**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 Dr. Pramod. D. Patil

Internal Guide H.O.D

Dept. of Computer Engg. 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 as two tables one for storing user details and second for storing project details. ORM (Object-Relatinal\_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 usersgmailaccout should be done in order to gain reading access to the maintained

**Spaming :**

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

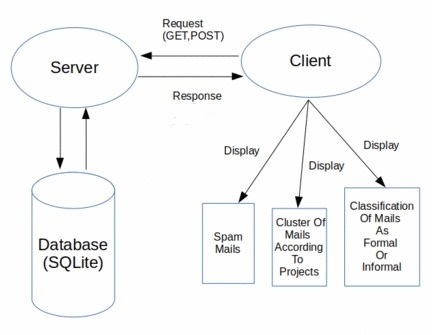
**Project clustering:**

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

**Formal or Informal:**

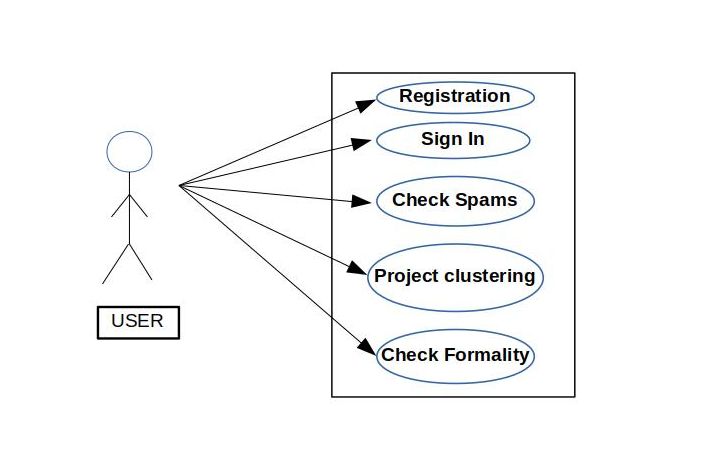
Classification of mails whether formal or informal is done on basis of clustering on the received mails

**V. Design Planning**

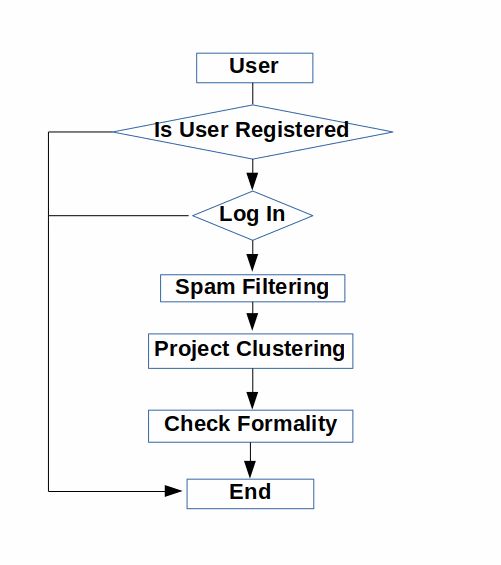


**Fig.2:System Architecture**

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**

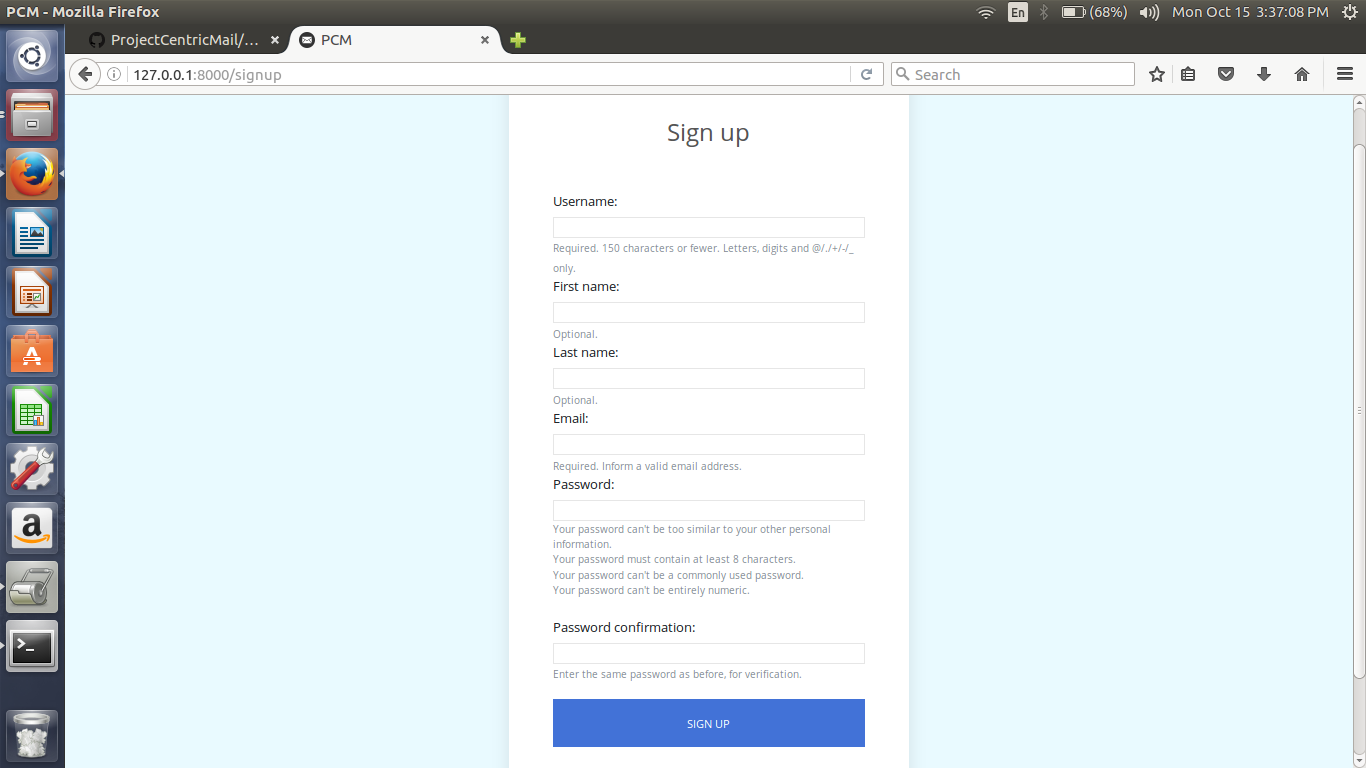


**Fig.4:Flow Chart**

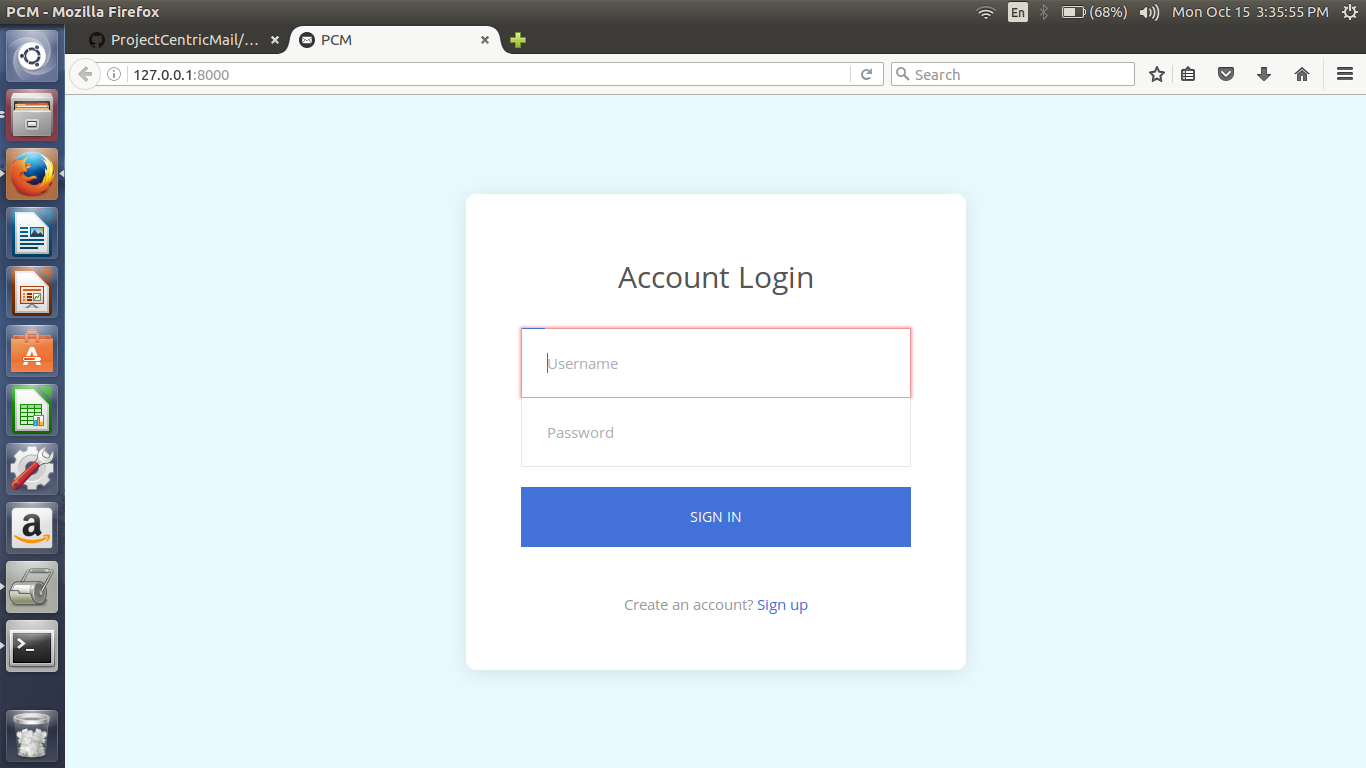
**VI. Implementation**

**Result of the project:**

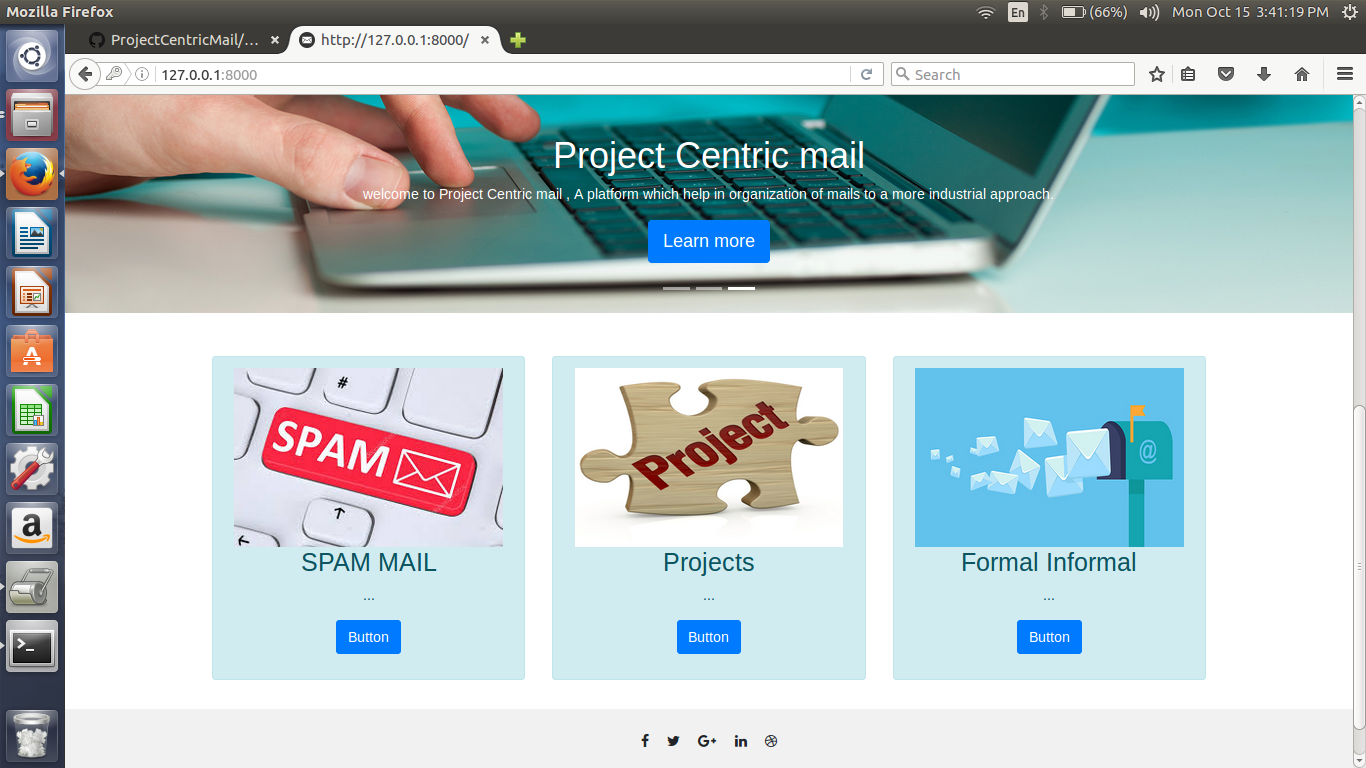
**SignUp Page:**



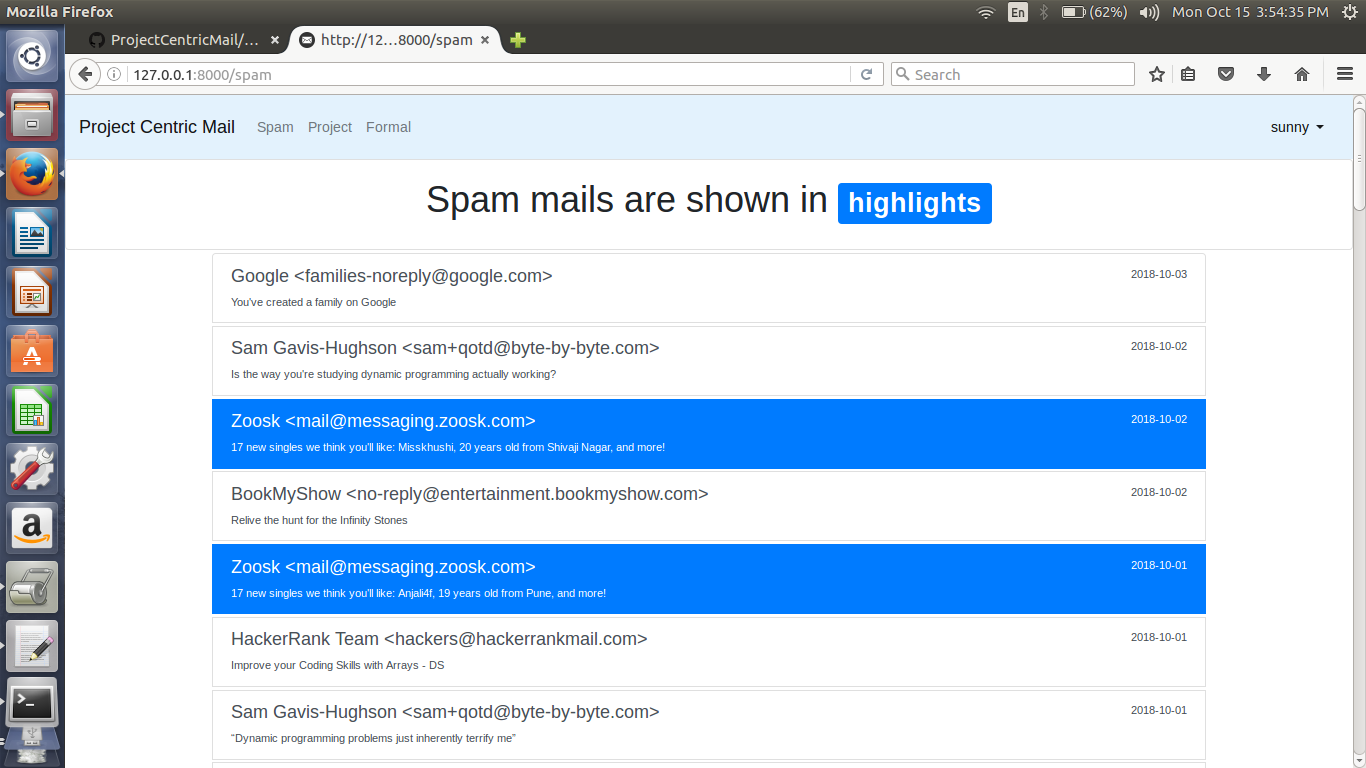
**Login Page:**

****

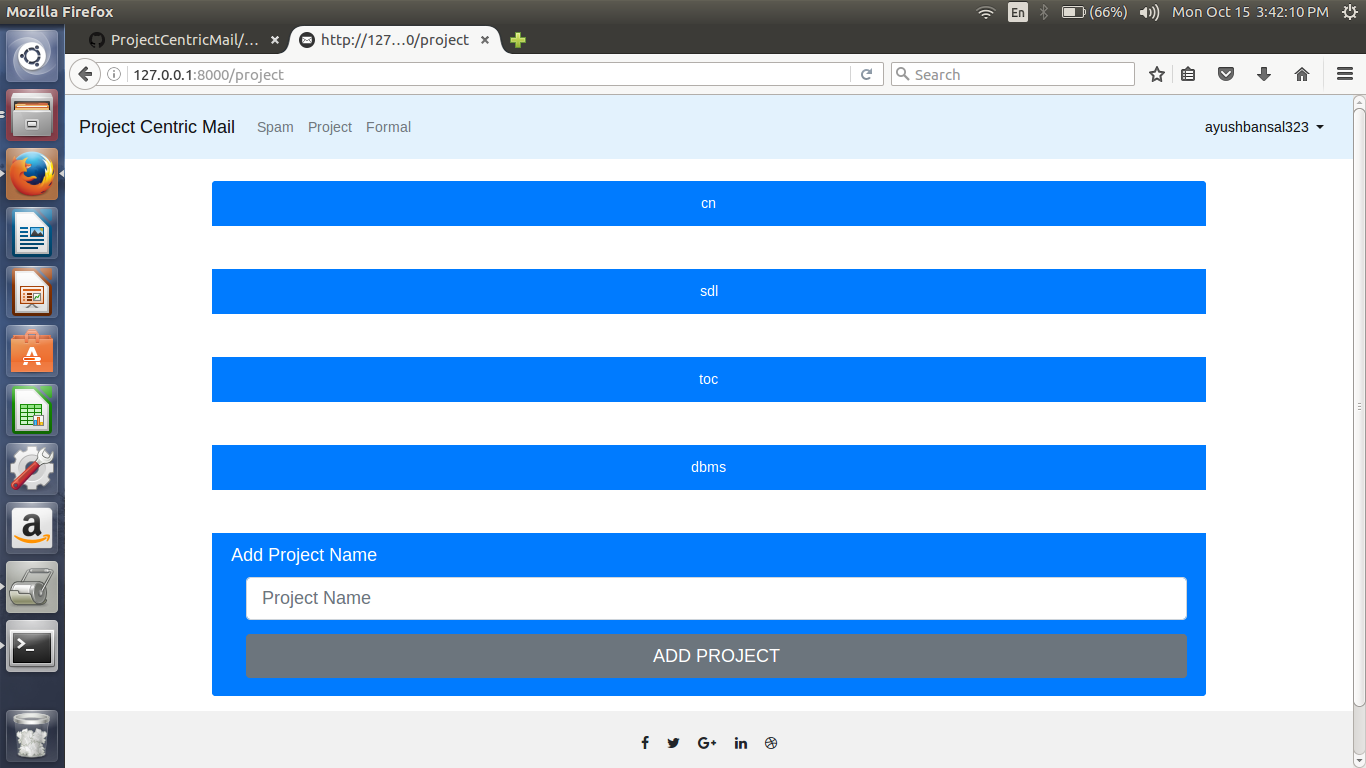
**Home Page :**

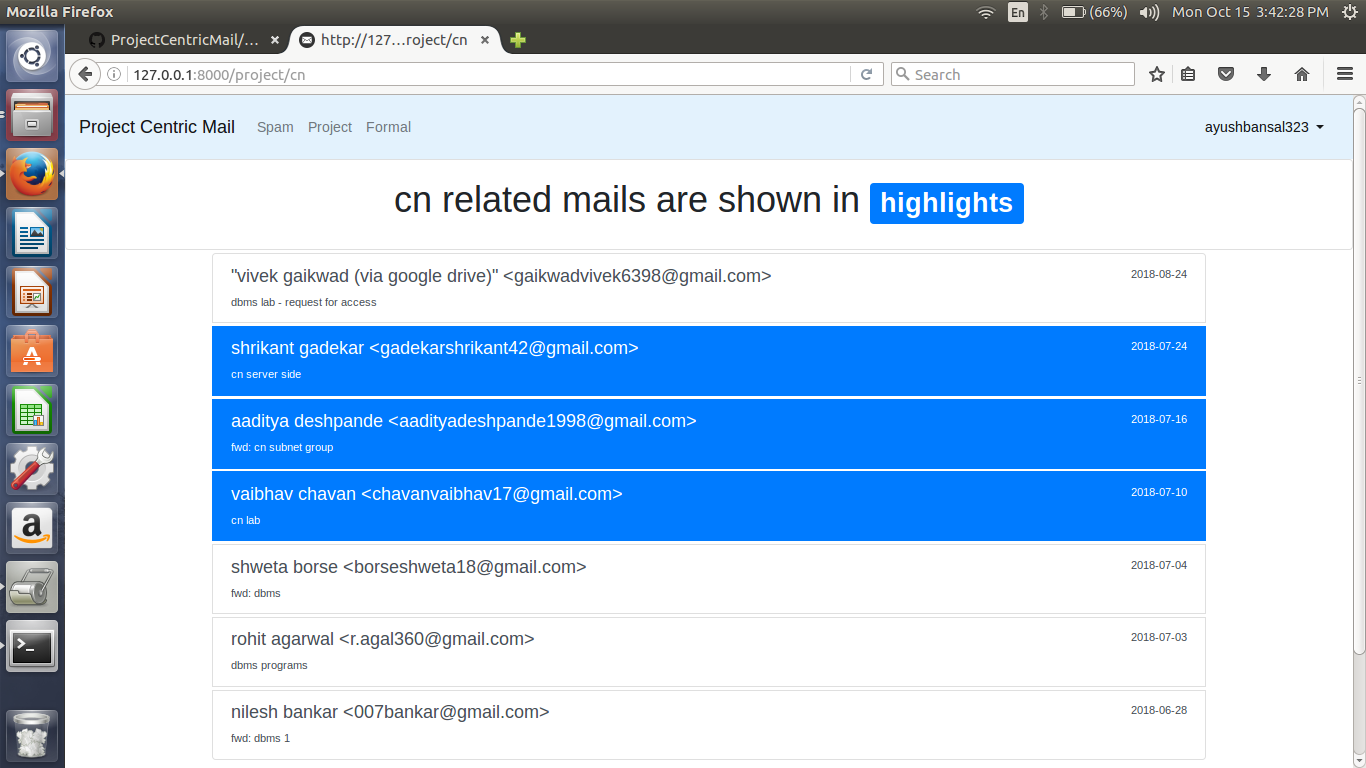


**Spam Detection:**

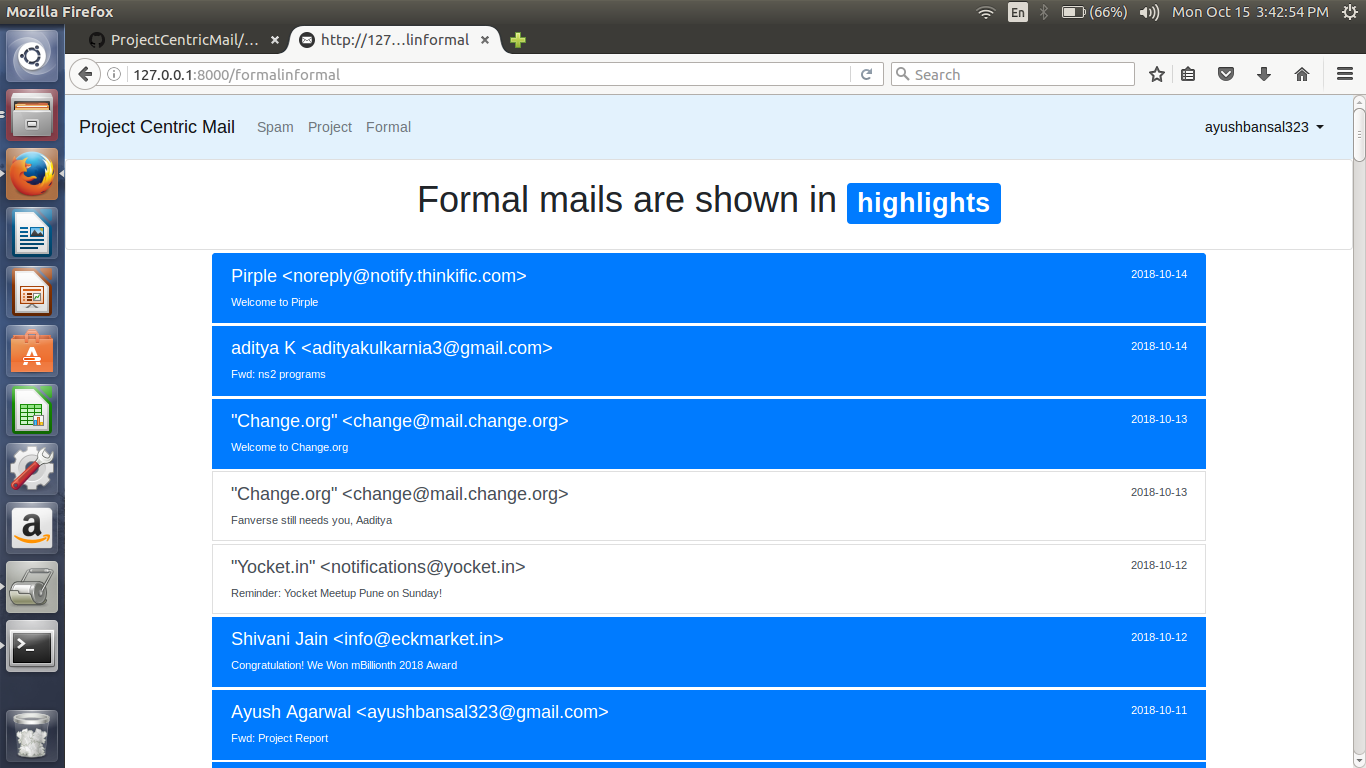


**Project-Names:**

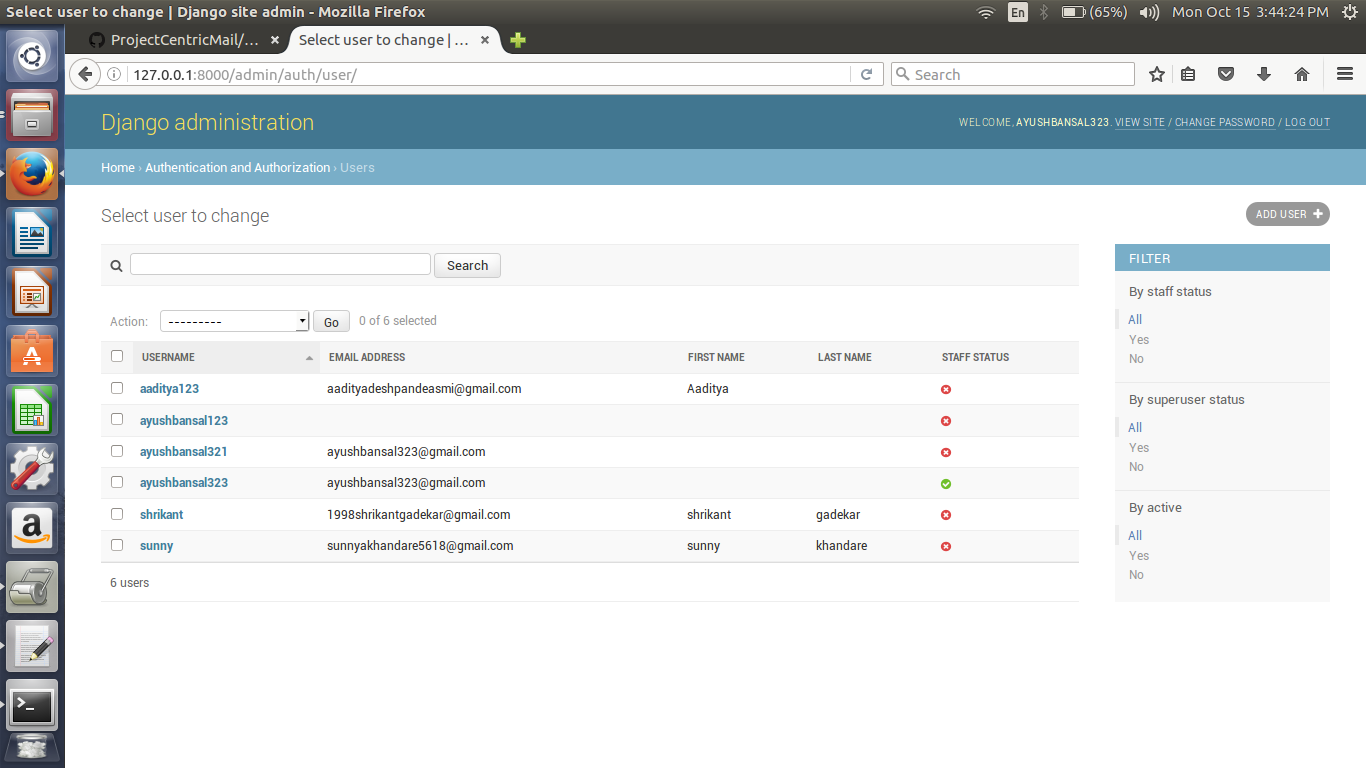


**Project Related Mails: **

**Formal-Informal Mails:**



**Users Database :**



**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](mailto:aadityadeshpande@gmail.com)  Password:  \*\*\*\*\*\*\*\* | Invalid Password | Invalid Password | Pass |
| 4 | Gmail Login | Gmail: [sunnykhandare1998@gmail.com](mailto: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 informal mails | All mails having formal and informal mails | Cluster on basis of formal or informal | As Expected | Pass |
| 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 the mange company,bussiness related mails.
3. Spaming mails are indentified.

**Disadvantages:**

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

**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](http://www.esilv.fr/)
2. [www.youtube.com](http://www.youtube.com/)
3. [www.analyticsvidhya.com](http://www.analyticsvidhya.com/)
4. <https://docs.djangoproject.com/en/2.1/>
5. [www.github.com](http://www.github.com)
6. [www.stackoverflow.com](http://www.stackoverflow.com)
7. [www.django.com](http://www.django.com)
8. [www.sqlite3.com](http://www.sqlite3.com)