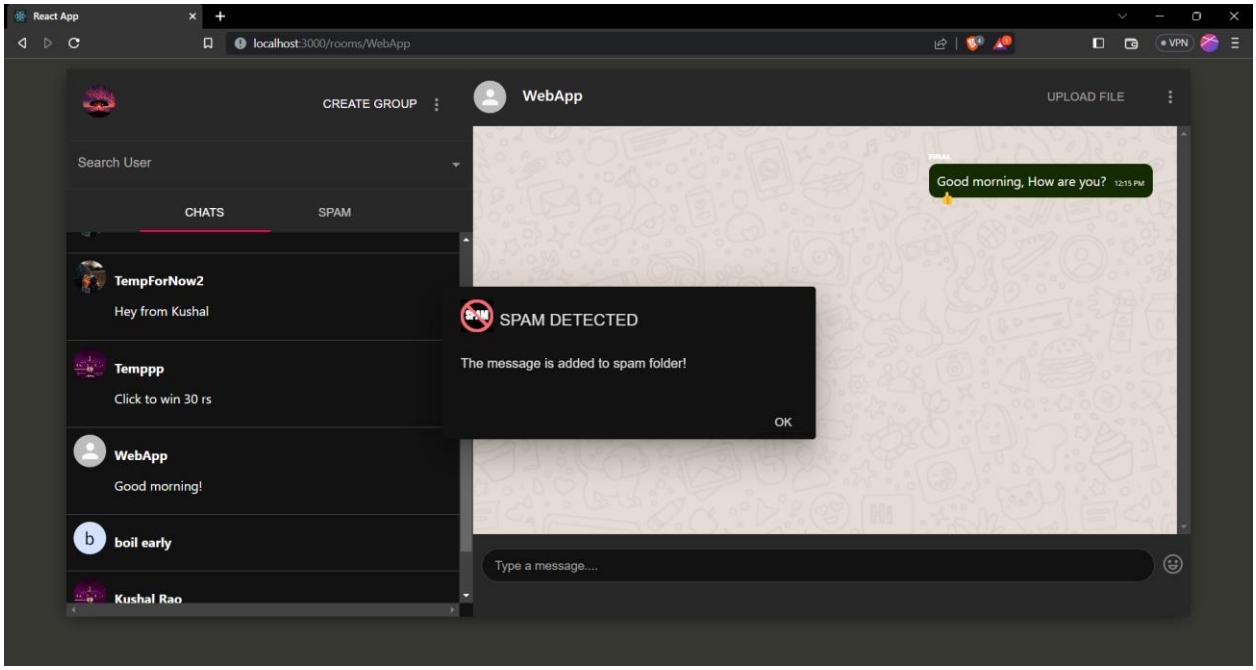


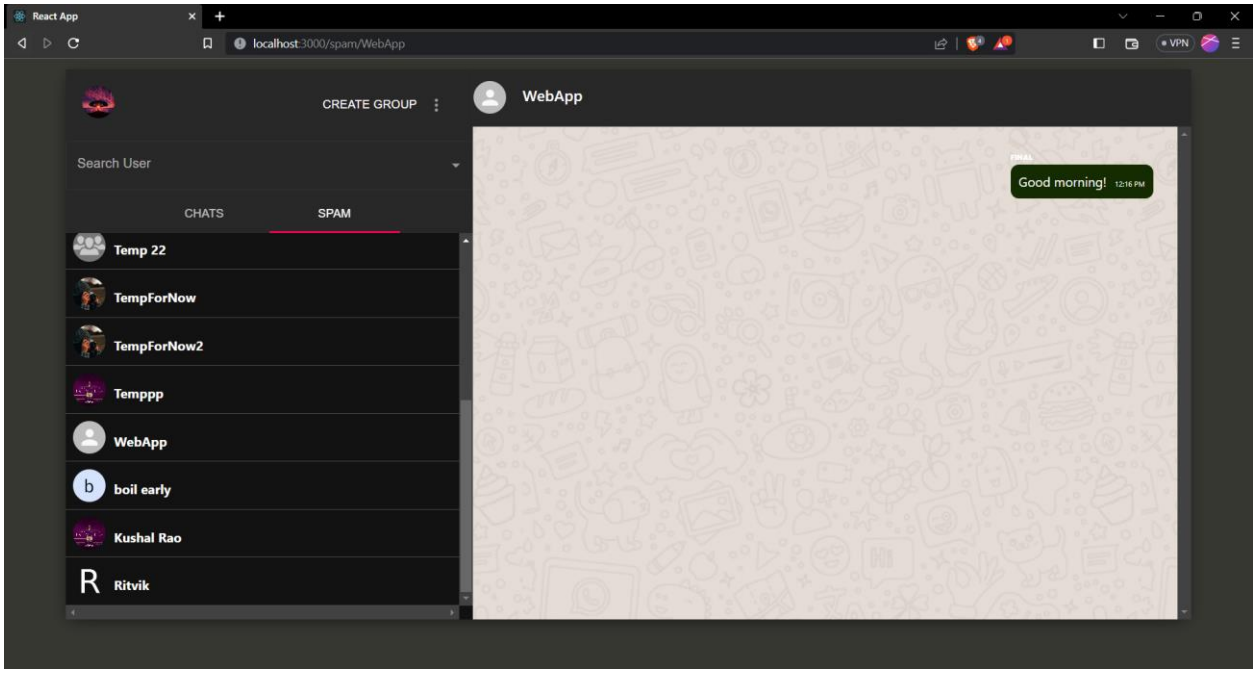
5.2.12 TC_12 - Start an Instant Chat on Mobile App

5.2.13 TC_13 - Sending ham messages in group chat on WebApp

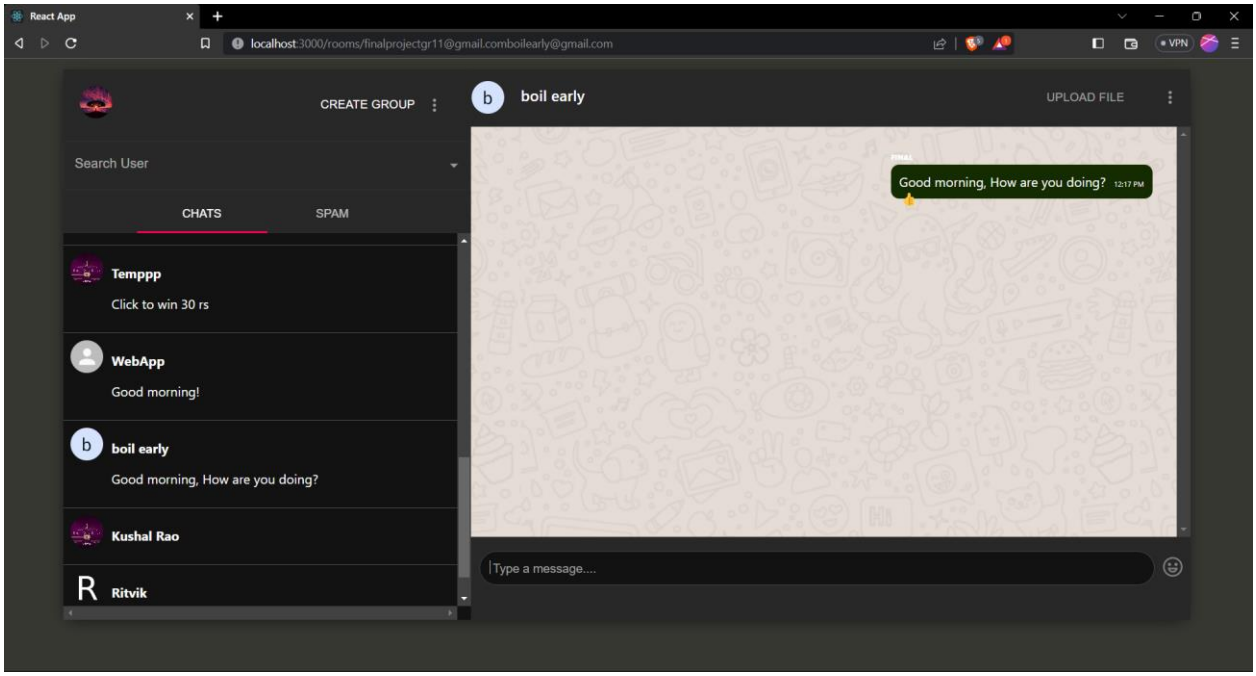


5.2.14 TC_14 - Sending spam message in group chat on WebApp

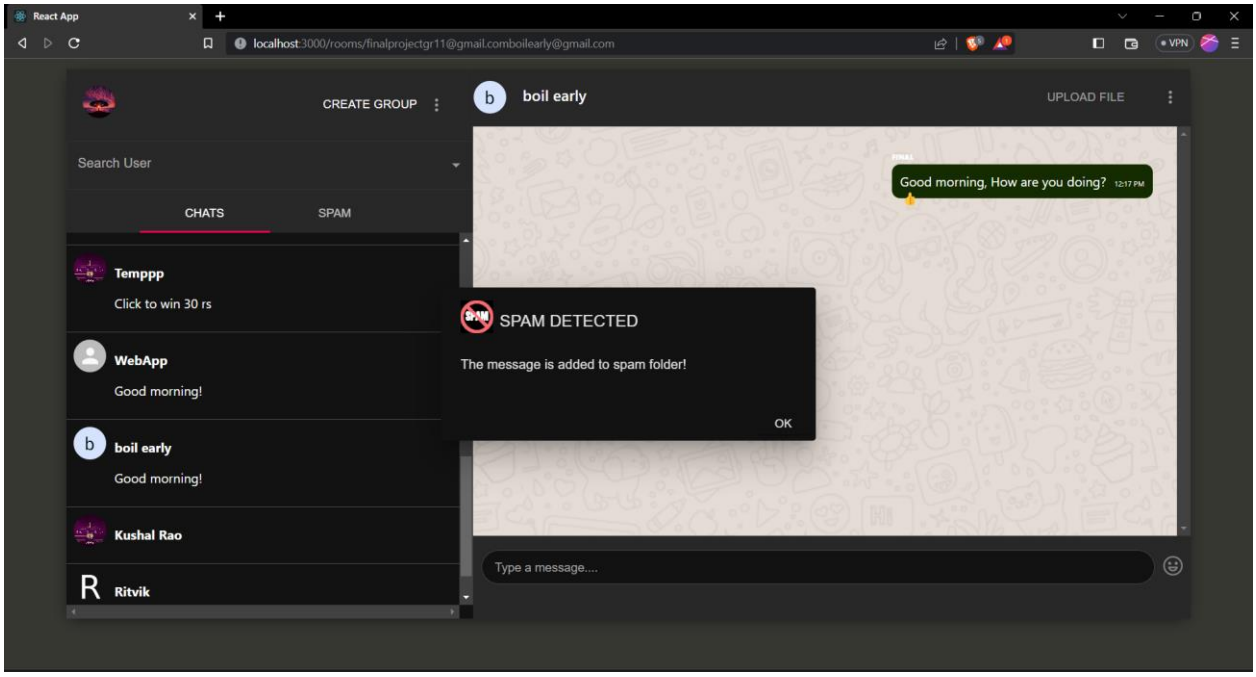


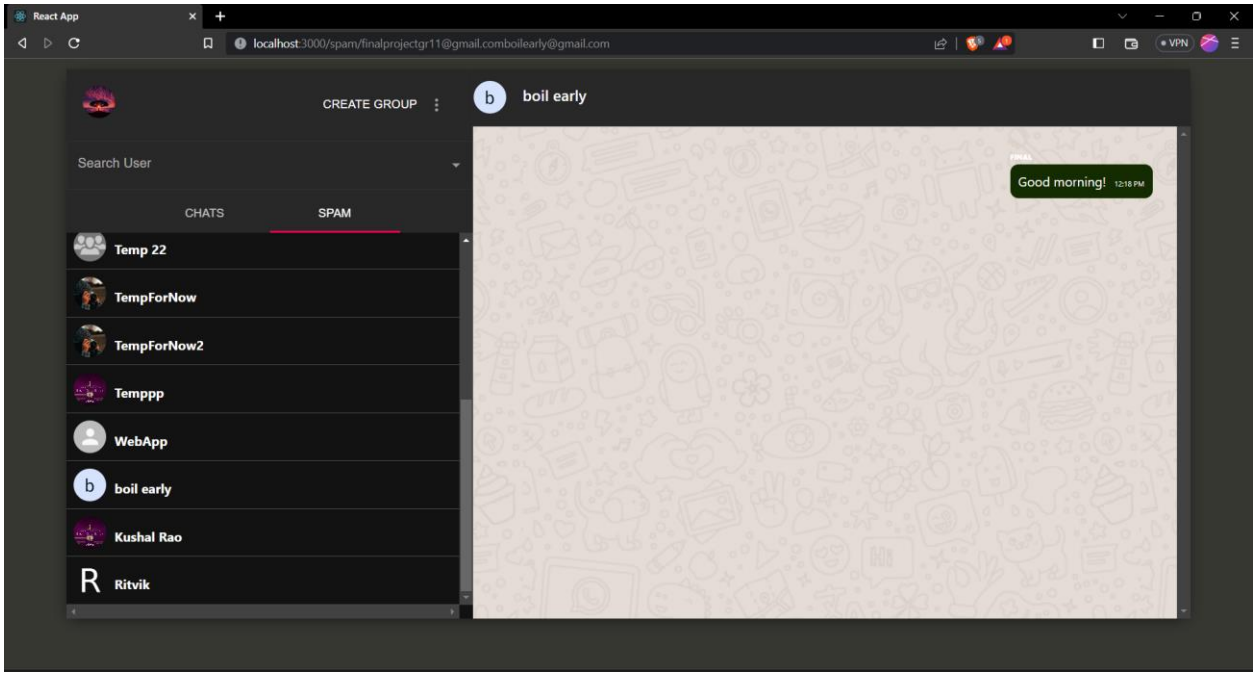


5.2.15 TC_15 - Send ham message in instant chat on WebApp



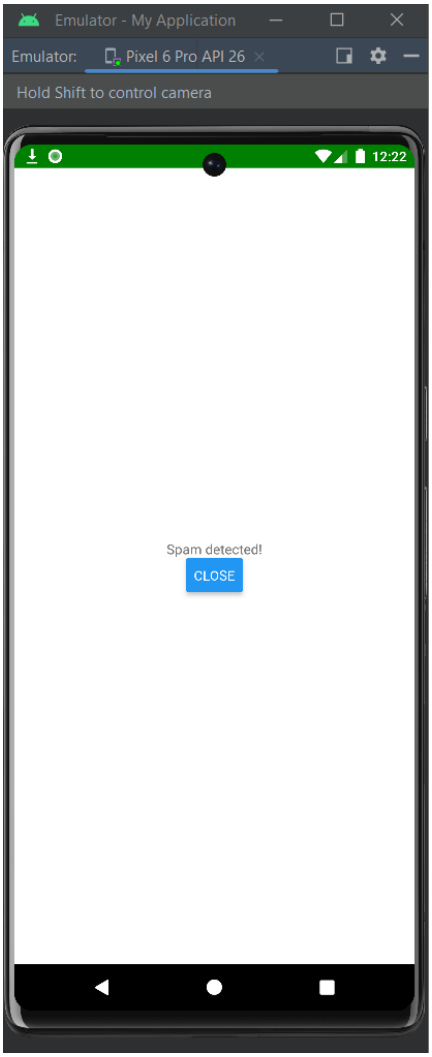
5.2.16 TC_16 - Send spam message in instant chat on WebApp



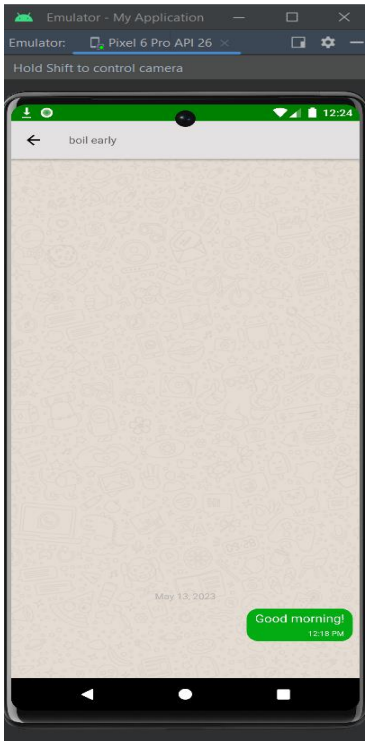


5.2.17 TC_17 - Send ham message in group chat on Mobile App

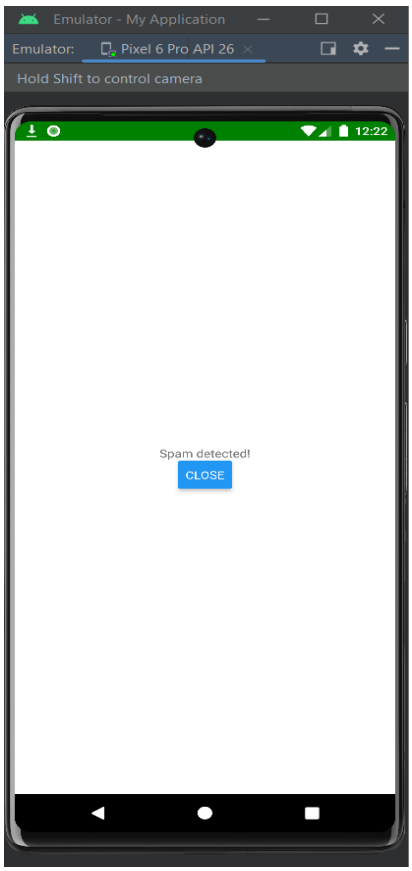
5.2.18 TC_18 - Sending spam message in group chat on Mobile App



5.2.19 TC_19 - Sending ham message in instant chat on Mobile App



5.2.20 TC_20 - Sending spam message in instant chat on Mobile App



06. PROJECT IMPLEMENTATION

6.1 OVERVIEW OF PROJECT MODULES

1.Login Functionality-The user can sign in using his/her gmail account.Once the user signs for the first time the email and the email profile picture url get saved in PostgreSQL.There is no register functionality for the application.Django has been used for login functionality.

2.Frontend- For frontend in the web application,React JS has been used.Material UI library has been used in web application.For mobile application,the development is done using React Native which is supported on both android and ios devices.Gifted chat library is used for displaying the chats in the mobile application using React Native.

A user after login is shown the chats with different users.The user can enter any name in the search bar at the top and search for a particular user to chat with. On clicking on any user's name, the chat screen for that user is opened.A user can send text,image,video and pdf messages to another user.On sending the message, if the message is spam then a popup appears that the message is spam and the message gets deleted from the normal messages screen.These spam messages are displayed on a separate screen.The user can create a group for group chats.Group chats here work in the same way as other chat applications.

3. LSTM Model for spam detection- For spam detection in messages, LSTM model has been used. The model is trained on the benchmark Kaggle spam detection dataset, which contains a total of 5,574 SMS messages. The dataset is split into training and testing sets, with 80% for training and 20% for testing. The trained model is evaluated on the testing set to measure its accuracy.The accuracy of the model on the training data is 99.91%. The accuracy of the model on the testing set is 98.73%.This indicates that the LSTM model is highly effective in detecting spam messages.

Model Architecture & Training-The LSTM model used in spam detection, is a type of recurrent neural network (RNN) that is designed to handle long-term dependencies.

The LSTM model is trained on the LabelEncoders tokenizers to predict whether a message is spam or not. The architecture of the LSTM model is as follows:

- Input Layer: Embedding layer -> (189, 32)
- LSTM Layer: lstm -> (100)
- Dropout_1: dropout -> (0.4)
- Dense_1: dense -> (20, activation = "relu")
- Dropout_2: dropout -> (0.3)
- Dense_2: dense -> (1, activation = "sigmoid")

Finally, the model is trained over 5 epochs and in batch size of 32 with Adam optimizer and binary cross-entropy loss function.

3.Backend- The trained Spam Detection model files are hosted using Django and all the user activities, happening at the mobile application or web application, go through the django backend.Various endpoints have been established which serve their specific purposes like "login/",”^user$/","chat_message/","searchUsers/",etc.Depeding on the type of request from the frontend, the requests are routed to the respective endpoints which then, in turn, sends the appropriate response back to the user.Firebase has been used for storing all the messages of the web application and mobile applications.

6.2 IMPLEMENTATION AND BASIC LOGIC

DataSet- We make use of an SMS spam dataset that Almeida and Hidalgo suggested. The number of records in this dataset is roughly 5,574. It has SMS text messaging in it. Talks in the English language that use text and numbers in phrases of varying lengths. This dataset's records are all already labelled. The normal communications are labelled as 0 and the spam ones as 1 (747 records) (4,827 records).

Data Preparation- Natural Language Processing (NLP) is used in this procedure to pre-process natural language data. NLP is a process that enables computers to comprehend natural language in a manner that is comparable to that of a human [15]. It uses a variety of methods to preprocess data into an understandable format for computers. In this study, we employ NLP approaches to convert SMS text data into sequence data so that we may use it to create SMS classification models using the LSTM and GRU algorithms. Additionally, we

pre-process data using word tokenization, padding, truncating, and word embedding algorithms. The following is a description of each approach we employ for data pre-processing in detail.

Word Tokenization-The process of turning the words in a sentence into index values denoted by a number is called word tokenization. In this procedure, we generate a word tokenizer using a specified number of interesting vocabulary terms. Word tokenizer is used to turn the words of a sentence into sequence data after being created. Tokenization converts words into indexes and sets the index to 0 for terms that are unknown.

Padding- In this procedure, we uniformly lengthen every sequence in the dataset for LSTM and GRU training.We determine the message length that is optimised using (1). After the message's length has been optimised, we pad data that is shorter than the ideal length by inserting a zero at the beginning of the sequence until it is the same length as the ideal one.

Word Embedding-This method transforms a word sequence that has already undergone preprocessing into a vector representation known as an embedding space that has more dimensions than the standard word data used to train LSTM and GRU algorithms. After padding and truncating the data, we employ the word embedding technique, setting the embedding size to 32, to add more dimensions for the data in order.

Modeling- We create categorization models for SMS spam based on LSTM algorithm, a deep learning technique.

LSTM- Hochreiter and Schmidhuber created the LSTM in 1997. By including cell states for remembering or forgetting data, it enhances the fundamental RNN method that resolves the vanishing problem. Cell gates are a type of structure found in the cell states. The input gate, forget gate, memory-cell state gate, and output gate are the four components that make up the cell gates. The input is utilised as a gate to control whether the input data is valuable enough to maintain or not. The prior hidden state that needs to be preserved in the memory cell of the current hidden state is controlled by the forget gate. Based on the data from the input gate and the forget gate, the memory-cell state gate is utilised to update the data. Based on the state of the memory cells, the output gate computes the network's output data.

A memory cell known as a "cell state" that preserves its state over time plays a key role in an LSTM model. The horizontal line that passes through the top of the diagram below represents the cell state. It can be pictured as a conveyor belt across which data simply and unaltered passes. The cell state may contain important information while the sequence is processed. As a result, even knowledge from earlier time steps might reach later time steps, diminishing the impact of short-term memory. Information is added to or withdrawn from the cell state via gates as the cell state travels. The gates, which determine which information is permitted on the cell state, are various neural networks.

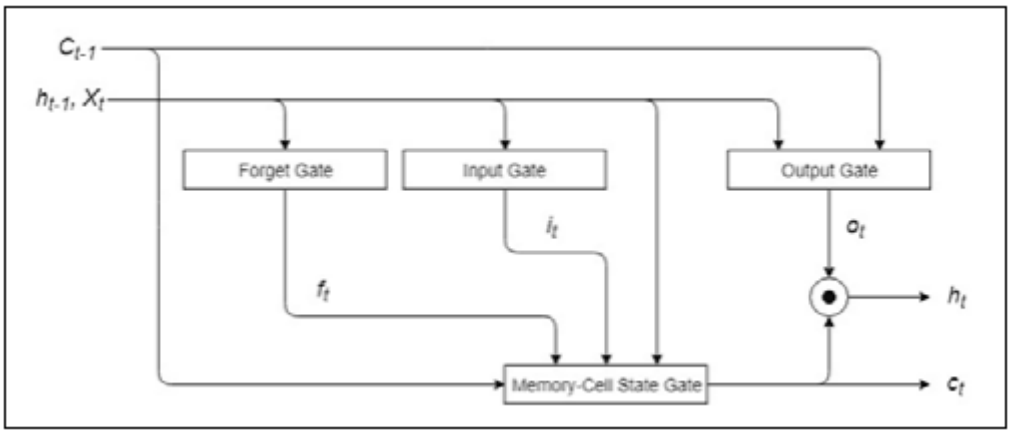


Fig 6.1 LSTM Memory Cell

6.3 ALGORITHM

Creating SMS Spam-Detection Model using LSTM:

1. Import SMS spam detection dataset from kaggle.

https://www.kaggle.com/uciml/sms-spam-collection-dataset

2. Label each SMS with spam or ham accordingly.
3. Perform pre-processing operations on the dataset.
4. Remove unnecessary columns
5. Remove stopwords
6. Vectorize inputs
7. Tokenization of inputs
8. Pad each input with 0s so that each input is of equal length

9. Create a 6-layer LSTM model with Dense and Dropout layers in between to avoid overfitting.
10. Use Adam optimizer and binary-cross entropy loss function to adjust weights of the model during training optimally.
11. Train the model using model.fit() function.
12. Store the trained weights.

Create a Backend Server:

1. Host the SMS Spam-Detection trained model on the server.
2. Open a port on which the frontend of the chatbot can communicate on.
3. Accept string/user messages from the frontend and give it as an input to the trained SMS Spam-Detection Model.
4. The model will return its predictions to the backend server, which will then send the predictions back to the frontend.

Create a Frontend Server:

1. Connect the frontend server to the frontend of the Chatbot (Mobile App/Web App)
2. For each client, create a thread to keep each client's request isolated from the other clients.
3. Accept user messages when the user hits the "send" button.
4. Send the messages to the backend server via the communicable port established at the backend server and wait for a response.
5. If the response is "spam", exit.
6. If the response is "ham", send an appropriate response to the user.

07. RESULTS

7.1 SCREENSHOTS

```
# check_spam = ["Free entry in 2 a wkly comp to win FA Cup finals"]
# check_spam = ["Nah I don't think he goes to usf"]
check_spam_msg = input()
check_spam_msg = [check_spam_msg]
sms_spam_tokenizer.fit_on_texts(X)
check_spam_tokenized = sms_spam_tokenizer.texts_to_sequences(check_spam_msg)
print(prediction(check_spam_tokenized, sms_spam_tokenizer))
check_ham_msg = input()
check_ham_msg = [check_ham_msg]
# print(check_spam_ham_msg)
check_ham_tokenized = sms_spam_tokenizer.texts_to_sequences(check_ham_msg)
# print("Text               : ", check_spam_ham_msg[0] )
# print("Numerical Sequence : ", check_spam_tokenized[-1])

print(prediction(check_ham_tokenized, sms_spam_tokenizer))
```

```
Free entry in 2 a wkly comp to win FA Cup finals
1/1 [==============================] - 0s 18ms/step
SPAM!
Nah I don't think he goes to usf
1/1 [==============================] - 0s 17ms/step
HAM!
```

Fig 7.1: LSTM Model for Spam Detection

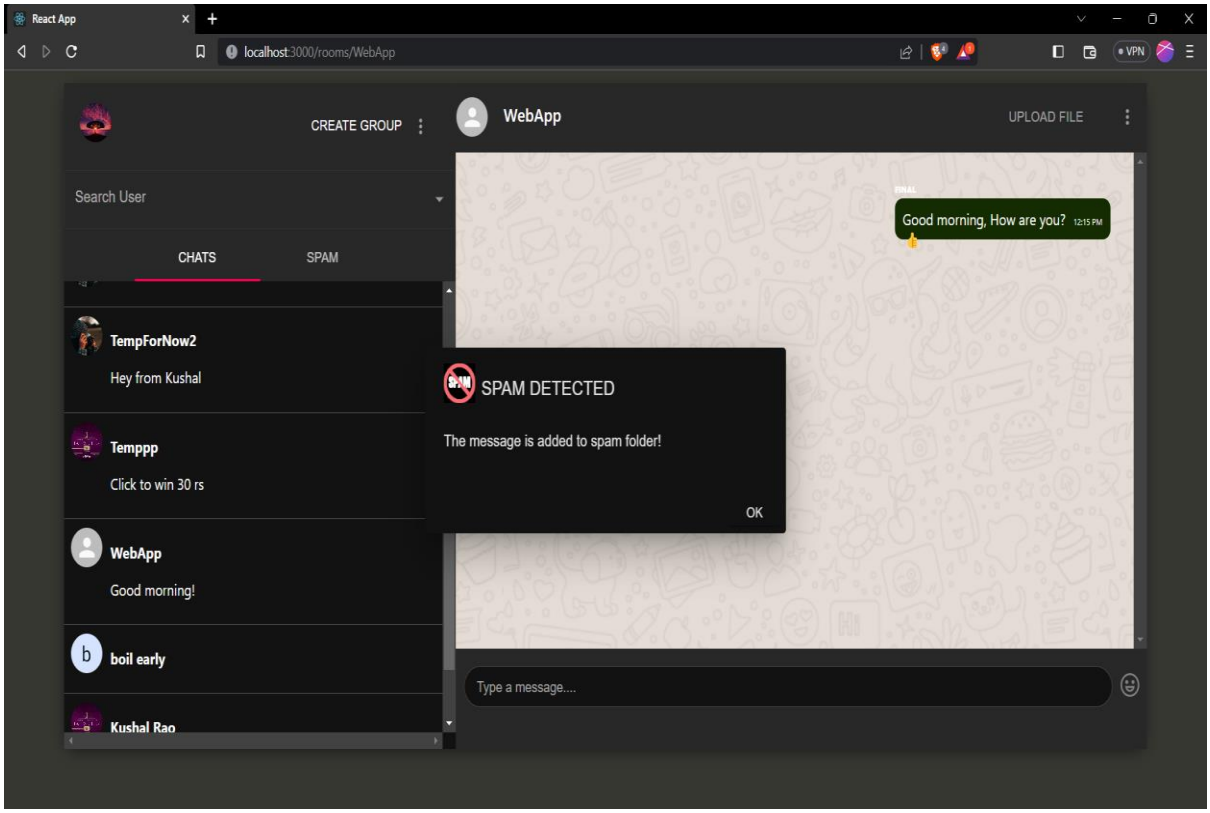


Fig 7.2 Sending a spam message on web application

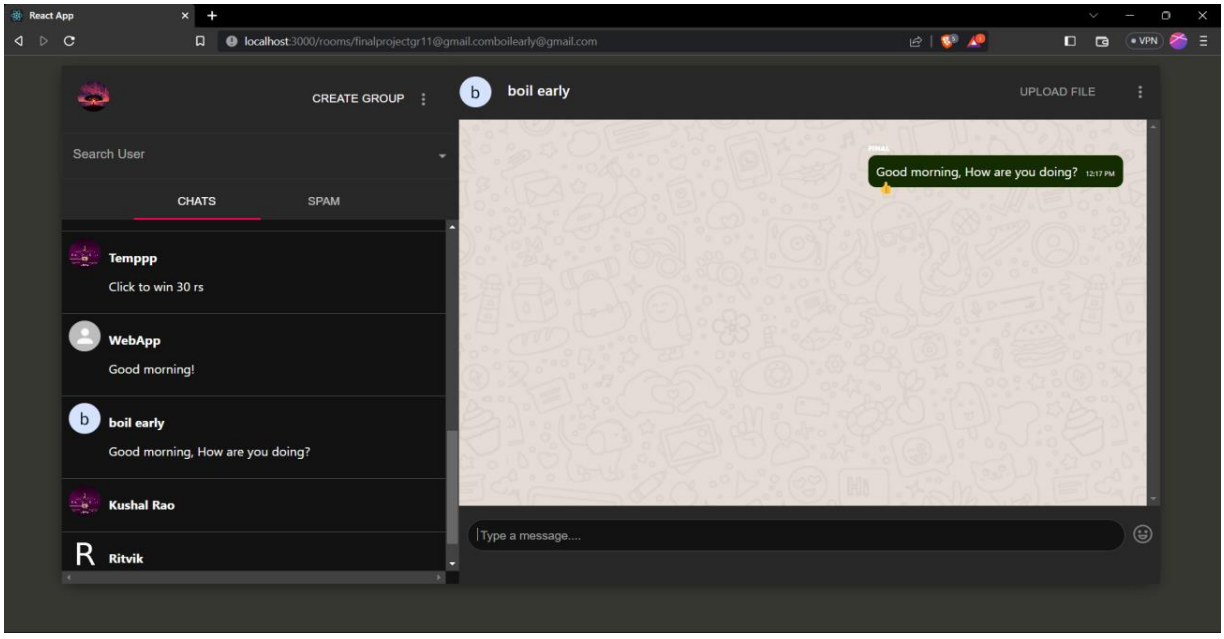


Fig 7.3 Sending a ham message on web application

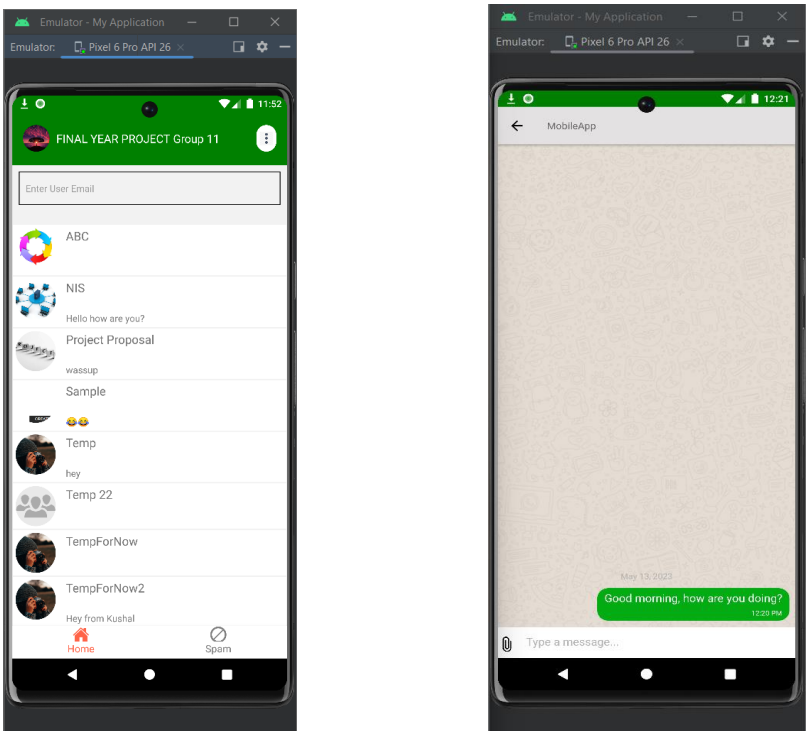


Fig 7.4 Sending a ham message on mobile application

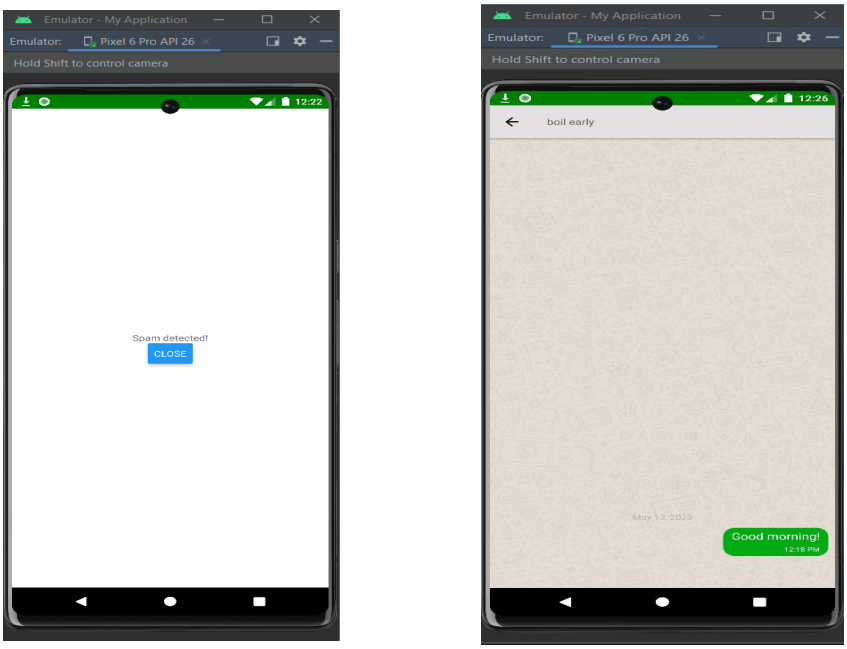


Fig 7.5 Sending a spam message on mobile application

08. CONCLUSION

8.1 CONCLUSION

Thus we have built a chat application (both web and mobile versions) that provides spam classification for text messages.The spam classification is based on LSTM (a deep learning algorithm). NLP methods were utilized to pre-process the text messages.These include word tokenization, padding, truncating, and word embedding techniques to put text data into sequence.The model has been trained on Kaggle dataset that contains 5,574 messages.Additionally for training against specific spam messages,different data was used.The chat application currenty provides good protection against spam messages and has almost all the functionalities of a regular chat application.This includes sending all types of messages(video,images,pdf files), giving emoji reactions, group chats,etc.

8.2 FUTURE WORK

Here we have developed mobile and web applications for chat app using spam detection.The spam detection is done for text messages and LSTM model (Long-Short Term Memory) has been used.The model is trained on the benchmark Kaggle spam detection dataset, which contains a total of 5,574 SMS messages. The dataset is split into training and testing sets, with 80% for training and 20% for testing.We would add more data for training and testing.We would add last seen functionality for the users and block feature to block certain users.

8.3 APPLICATIONS

The project has been sponsored by BlinkAds and is for the company's internal use.It is available for both mobile and desktop devices.It provides highly efficient spam detection for text messages.Using this chat application, the users can send text, image, video and pdf messages to each other.Functionality for group messaging and emoji reactions is also available.The normal messages are shown on one screen and spam messages are shown on another screen.This is very useful to the user and saves his/her time spent on reading and deleting spam messages.

APPENDIX B: PLAGIARISM REPORT

turnitin

Similarity Report ID: oid:8054:35934358

PAPER NAME

report_final_year_sem2.docx

WORD COUNT	CHARACTER COUNT
5852 Words	31420 Characters
PAGE COUNT	FILE SIZE
55 Pages	8.8MB
SUBMISSION DATE	REPORT DATE
May 22, 2023 8:29 PM GMT+5:30	May 22, 2023 8:30 PM GMT+5:30

7% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 6% Internet database
- 4% Publications database
- Crossref database
- Crossref Posted Content database

Excluded from Similarity Report

- Submitted Works database
- Quoted material

REFERENCES

[1] M.Rubin Julis, S.Alagesan, “Spam Detection In Sms Using Machine Learning Through Text Mining”, 2020

[2] . F. Kai Petersen, Shahid Mujtaba, Michael Mattsson, "Systematic Mapping Studies in Software Engineering," 2008.

[3] Muhammad Iqbal, Malik Muneeb Abid, Mushtaq Ahmad, Faisal Khurshid, “Study on the Effectiveness of Spam Detection Technologies”, 2016

[4] Luo GuangJun, Shah Nazir, Habib Ullah Khan, Amin Ul Haq, “Spam Detection Approach for Secure Mobile Message Communication Using Machine Learning Algorithms”, 2020

[5] Nikhil Kumar, Sanket Sonowal, Nishant, “Email Spam Detection Using Machine Learning Algorithms”, 2020

[6] Pumrapee Poomka, Wattana Pongsena, Nittaya Kerdprasop, and Kittisak Kerdprasop,— SMS Spam Detection Based on Long Short-Term Memory and Gated Recurrent Unit , 2019

[7] https://www.analyticsvidhya.com/blog/2021/05/sms-spam-detection-using-lstm-a-hands-on-guide/