FLUTO MESSENGER APPLICATION

(WEB APPLICATION)

TEAM - 15

K. ASHA 1602-19-737-005

K.PALLAVI SUMA 1602-19-737-025

P.KEERTHANA 1602-19-737-183

PROBLEM STATEMENT

- In the present generation there are many more social media applications where we can interact with each other.
- People who are in social media are prone to mental abuses by unwanted messages , spam messages, threat and hate messages. Due to these problems, people are facing many severe issues.
- Our Fluto messenger Application is an attempt to free people from such situations by categorizing messages into spam messages and asking the user's permission for viewing them.

APPROACH/SOLUTION

- The main feature of messenger application is categorizing spam messages in the messaging .
- We chose Random Forest, Support Vector Machine algorithms to detect and classify messages as spam and ham.
- The messages considered as spam will be classified under spam section in chat area.

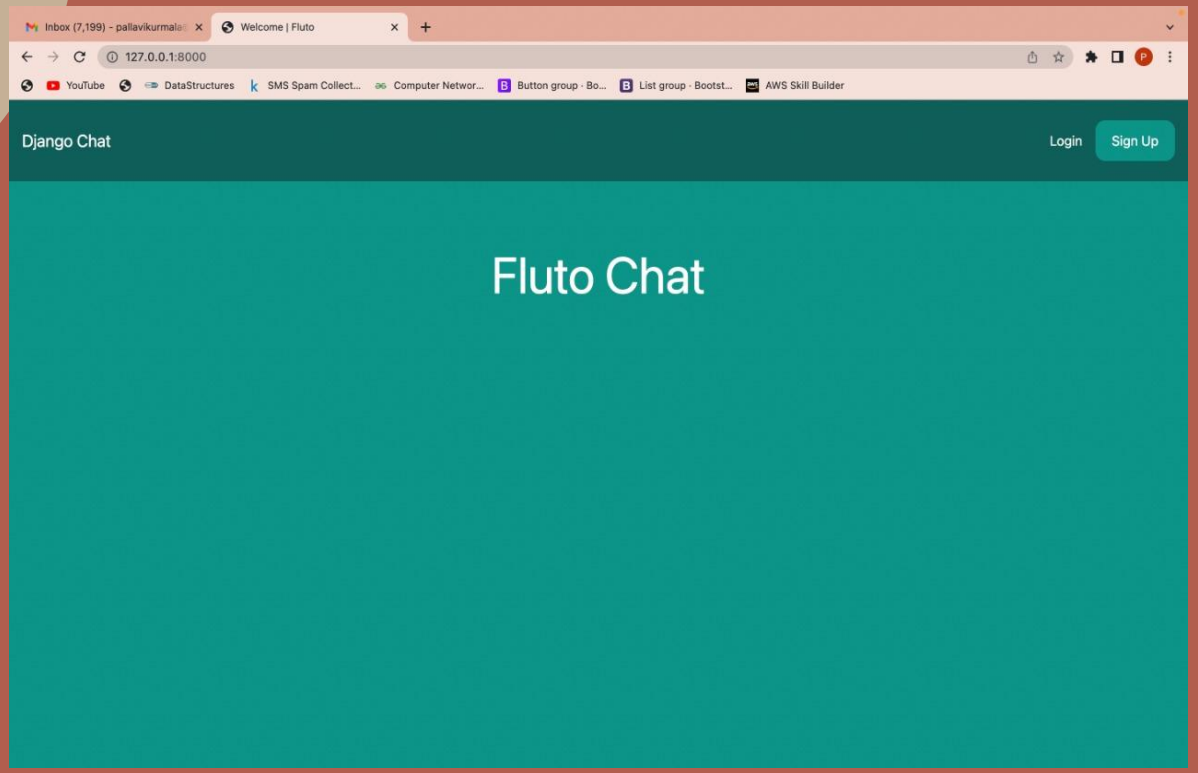
EXTENSION

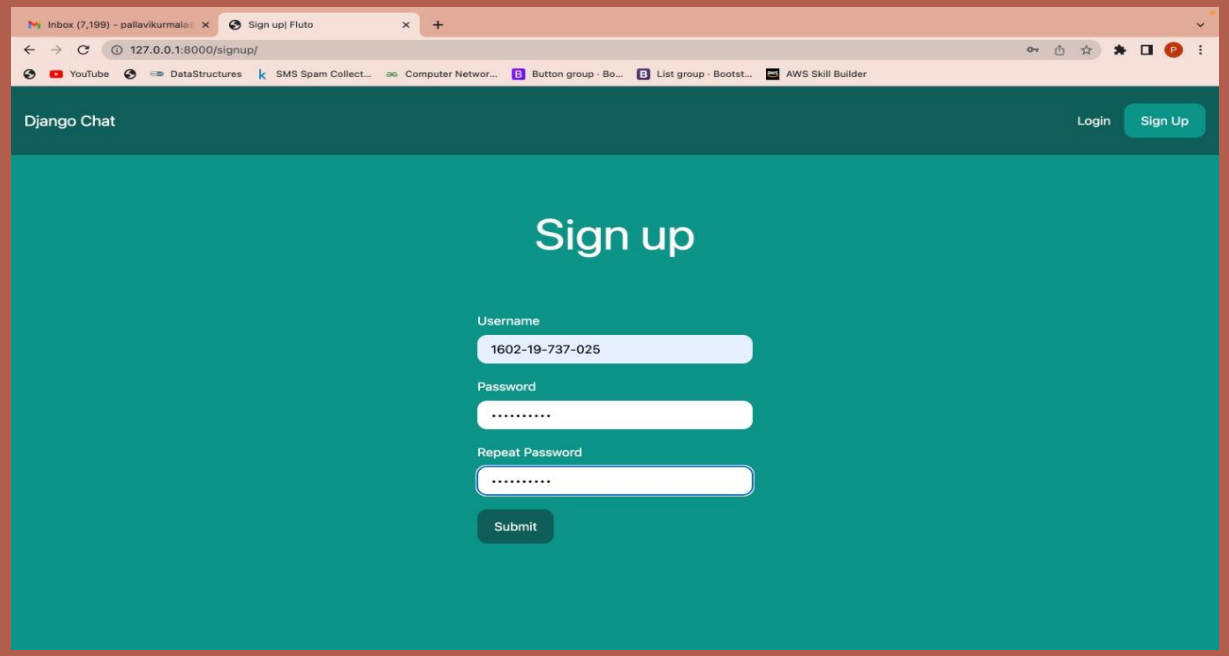
- We have extended project from mobile application to web application.
- We are also adding spam message detection using machine learning algorithm.

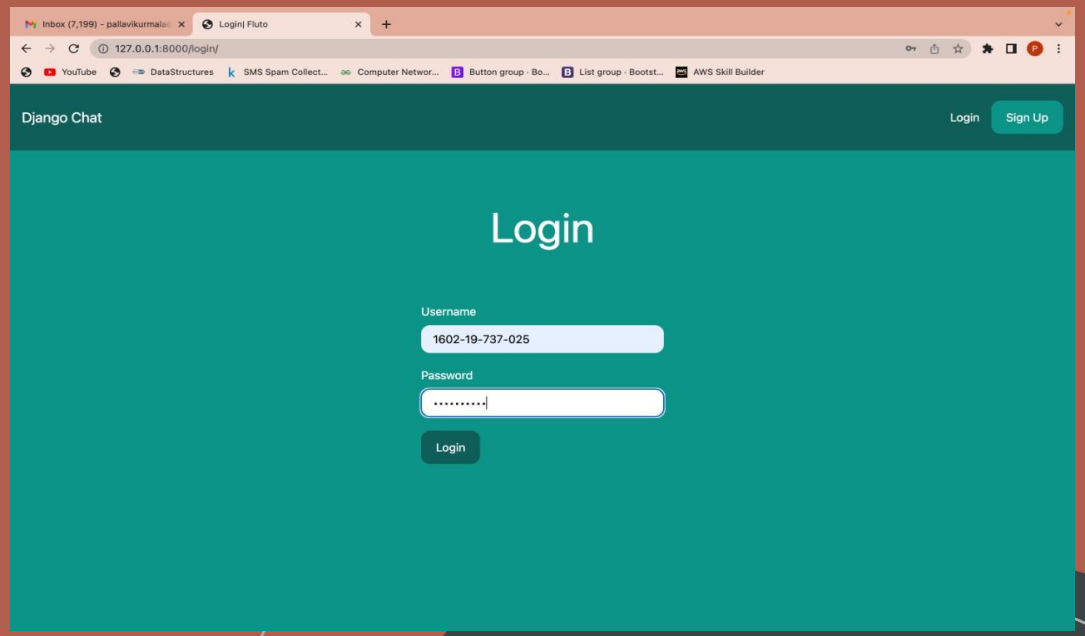
SOFTWARE REQUIREMENTS

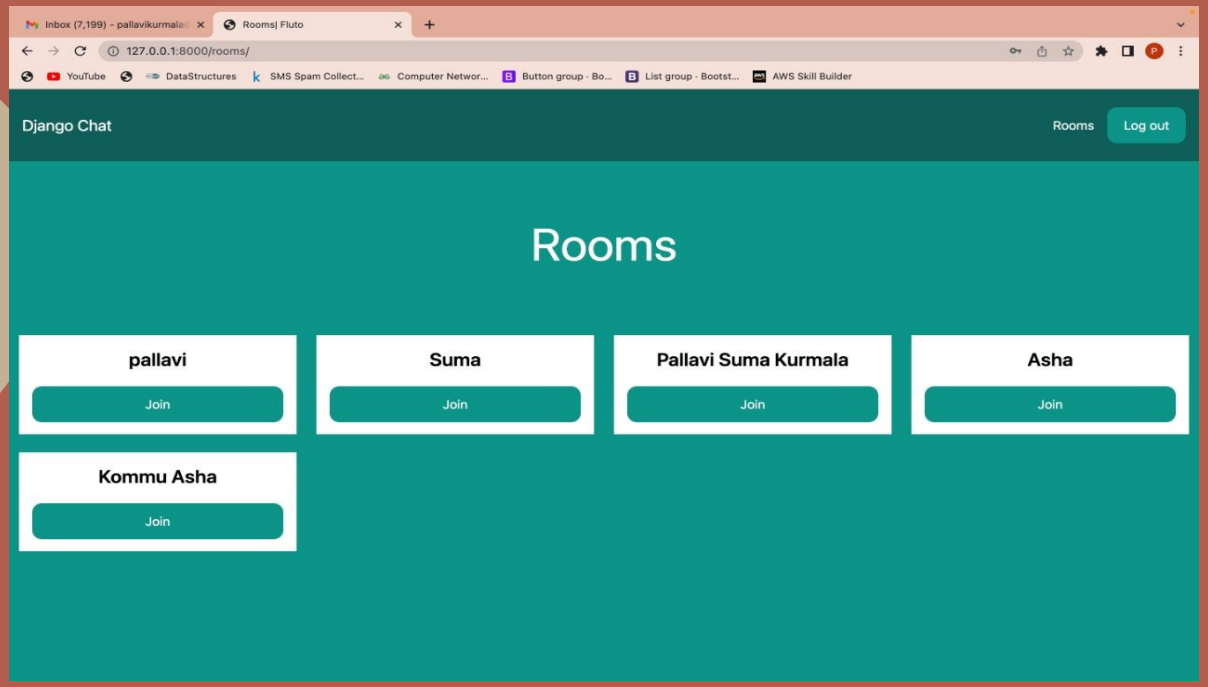
- HTML AND CSS
- JAVA SCRIPT
- Django Frame work.
- Jupyter notebook –for python
- Libraries –numpy , sklearn, mathplot, Pandas

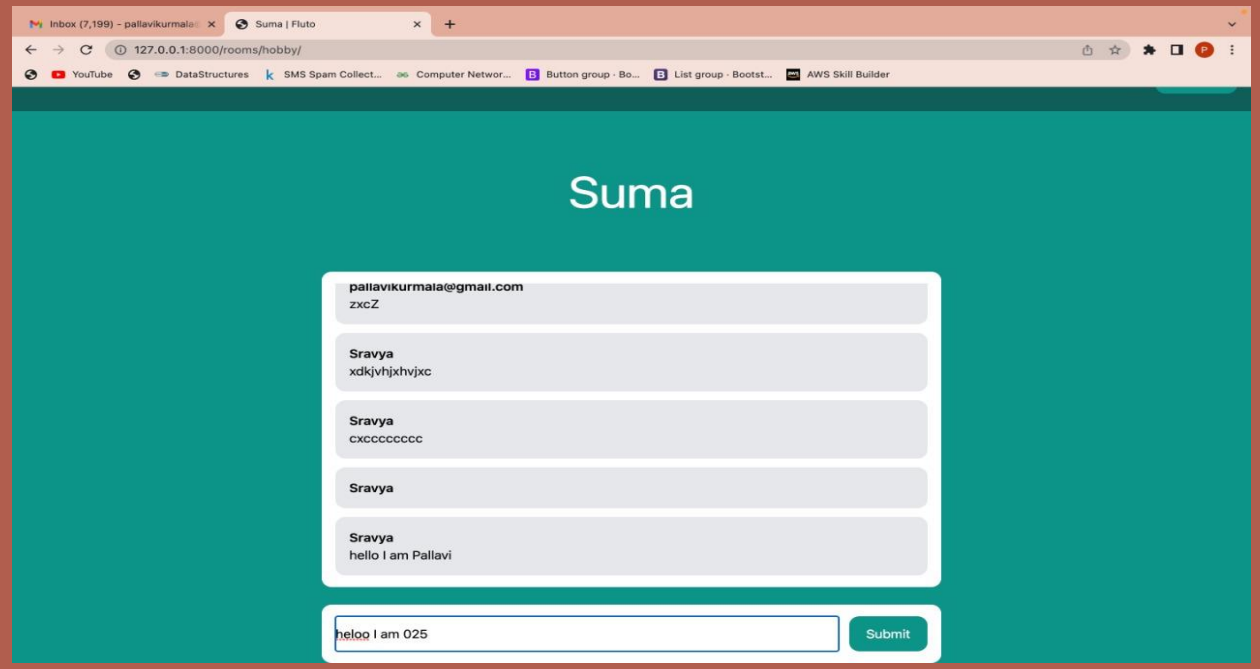
RESULTS

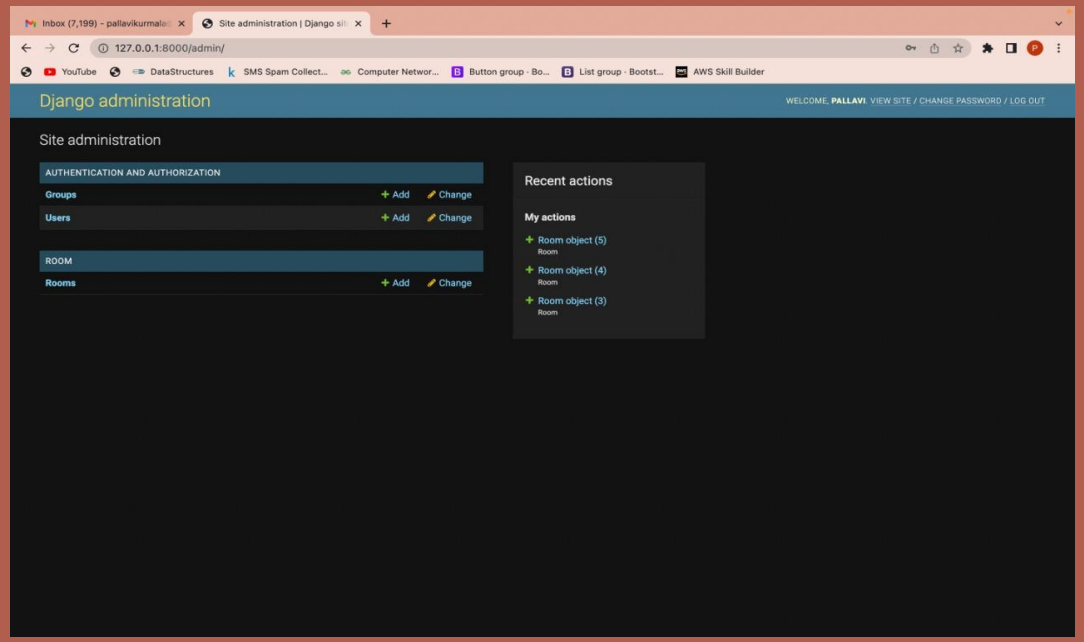

Welcome | Fluto
127.0.0.1:8000
Django Chat
Login
Sign Up
Fluto Chat

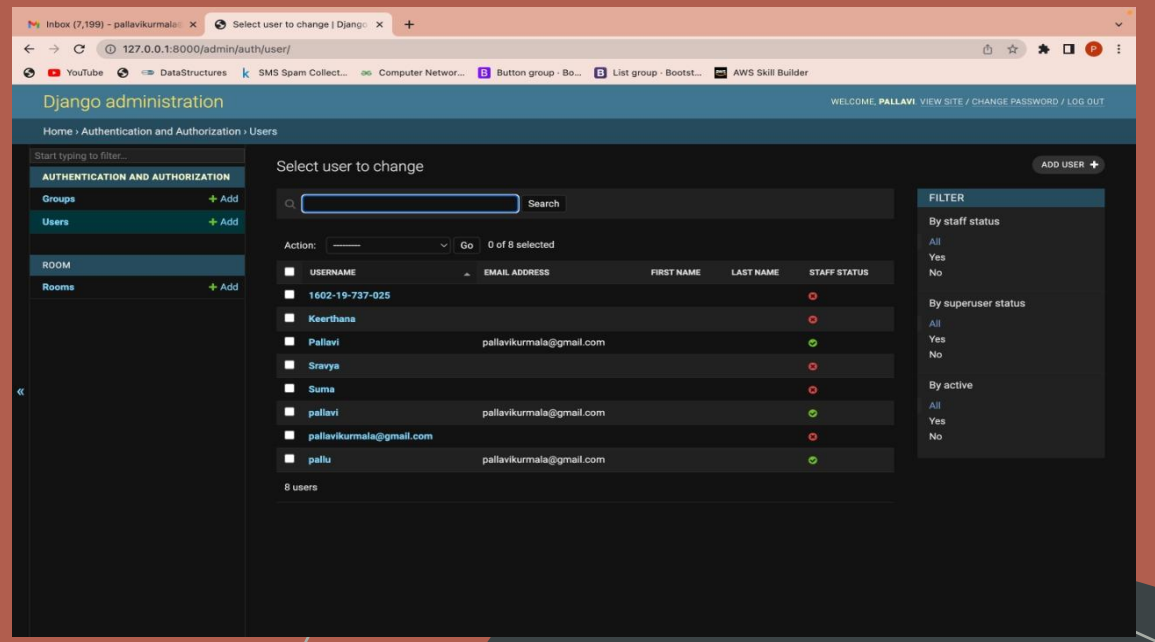

Sign up| Fluto
127.0.0.1:8000/signup/
Django Chat
Login
Sign Up
Sign up
Username
1602-19-737-025
Password
Repeat Password
Submit

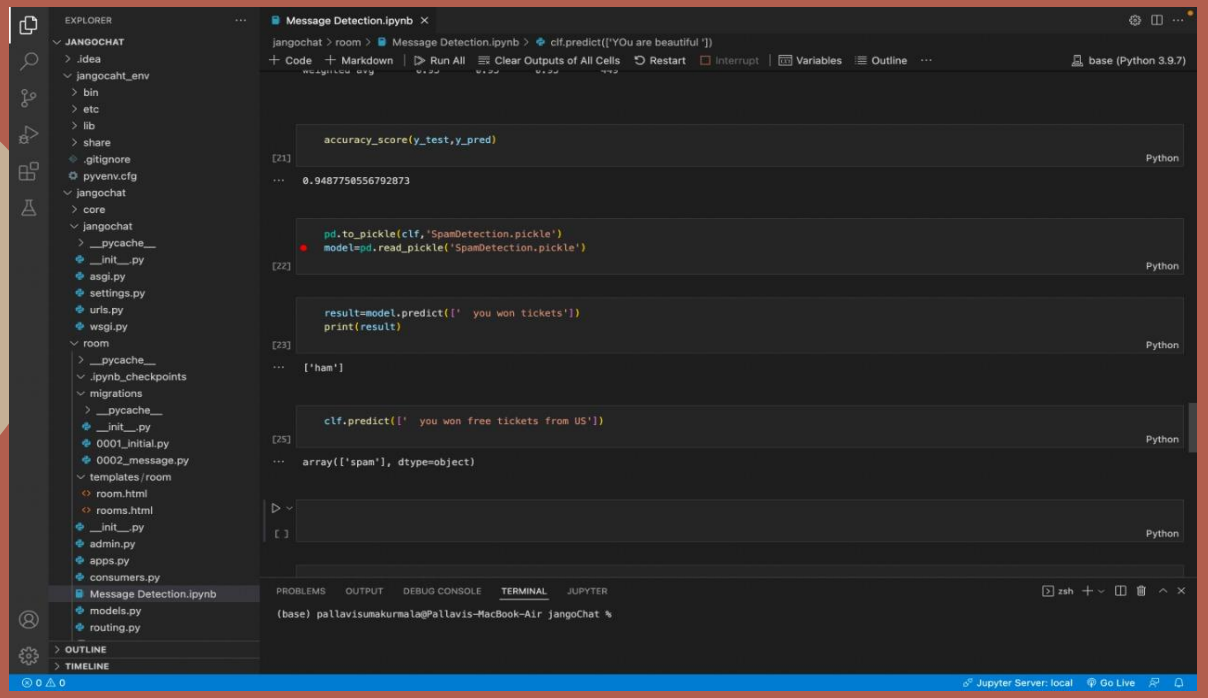

Login| Fluto
127.0.0.1:8000/login/
Django Chat
Login
Sign Up
Login
Username
1602-19-737-025
Password
Login

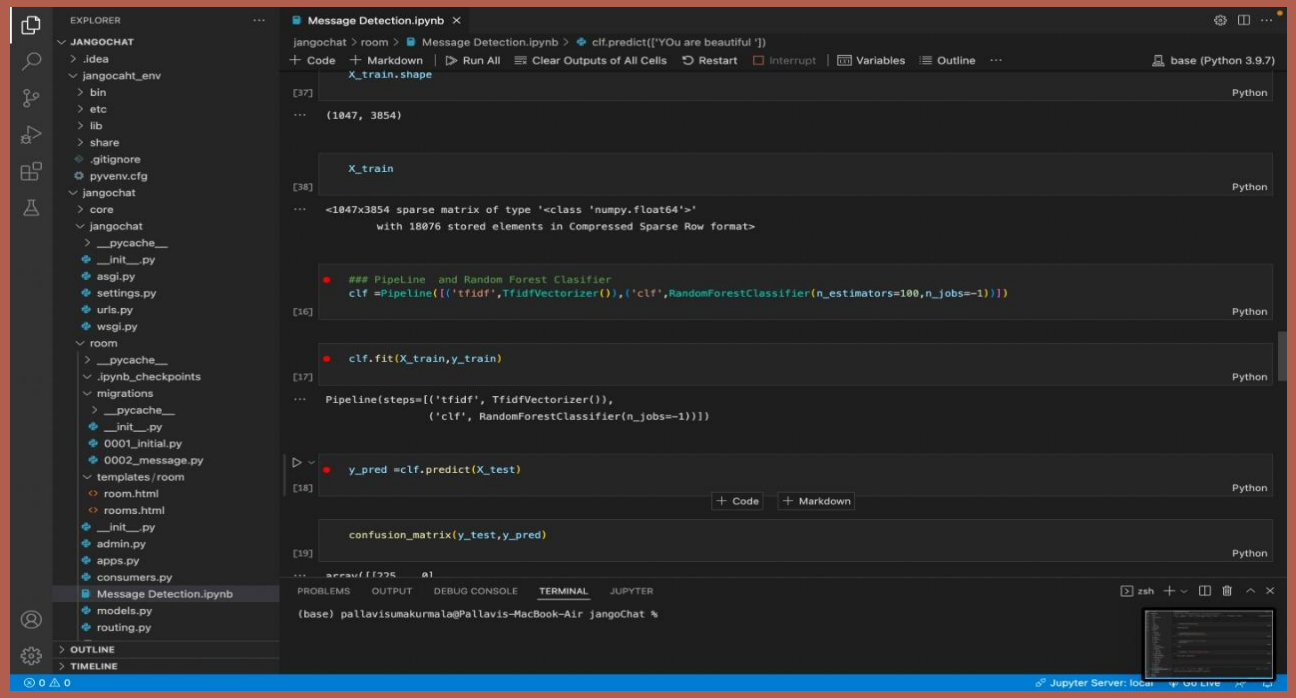

Rooms| Fluto
127.0.0.1:8000/rooms/
Django Chat
Rooms
Log out
Rooms
pallavi
Join
Suma
Join
Pallavi Suma Kurmala
Join
Asha
Join
Kommu Asha
Join

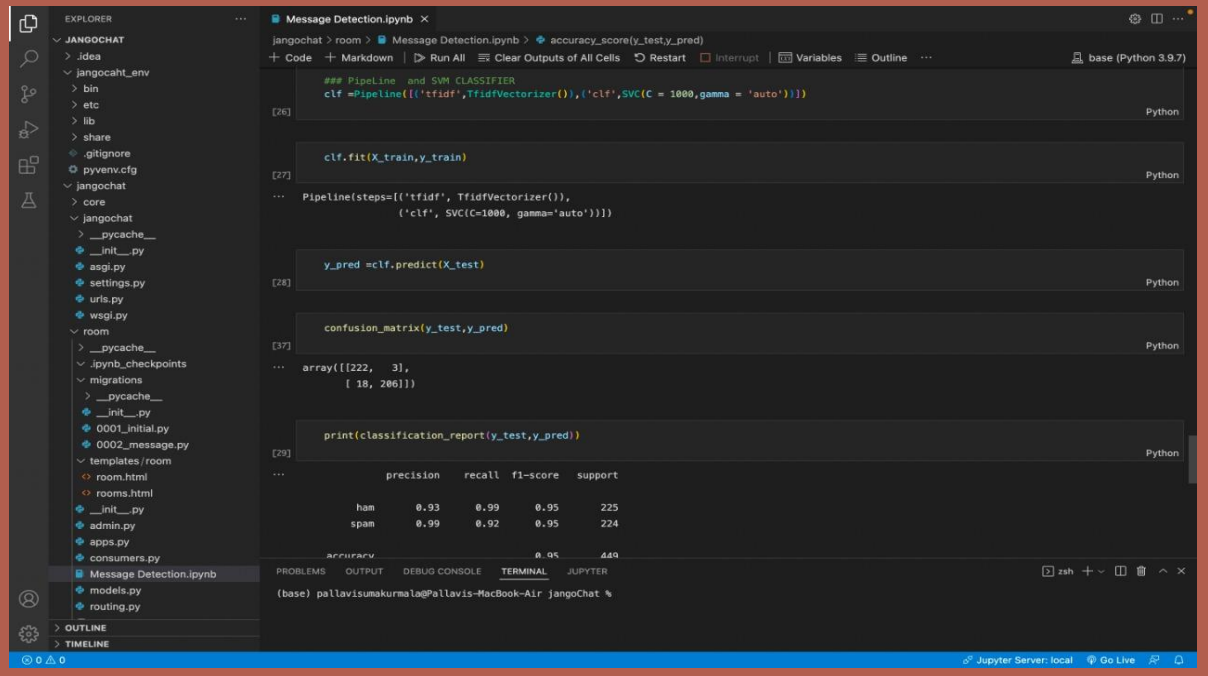

Suma | Fluto
127.0.0.1:8000/rooms/hobby/
Suma
pallavikurmala@gmail.com
zxcZ
Sravya
xdkjvhjxhvjxc
Sravya
cxccccccccc
Sravya
Sravya
hello I am Pallavi
heloo I am 025
Submit

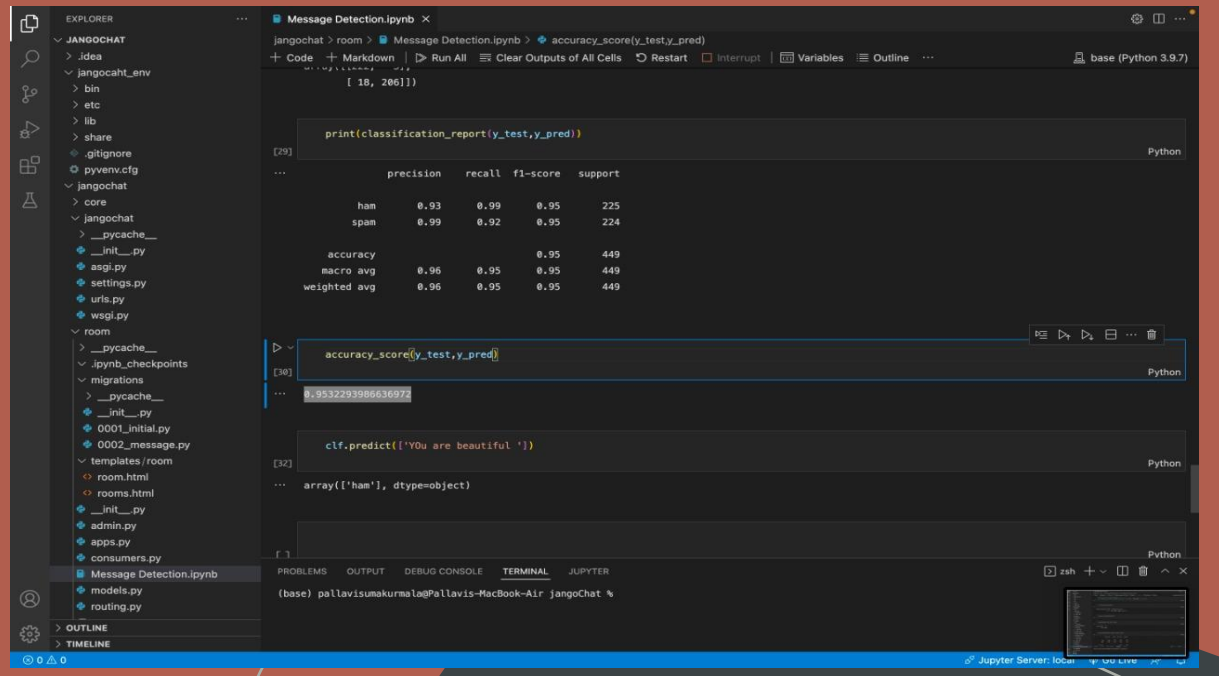

Site administration | Django site admin
127.0.0.1:8000/admin/
Django administration
WELCOME, PALLAVI. VIEW SITE / CHANGE PASSWORD / LOG OUT
Site administration
AUTHENTICATION AND AUTHORIZATION
Groups + Add Change
Users + Add Change
ROOM
Rooms + Add Change
Recent actions
My actions
Room object (5)
Room object (4)
Room object (3)

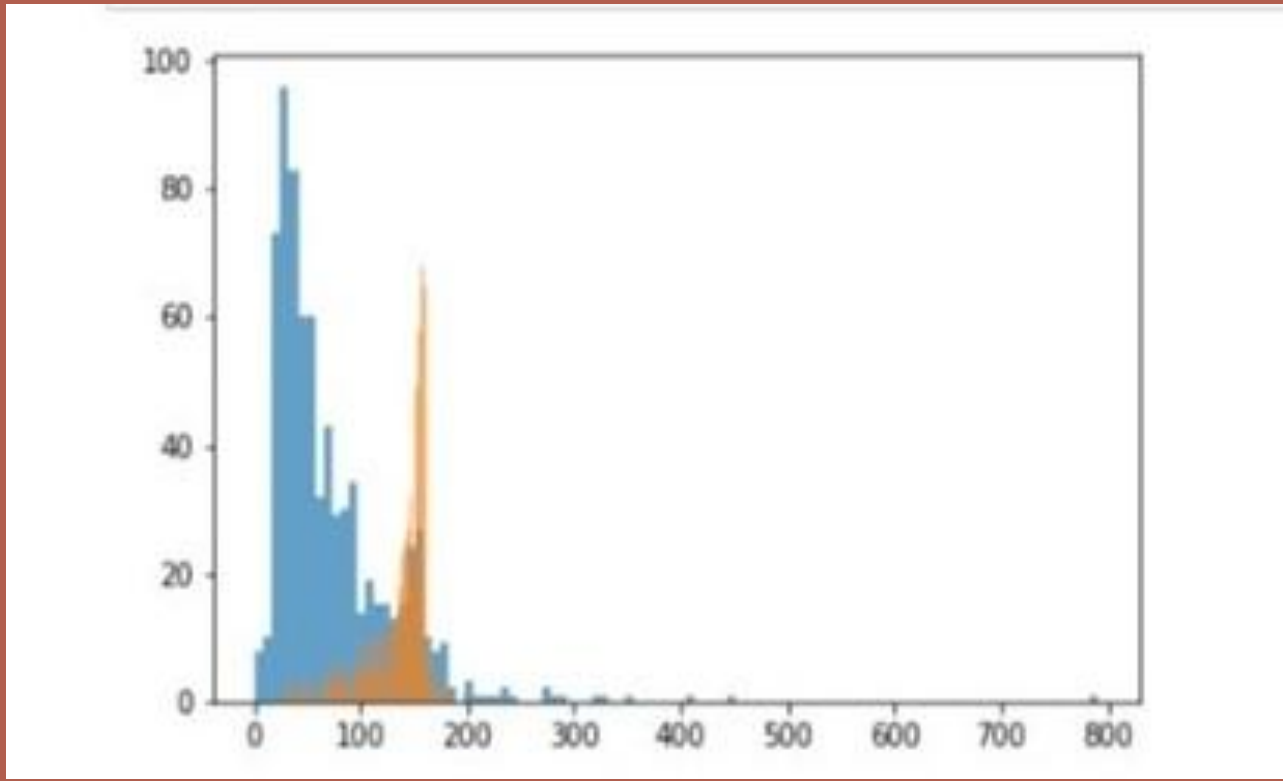

Select user to change | Django site admin
127.0.0.1:8000/admin/auth/user/
Django administration
WELCOME, PALLAVI. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Authentication and Authorization › Users
Select user to change
ADD USER
Search
Action: Go 0 of 8 selected
USERNAME
EMAIL ADDRESS
FIRST NAME
LAST NAME
STAFF STATUS
1602-19-737-025
Keerthana
Pallavi pallavikurmala@gmail.com
Sravya
Suma
pallavi pallavikurmala@gmail.com
pallavikurmala@gmail.com
pallu pallavikurmala@gmail.com
8 users
FILTER
By staff status
By superuser status
By active


Message Detection.ipynb
accuracy_score(y_test,y_pred)
0.9487750556792873
pd.to_pickle(clf,'SpamDetection.pickle')
model=pd.read_pickle('SpamDetection.pickle')
result=model.predict([' you won tickets'])
print(result)
['ham']
clf.predict([' you won free tickets from US'])
array(['spam'], dtype=object)
(base) pallavisumakurmala@Pallavis-MacBook-Air jangoChat %

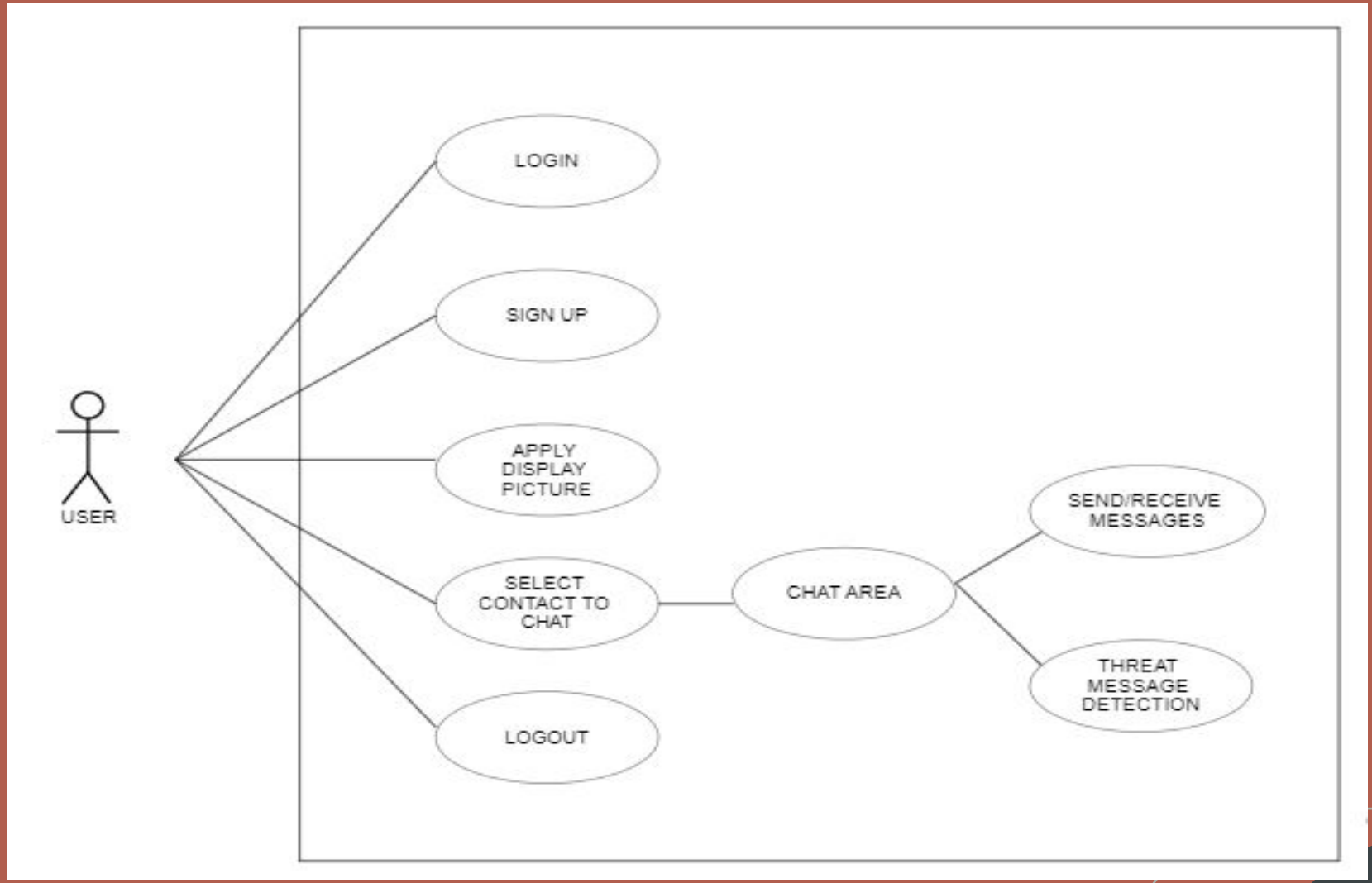

Message Detection.ipynb
X_train.shape
(1047, 3854)
X_train
<1047x3854 sparse matrix of type '<class 'numpy.float64'>'
with 18076 stored elements in Compressed Sparse Row format>
PipeLine and Random Forest Clasifier
clf =Pipeline([('tfidf',TfidfVectorizer()),('clf',RandomForestClassifier(n_estimators=100,n_jobs=-1))])
clf.fit(X_train,y_train)
Pipeline(steps=[('tfidf', TfidfVectorizer()),
('clf', RandomForestClassifier(n_jobs=-1))])
y_pred =clf.predict(X_test)
confusion_matrix(y_test,y_pred)
(base) pallavisumakurmala@Pallavis-MacBook-Air jangoChat %


Message Detection.ipynb
PipeLine and SVM CLASSIFIER
clf =Pipeline([('tfidf',TfidfVectorizer()),('clf',SVC(C = 1000,gamma = 'auto'))])
clf.fit(X_train,y_train)
Pipeline(steps=[('tfidf', TfidfVectorizer()),
('clf', SVC(C=1000, gamma='auto'))])
y_pred =clf.predict(X_test)
confusion_matrix(y_test,y_pred)
array([[222, 3],
[18, 206]])
print(classification_report(y_test,y_pred))
precision recall f1-score support
ham 0.93 0.99 0.95 225
spam 0.99 0.92 0.95 224
(base) pallavisumakurmala@Pallavis-MacBook-Air jangoChat %


Message Detection.ipynb
[18, 206]])
print(classification_report(y_test,y_pred))
precision recall f1-score support
ham 0.93 0.99 0.95 225
spam 0.99 0.92 0.95 224
accuracy 0.95 449
macro avg 0.96 0.95 0.95 449
weighted avg 0.96 0.95 0.95 449
accuracy_score(y_test,y_pred)
0.9532293986636972
clf.predict(['YOu are beautiful '])
array(['ham'], dtype=object)
(base) pallavisumakurmala@Pallavis-MacBook-Air jangoChat %


100
80
60
40
20
0
0
100
200
300
400
500
600
700
800

USE CASE DIAGRAM



FUTURE SCOPE

- Including voice messaging , calling.
- Include web search option by long pressing a message and many more.

REFERENCES

- http://navajyotijournal.org/Aug_2017_issue/Aug2017_7.pdf
- https://iopscience.iop.org/article/10.1088/1742-6596/1797/1/012017/pdf
- https://www.w3schools.com/ai/ai_machine_learning.asp

THANK YOU!