

Project Report On
“Project Centric Mail”

SUBMITTED TO SAVITRIBAI PHULE PUNE UNIVERSITY

BACHELOR OF ENGINEERING

(Computer Engineering)

By

- 1. Ayush Agarwal**
- 2. Aaditya Deshpande**
- 3. Shrikant Gadekar**
- 4. Sunny Khandare**

UNDER THE GUIDANCE OF

Prof. Bhagyashree Bhoyar



Department of Computer Engineering

Dr.D.Y.Patil Institute of Technology Pimpri, Pune-18

2018-2019



Dr.D.Y.Patil Institute of Technology Pimpri, Pune-18

Department of Computer Engineering

CERTIFICATE

This is to certify that the project entitled

“Project Centric Mail”

Submitted By

Ayush Agarwal	T150244207
Aaditya Deshpande	T150244247
Shrikant Gadekar	T150244251
Sunny Khandare	T150244298

is a bonafide work carried out by them under the supervision of **Prof. Bhagyashree Bhoyar** and it is approved for the partial fulfillment of the requirement of SavitribaiPhule Pune University for the award of the Degree of Bachelor of Engineering(Computer Engineering)

Prof. BhagyashreeBhoyar

Internal Guide

Dept. of Computer Engg.

Dr. Pramod. D. Patil

H.O.D

Dept. of Computer Engg.

Dr.S. N. Mali

Principal

Dr. D. Y. Patil Institute of Technology, Pimpri, Pune

Signature of Internal Examiner

Signature of External Examiner

Content

Sr.No	Name	Page No
1	Introduction	6
2	Objective	7
3	Theory	8
4	Requirement Analysis	9-10
5	Design Planning	11-13
6	Implementation	14-17
7	System Requirement	18
8	Testing	19-20
9	Maintenance	21
10	Advantages & Disadvantages	22
11	Future Scope	23
12	Conclusion	24
13	Reference	25

List of Figure

Sr.No	Name	Page No
1	Waterfall Model	9
2	System Architecture	11
3	Use Case Diagram	12
4	Flow Chart	13

List of Table

Sr.No	Name	Page No
1	Testing	19-20

I. Introduction

“To realize the value of one second, ask the person who just missed a bus or train.”

This project, the Project Centric Mailbox (PCM) is based on a simple observation. In business, it is not easy to end up in his mailbox. Following this, the student engineers found a solution to improve the mailbox, with great attention to the visualization part.

In this project if someone works on five projects, they will have five clusters. In a concrete use case, a person arrives on the PCM. It allows the access to its mails, and these are imported on a database SQLite, encrypted and secured. Once connected, the person arrives on a home page, where they can see the different clusters, and go to the page that interests them. The mails of the various projects are no longer mixed, which facilitates the visibility of these. The second big change is using pattern matching for identification of spamming mails.

The third big feature is classification of mails on the basis of whether it is formal or informal. It helps the individual to identify the mails which belongs to its business ,projects ,etc .For this supervise machine learning is used .

google API are used to fetch mails from gmail, for this authentication from user is required.

II. Objective

1. Create a user account and sign in
2. Finding the spam mail in the Gmail account
3. Clustering the mails according to the projects.
4. To identify the formality of mails.
5. To ease the usage of mails.

III. Theory

Django framework is used to run the server. The port used is 8000. Django templating language is used to create html page at the server.

Sqlite3 database is used which has two tables one for storing user details and second for storing project details. ORM (Object-Relational_Method) is used to create the tables in database and for retrieving the data from database . ORM uses object to create tables .

The project sorts mails on basis of spamming, formal-informal and projects of user.

In detection of spamming the concept of pattern matching is used. This pattern matching works exactly like a machine learning algorithm specifically **Support Vector Machine**. The spam's and advertisement are detected according to the training set provided, it calculates probability and a spam score if the probability increases the given mails are detected as a spam.

In Formal informal mails machine learning is used .Structured Machine learning have been used for classification. First dataset for training is provided and then on the basis of training mails are classified as formal or informal. The machine learning library used is **skit-learn**. This library is used majority when supervised learning algorithm is to be implemented.

For frontend django templating language is used , HTML and Bootstrap (CSS library) .To create a user friendly web pages.

In project-sorting mails are sorted on the basis of project names provided by the user. Usually sorting mails are difficult according to project names, as inbox is having various kinds of mails gathered. So by using simple python code sorting is done on user's inbox on the basis of project name by searching project name in subject of mail, mail's body and snippet of project.

IV. Required Analysis

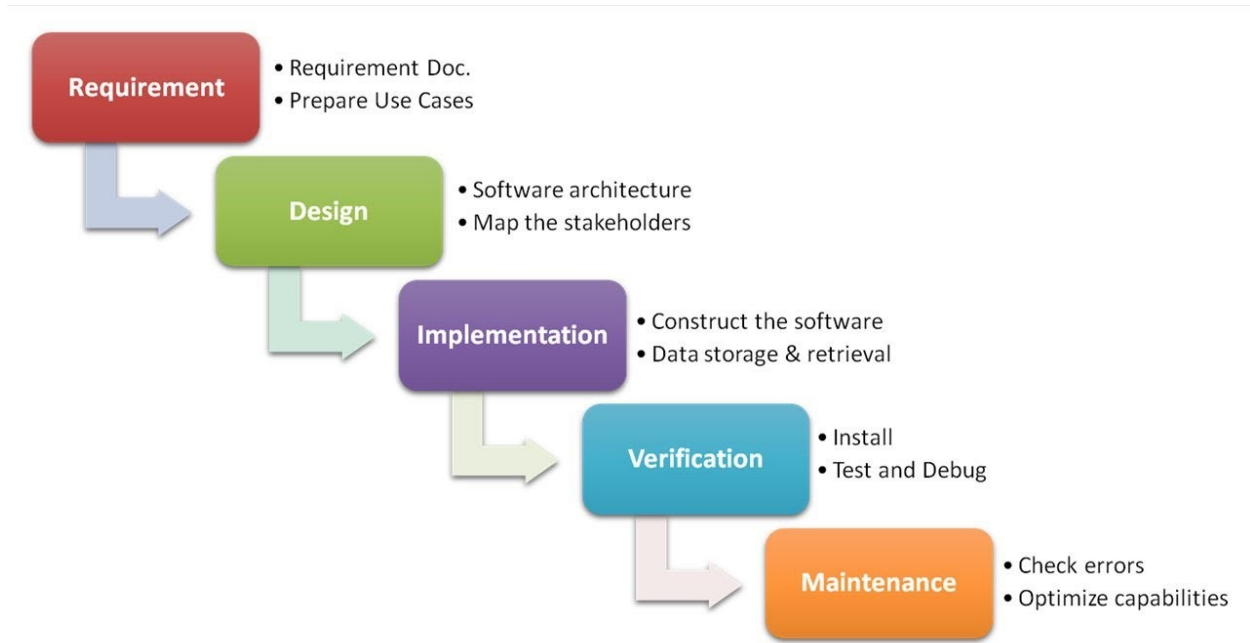


Fig.1:Waterfall Model

Login :

User should be able to login to website with their username and password

Logout:

Logged in users should be able to log out of the site

Signup:

User should be able to create a account on the basis of unique username and email id .

HomePage:

Homepage should contain basic introduction of project and a link to spamming , formal-informal and project pages.

Authentication:

Authentication with the usersgmailaccount should be done in order to gain reading access to the maintained

Spaming :

This page should show all the spamming mail received by the user

Project clustering:

Clustering(grouping) should be done on the basis of project and users should be shown mails according to the project

Formal or Informal:

Classification of mails whether formal or informal is done on basis of clustering on the received mails

V. Design Planning

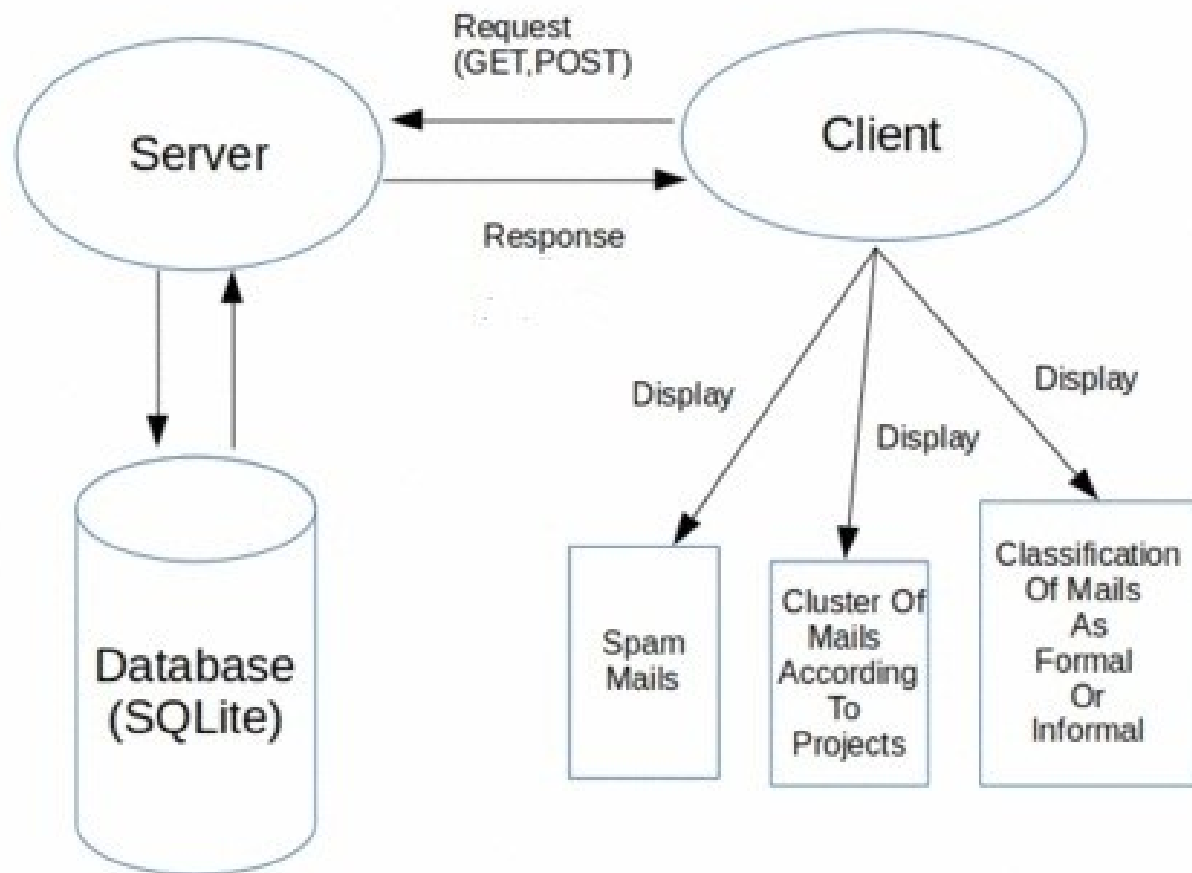


Fig.2: System Architecture

Server will host the web pages and Respond to GET and POST Request by the Client and Server will also be connected to the database . Database will save the users information and project names. On the Client side user will be displayed the mails according to Spam , Project , Formal-Informal and user can interact with the webpages .

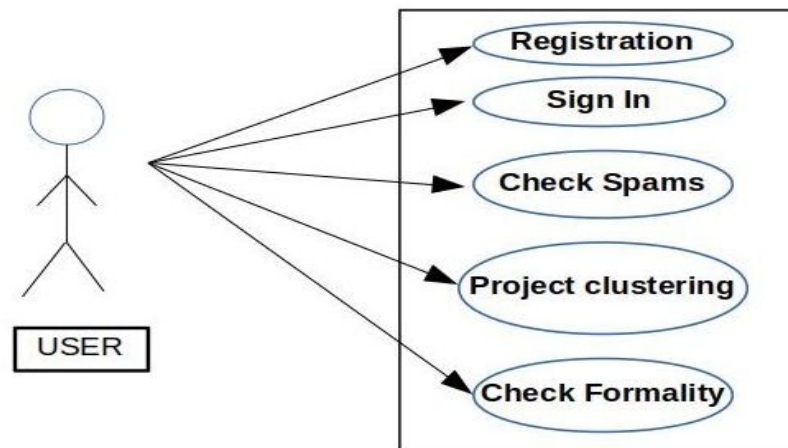


Fig.3: Use Case Diagram

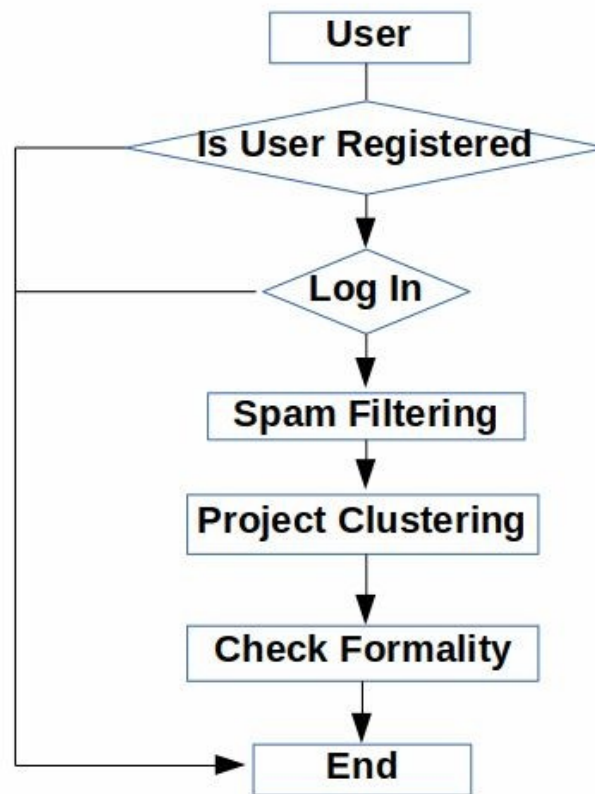
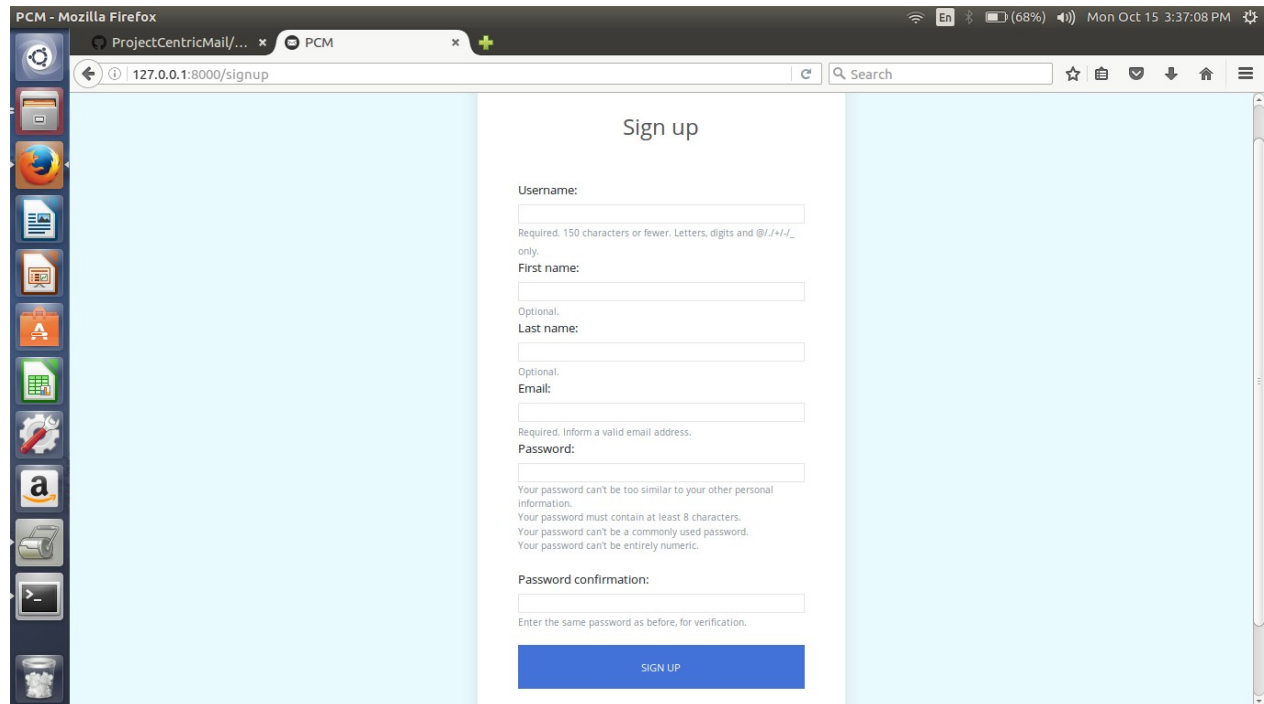


Fig.4:Flow Chart

VI. Implementation

Result of the project:

SignUp Page:

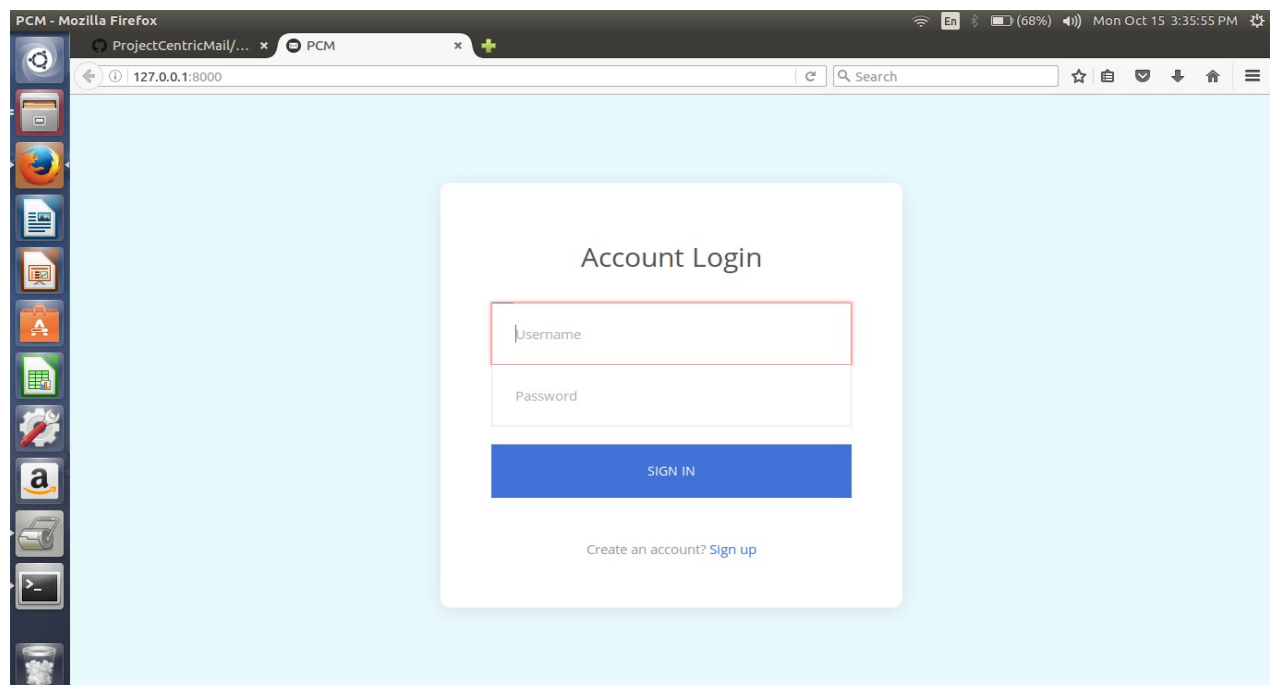


The screenshot shows a web browser window with the title "PCM - Mozilla Firefox". The address bar displays "127.0.0.1:8000/signup". The page content is titled "Sign up" and features a form with the following fields and labels:

- Username:** A text input field with a required note: "Required. 150 characters or fewer. Letters, digits and @/./+/-_ only."
- First name:** A text input field with an optional note.
- Last name:** A text input field with an optional note.
- Email:** A text input field with a required note: "Required. Inform a valid email address."
- Password:** A text input field with a required note and several password requirements: "Your password can't be too similar to your other personal information.", "Your password must contain at least 8 characters.", "Your password can't be a commonly used password.", "Your password can't be entirely numeric."
- Password confirmation:** A text input field with a required note: "Enter the same password as before, for verification."

A blue "SIGN UP" button is located at the bottom of the form.

Login Page:

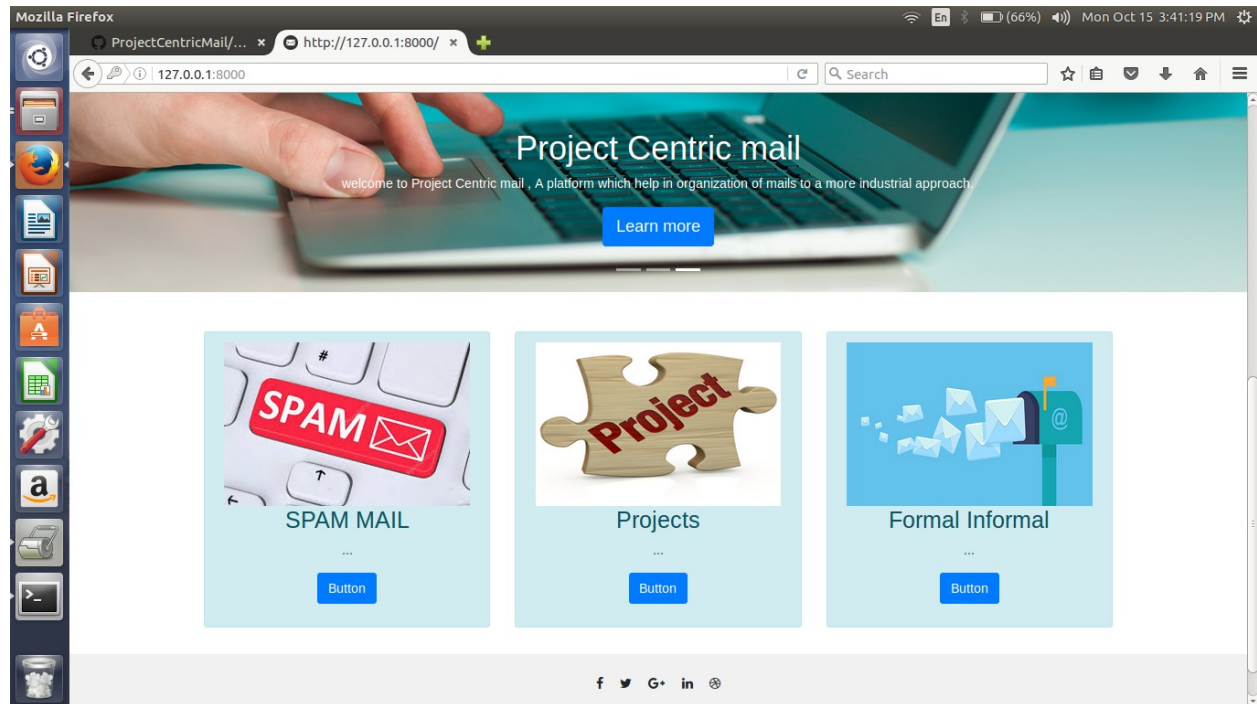


The screenshot shows a web browser window with the title "PCM - Mozilla Firefox". The address bar displays "127.0.0.1:8000". The page content is titled "Account Login" and features a form with the following fields and labels:

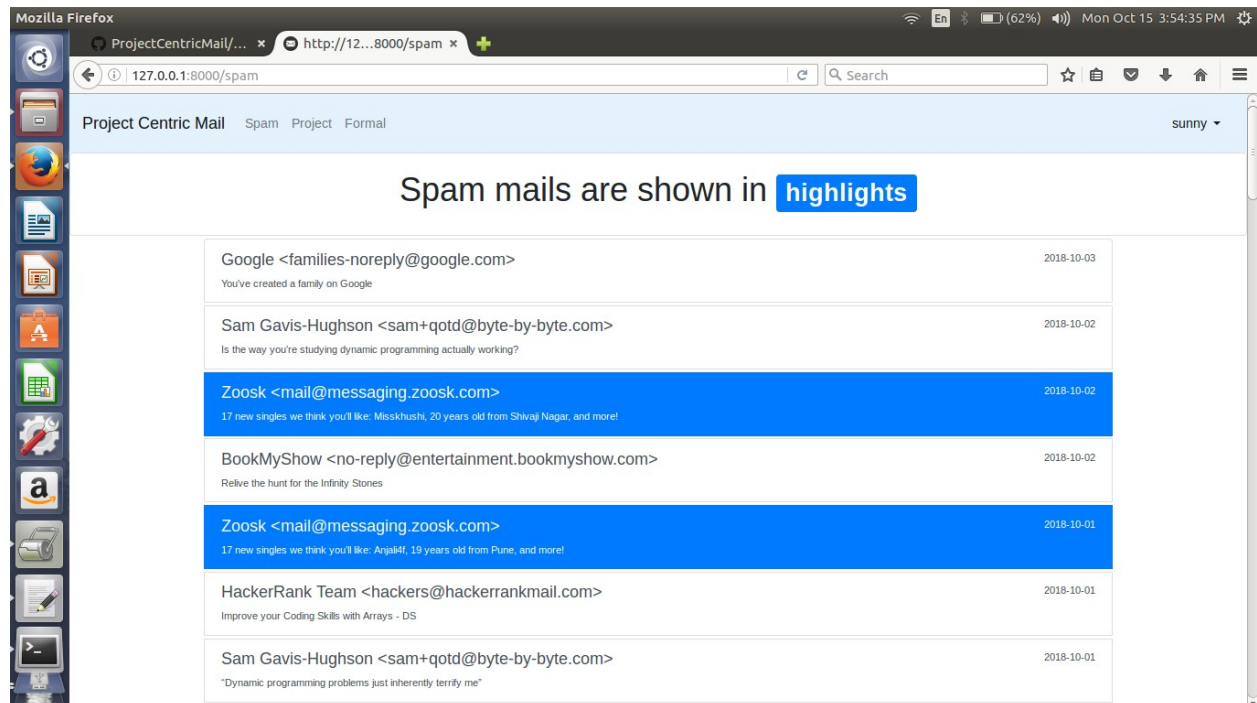
- Username:** A text input field.
- Password:** A text input field.

A blue "SIGN IN" button is located below the password field. At the bottom of the form, there is a link: "Create an account? [Sign up](#)".

Home Page :



Spam Detection:



Project-Names:

The screenshot shows a web browser window with the URL `http://127.0.0.1:8000/project`. The page title is "Project Centric Mail". The navigation bar includes "Spam", "Project", and "Formal". The user's name "ayushbansal323" is in the top right. The main content area has a blue header with the text "cn". Below this is a blue header with the text "sdl". Below that is a blue header with the text "toc". Below that is a blue header with the text "dbms". At the bottom, there is a blue header with the text "Add Project Name". Below this header is a text input field with the placeholder "Project Name". Below the input field is a blue button with the text "ADD PROJECT". At the bottom of the page, there are social media icons for Facebook, Twitter, Google+, LinkedIn, and YouTube.

Project Related Mails:

The screenshot shows a web browser window with the URL `http://127.0.0.1:8000/project/cn`. The page title is "Project Centric Mail". The navigation bar includes "Spam", "Project", and "Formal". The user's name "ayushbansal323" is in the top right. The main content area has a blue header with the text "cn related mails are shown in highlights". Below this header is a table with 7 rows. The first row is highlighted in blue. The second row is highlighted in blue. The third row is highlighted in blue. The fourth row is highlighted in blue. The fifth row is highlighted in blue. The sixth row is highlighted in blue. The seventh row is highlighted in blue.

Project Name	Date
"vivek gaikwad (via google drive)" <gaikwadvivek6398@gmail.com> dbms lab - request for access	2018-08-24
shrikant gadekar <gadekarshrikant42@gmail.com> cn server side	2018-07-24
aaditya deshpande <aadityadeshpande1998@gmail.com> fwd: cn subnet group	2018-07-18
vaibhav chavan <chavanvaibhav17@gmail.com> cn lab	2018-07-10
shweta borse <borseshweta18@gmail.com> fwd: dbms	2018-07-04
rohit agarwal <r.agal360@gmail.com> dbms programs	2018-07-03
nilesh bankar <007bankar@gmail.com> fwd: dbms 1	2018-06-28

Formal-Informal Mails:

The screenshot shows a web browser window with the URL `http://127.0.0.1:8000/Formalinformal`. The page title is "Project Centric Mail" and the user is logged in as "ayushbansal323". The main heading is "Formal mails are shown in highlights". Below this, a list of emails is displayed, each with a blue highlight bar. The emails are:

- Pirple <noreply@notify.thinkific.com> (2018-10-14)
- aditya K <adityakulkarnia3@gmail.com> (2018-10-14)
- "Change.org" <change@mail.change.org> (2018-10-13)
- "Change.org" <change@mail.change.org> (2018-10-13)
- "Yocket.in" <notifications@yocket.in> (2018-10-12)
- Shivani Jain <info@eckmarket.in> (2018-10-12)
- Ayush Agarwal <ayushbansal323@gmail.com> (2018-10-11)

Users Database :

The screenshot shows the Django administration interface for the "Users" section. The page title is "Django administration" and the user is logged in as "ayushbansal323". The main heading is "Select user to change". Below this, a search bar and a table of users are displayed. The table has columns: USERNAME, EMAIL ADDRESS, FIRST NAME, LAST NAME, and STAFF STATUS. The users listed are:

USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/> aaditya123	aadityadeshpandearmi@gmail.com	Aaditya		<input type="radio"/>
<input type="checkbox"/> ayushbansal123				<input type="radio"/>
<input type="checkbox"/> ayushbansal321	ayushbansal323@gmail.com			<input type="radio"/>
<input type="checkbox"/> ayushbansal323	ayushbansal323@gmail.com			<input checked="" type="radio"/>
<input type="checkbox"/> shrikant	1998shrikantgadekar@gmail.com	shrikant	gadekar	<input type="radio"/>
<input type="checkbox"/> sunny	sunnykhandare5618@gmail.com	sunny	khandare	<input type="radio"/>

6 users

On the right side, there is a "FILTER" section with three filters:

- By staff status: All, Yes, No
- By superuser status: All, Yes, No
- By active: All, Yes, No

VII. System Requirement

Software requirements:

1. Front end: HTML,Bootstrap
2. Back end: Python,SQL
3. Frame work: Django
4. DBMS: SQLite

Hardware requirements:

1. Processor: Intel core 7th generation
2. Hard disk: minimum 200mb
3. RAM: 512MB
4. GPU: 2GB

VIII. Testing

Sr.no	Test Scenario	Test Input	Expected Result	Actual Result	Pass/Fail
1	Log in	Username: abcd Password: Hello@123	Invalid Username	Invalid Username	Pass
2	Log in	Username: Aaditya123 Password: Hello@123	Successful Log in	Valid Username and Password	Pass
3	Gmail Login	Gmail: aadityadeshpande@gmail.com Password: *****	Invalid Password	Invalid Password	Pass
4	Gmail Login	Gmail: sunnykhandare1998@gmail.com Password: *****	Successful Log in	Authentication Successful	Pass
5	Scanning Mails	Mail's having different String format i.e. confidential mails	Mails not Scanned	As Expected	Pass
6	Spam	Mail containing Spam's	Mails Clustered	As Expected	Pass
7	Project mail clustering	Mails based on different projects of user	Cluster on basis of project	As Expected	Pass
8	Formal and	All mails having formal and informal	Cluster on	As Expected	Pass

Project Centric Mail

	informal mails	mails	basis of formal or informal		
9	Display	View of clustered mails on a webpage	All three types displayed	As Expected	Pass
10	Log out	Click on Logout	Entry in log file & logout	Successful Logout	Pass

IX. Maintenance

1. Browser needs to be updated to support HTML5
2. Python3 and its libraries to be updated regular basis
3. Django to be updated to get advance security features

X. Advantages & Disadvantages

Advantages:

1. proper grouping of mails according to project.
2. Easy to manage company, business related mails.
3. Spamming mails are identified.

Disadvantages:

1. In one iteration only 100 mails can be processed, Cost will increase to process more mail.
2. Fetching mails from gmail API are time consuming.
- 3.

XI. Future Scope

For future development, many tools could be integrated into a new version. One can imagine the calendar / planning incorporated in the project view, project management tools, a system of sentiment studies ,etc.

XII. Conclusion

Project Centric Mail can found very useful to large body communities who have to handle huge number of mails everyday and; have to filter spam mails out of it ,wants to have cluster mails according to projects and wants to handle only formals mails.Thus Project Centric Mail(PCM) can be found useful for many purposes.

XIII. Reference

1. www.esilv.fr
2. www.youtube.com
3. www.analyticsvidhya.com
4. <https://docs.djangoproject.com/en/2.1/>
5. www.github.com
6. www.stackoverflow.com
7. www.django.com
8. www.sqlite3.com