A REPORT ON

COMMUNITY BUILDING PLATFORM

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

NEHA KOLEKAR	**BC-121**
AKSHITA SHINDE	**BC-140**
PRADYUMNA DESHPANDE	**BC-207**
ANUJ PADMAWAR	**BC-227**

Under The Guidance of

PROF. GEETHA CHILLARGE

DEPARTMENT OF COMPUTER ENGINEERING
Marathwada Mitra Mandal's College of Engineering
Karvenagar, Pune-411052
Savitribai Phule Pune University

2020-21

CERTIFICATE

This is to certify that the project report entitled

"COMMUNITY BUILDING PLATFORM"

Submitted by

NEHA KOLEKAR	PRN: 71826638C
AKSHITA SHINDE	PRN: 71826904H
PRADYUMNA DESHPANDE	PRN: 71826436D
ANUJ PADMAWAR	PRN:71826749E

are bonafide students of this institute and the work has been carried out by them under the supervision of **Prof. Geetha Chillarge** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering**(Computer Engineering).

Prof. Geetha Chillarge
Internal Guide,
Dept. of Computer Engg.

Prof. H. K. Khanuja
Head,
Dept. of Computer Engg.

Dr. S.M. Deshpande
Principal,
Marathwada Mitra Mandal's College of Engineering, Karvenagar, Pune – 411052

Date: 22-05-2021
Place: MMCOE

Acknowledgment

We take this to express our deep sense of gratitude towards our esteemed guide Prof. Geetha Chillarge for giving us this splendid opportunity to select and present this project and also providing facilities for successful completion.

I thank Prof. H. K. Khanuja, Head, Department of Computer Engineering, for opening the doors of the department towards the realization of the project, all the staff members, for their indispensable support, priceless suggestion and for most valuable time lent as and when required. With respect and gratitude, we would like to thank all the people, who have helped us directly or indirectly.

NEHA KOLEKAR	BC-121
AKSHITA SHINDE	BC-140
PRADYUMNA DESHPANDE	BC-207
ANUJ PADMAWAR	BC-227

Abstract

Community development is the process where community members come together to strengthen the civil society by prioritising the actions of communities, and their perspectives in the development of social, economic and environmental policy. Community development workers help communities to bring about social change and improve the quality of life in their local area. In today's world, online applications have become an important component of the industry. Businesses can now grow and become easier by using the web apps, and achieve their goals quickly. The development of a web application involves identifying product requirements, designing, coding, and testing using frameworks and technologies. Applications are built on top of frameworks so that the fundamental requirements in developing a web application are already set up.

Technical Keywords

(1) I.7.5 [MVC]: Model View Controller

(2) I.5.4 [MVT]: Model View Template

(3) I.4.0 [Framework]: An essential supporting structure

(4) E.2 [Web Server Application Development]: Dynamic and Static websites

(5) J.1 [DOM]: Document Object Model

(6) J.2 [pip]: Python Packet manager

(7) J.3 [CMS]: Content management system

Contents

List of Figures

Chapter 1

Introduction

A community building platform is an online space created by an organization or a brand, where members, customers and fans alike can congregate, ask questions, receive peer-to-peer support, discuss interests surrounding the brand and make social connections. It is essentially a home for you to house your internal communities. A central space where your audience and customers can share their ideas, tips, experiences, create bonds over mutual interests, and where you can support your community. The idea of the community building platform has garnered a fair amount of attention over the past few years, especially for providing customer experience. It has become a priority for many organizations. In fact, increasing customer engagement through improved customer experience is one of the top business priorities and a community building platform is one of the most effective ways to do this. The community is a resource for the customers. A place where they can find information and answers quickly. They can bounce ideas off of each other and work through problems. They can meet new people who share their passions and interests. People go to communities to feel accepted and nerd out over interests.

For many, communities may feel like home, consisting of a "family of invisible friends", additionally these 'friends' can be connected through gaming communities and gaming companies. Those who wish to be a part of an online community usually have to become a member via a specific site and thereby gain access to specific content or links, but today's society and economy where media life has been fully integrated into the average day; the online community has stemmed to a full-blow culture. Also, a community can act as an information system where members can post, comment on discussions, give advice or collaborate, and include medical advice or specific health care research as well.

1.1 Motivation

Community development will lead people to become more responsible, develop healthy lifestyles, empower, and provide economic opportunities. Community work takes place in particular geographical areas, focusing on identifying their needs, issues and strategies. It can also be concentrated on a particular area in identifying the problems of the human beings or region. This application is useful in various areas like alumni, matrimonial sites, language exchange and job distribution.

1.2 Problem Definition

The website will be able to provide a platform for creating a community and will let the admin handle it throughout its life cycle. In later stages by using AI technology user insights, user classification and recommendation systems will be integrated.

Chapter 2

Literature Survey

Sr.No.	Title of paper	Year	Authors	Advantages	Technology Used	Limitations
1.	Full-stack web development using Django REST framework and React	2018	Joel Vainikka, Metropolia University of applied Sciences	The goal of this project was to create a more flexible and scalable system with RESTful API, which can operate on desktop web browsers and mobile devices.	Django, React.js, FEL API, eBot DB	The project itself did not get far enough for the possible conflicts to say if there are some structural problems in the system.
2.	Review Paper on Web Frameworks, Databases and Web Stacks	2020	Jyoti Shetty , Akshaya Kumar Joish International Research Journal of Engineering and Technology (IRJET)	The choosing of web stack should be done based on few critical factors like time needed to take the product to market, long term scalability and maintenance etc.	Django, Nodejs, angularjs, Spring Stack, MEAN/MERN, LAMP/LEMP	Critical factors such as efficiency should be considered
3.	Python Libraries and Packages for Web Development-A Survey	2019	S.L. Kavya, Dr. S. Sarathambekai PG Student, Department of IT, PSG College of Technology, Assistant Professor	This survey paper comprises various python libraries and modules that are been used in web development. Build more functions with fewer lines of code.	Django, HTML,CSS	
4.	CREATING A WEB APPLICATION WITH REACT NATIVE	2018	Siddegowda ,Gitu mani Borah, Chandru KA , Asst prof. of MCA Dep., New Horizon College of Engineering, Bangalore, Bangalore	React and other similar JavaScript libraries ease the development of snappy, event-driven user interfaces that are fast at responding to state changes.	Reactjs, HTML, Nodejs, JSX	The manual installation process has to be used when we want to use react.
5.	DJANGO THE PYTHON WEB FRAMEWORK	2018	Prof. B Nithya Ramesh, Aashay R Amballi, Vivekananda Mahanta Autonomous College, Department of Master of Computer Applications, NHCE, Bangalore, India	This article gives an overview of why to choose Django over any other framework.	Mysql, Django	

Sr.No.	Title of paper	Year	Authors	Advantages	Technology Used	Limitations
6.	Using Firebase Cloud Messaging to Control Mobile Applications	2019	Mohamed Abdalla Mokar, Sallam Osman Fageeri, Saif Eldin Fatton Faculty of cS and IT, Department of IT Alazhari University, Sudan	we get some result that is the proposed system can help the developers to send update faster to mobile application using FCM with efficient and less effort	Firebase(Google)	This approarch have some limitations one can not able to predict the correct outcomes thorugh this approarch.
7.	MANAGING USER DATA OF THE WEB APPLICATION FOR COMPUTER-BASED TESTING ACADEMIC PERFORMANCE OF THE STUDENT	2019	Y. Parfonov	The results can be applied in the development of user management subsystems for Django-based web applications with several types of users	Django , python	This does not include a high level of security.
8.	Python as a Tool for Web Server Application Development Python as a Tool for Web Server Application Development	2017	Sheetal Taneja , Pratibha R. Gupta	Web development in Python can be done using CGI or WSGI that uses python standard libraries.	Python, Web server application development, frameworks, WSGI, CGI, PHP, Ruby	
9.	A Django Based Educational Resource Sharing Website: Shreic	2020	Adamya Shyam, Nitin Mukesh	the project is developed using the proper Software Engineering process, following the Iterative Model of SDLC.	Django, HTML, CSS, SDLC, Python, Testing	

Table 2.1: Literature survey

Chapter 3

Software Requirement Specification

3.1 Introduction

Online Communities allows relationship forming between users from distinct backgrounds, resulting in a tenacious social structure. A prominent output of this structure is the generation of massive amounts of information, offering users exceptional service value proposition. However, a drawback of such information overload is sometimes evident in users' inability to find credible information of use to them at the time of need. Social media sites are already ranging from daily news and updates on critical events to entertainment, connecting with family and friends, reviews and recommendations on products/services and places, workplace management, and keeping up with the latest in fashion, to name but a few. In this Chapter we will describe the software requirements for our app.

3.1.1 Project Scope

This project is not restricted by any one particular field and can be used for promoting or development in almost any of the fields, be it social, business or economic. Hence the scope of this project is very broad and long term.
Community development will lead people to become more responsible, develop healthy lifestyles, empower, and provide economic opportunities. Community work takes place in particular geographical areas, focusing on identifying their needs, issues and strategies. It can also be concentrated on a particular area in identifying the problems of the human beings or region. This application is useful in various areas like alumni, matrimonial sites, language exchange and job distribution.

3.1.2 User Classes and Characteristics

- Credentials
  - Registration Page.
  - User Login.
  - Admin Login.
  - Super Admin Login.
- User
  - Can register to join a community.
  - Will receive opportunities through the community.
  - Can connect and communicate with other community members over the platform.
- Admin
  - Can monitor and control/filter the content on the community.
  - Maintains the rules and regulations and handles events and schedules on the community.
  - Can implement guidelines as required and put a user in review for any violations.
  - If required acts as a medium between users and sponsors/mentors.
- Super Admin
  - Verifies a user and decides whether he can join the community.

  - Accepts sponsors and mentors if deemed relevant and beneficial to the communtiy.
  - Has full authority to ban or exonerate or penalize the user as required when placed in review by an admin.
  - Assigns or removes admins as and when required.
  - Has access to everything and maintains the overall community.

3.1.3 Assumptions and Dependencies

- Every user must provide legal ID(s) and documents required at the time of registration which must be verified as valid by the super admin.
- Users must have a decent internet speed and bandwidth to view all the content and features on the website.

3.2 Functional Requirements

3.2.1 Document Verification

- The document will be verified manually when a user is registered.

3.2.2 Credential Validation

- Login process is validated for everyone (user, admin, super admin) using AI based algorithm(s).

3.2.3 Reactive User Interface

- User Interface is very reactive and attractive.
- User Interface is very user-friendly.

3.3 Nonfunctional Requirements

- To-the-point content availability
- Efficient data flow and workflow

3.3.1 Performance Requirements

- Reactive UI
- Efficiency
- Relevancy

3.3.2 Security Requirements

- Data security
- User Data verification
- User validation

3.3.3 Software Quality Attributes

- Correctness
- Reliability
- Adequacy

3.4 System Requirements

3.4.1 Database Requirements

- Data is unstructured.
- Some data may and must be available and some data won't always be available.

Thus, system must use NoSQL. For this implementation, we would use Firebase DB.

3.4.2 Software Requirements

- Django 3.1.4 Framework and Web Server for data collection, warehousing and UI.
- Firebase platform along Firebase DB setup.
- Latest versions of React JS, HTML, CSS, JS, PHP, Python installed.

3.4.3 Hardware Requirements

- Web Server Machine
  - Windows 10 64-bit X64 architecture
  - 2GB RAM or above
  - 100GB HDD
- Android 5.0 or above/ Windows 7 or above/ Linux 14.04 or above
- Internet Explorer 9 or above/ Google Chrome (Latest version recommended)/ Mozilla Firefox (Latest version recommended)
- Stable internet connection

3.5 Analysis Models: SDLC model to be applied

3.5.1 Agile Model:

The Agile model is a Software Development Model that uses a combination of both incremental and iterative models. The iterative approach is taken to build the working software after each increment. At the end of development life cycle, the final build holds all the features required.
Agile web development is a methodology based on the principles outlined in the Manifesto for Agile Software Development.The main goal of this approach is to provide flexibility and "satisfy the customer through early and continuous delivery of valuable software."
The states used for Agile SDLC will be as follows:

1. Planning
2. Requirements Analysis
3. Design
4. Coding
5. Unit Testing and
6. Acceptance Testing

3.5.2 COCOMO Model:

Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality.
The key parameters which define the quality of any software products, which are also an outcome of the Cocomo are primarily Effort Schedule:

- **Effort:**
  Amount of labor that will be required to complete a task. It is measured in person-months units.
- **Schedule:**
  Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put. It is measured in the units of time such as weeks, months.

There are three types of COCOMO Models:

1. Basic COCOMO Model
2. Intermediate COCOMO Model
3. Detailed COCOMO Model

The Six phases of detailed COCOMO are:

1. Planning and requirements
2. System design
3. Detailed design
4. Module code and test
5. Integration and test
6. Cost Constructive model

The effort is calculated as a function of program size and a set of cost drivers are given according to each phase of the software lifecycle.

Chapter 4

System Design

4.1 System Architecture

Web application architecture defines the interactions between applications, middleware systems and databases to ensure multiple applications can work together. Software architecture of a system describes its major components, their relationships, and how they interact with each other. In the proposed architecture, the website will be able to provide a platform for creating a community and will let the admin handle it throughout its life-cycle.In later stages by using AI technology user insights, user classification and recommendation systems will be integrated. The user interface will allow you to register into the system. Here we have provided users with an interactive user interface. At the time of registration, a user can give details about his interest and in which type of community he/she is interested in. depending on that the user will be grouped into groups. After verifying the account details a user can look at the things he is interested in. for example, if user has selected the alumni student then he can see job recommendations, contact details of other teachers and courses and updates about other students if he clicks on the user profile. We are proposing a system in which users can then contact the teacher for help related to jobs or projects and he/she can apply for a job. The admin is the person who has authority to verify the persons details and add him into the community.

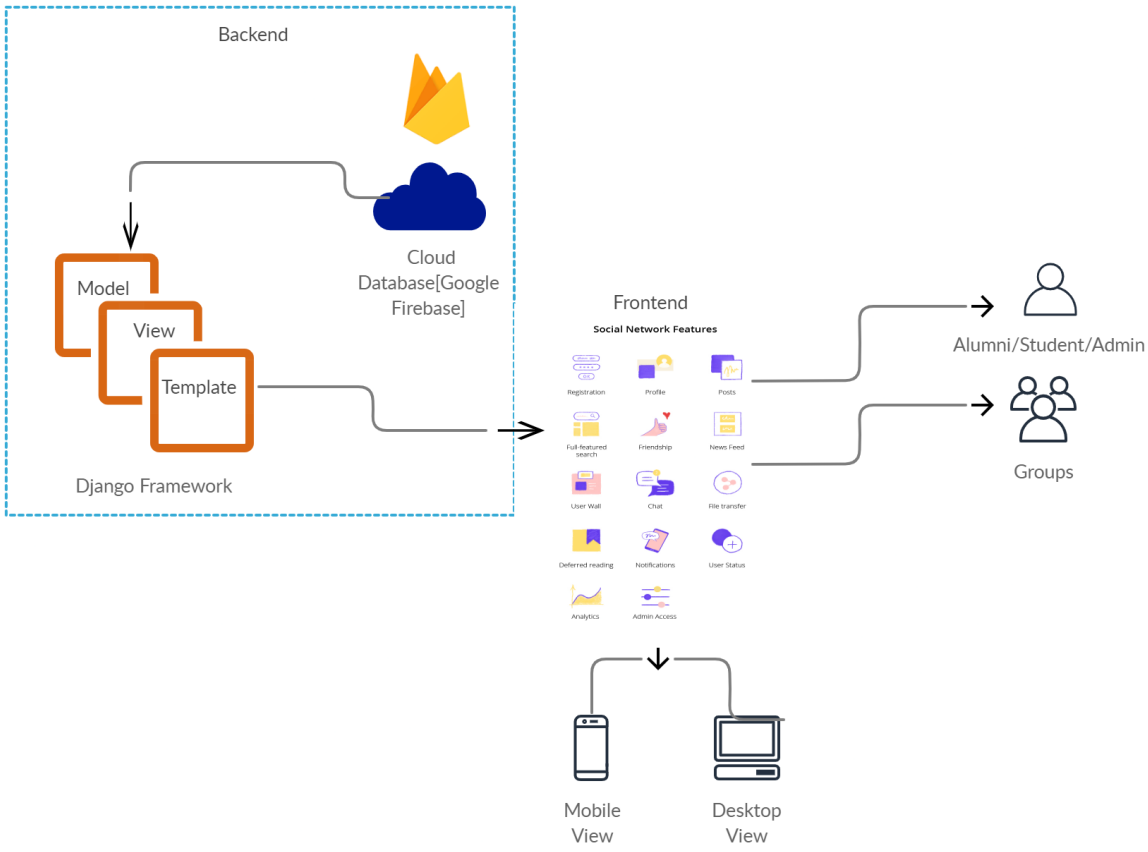


Figure 4.1: System Architecture

4.2 Entity Relationship Diagram

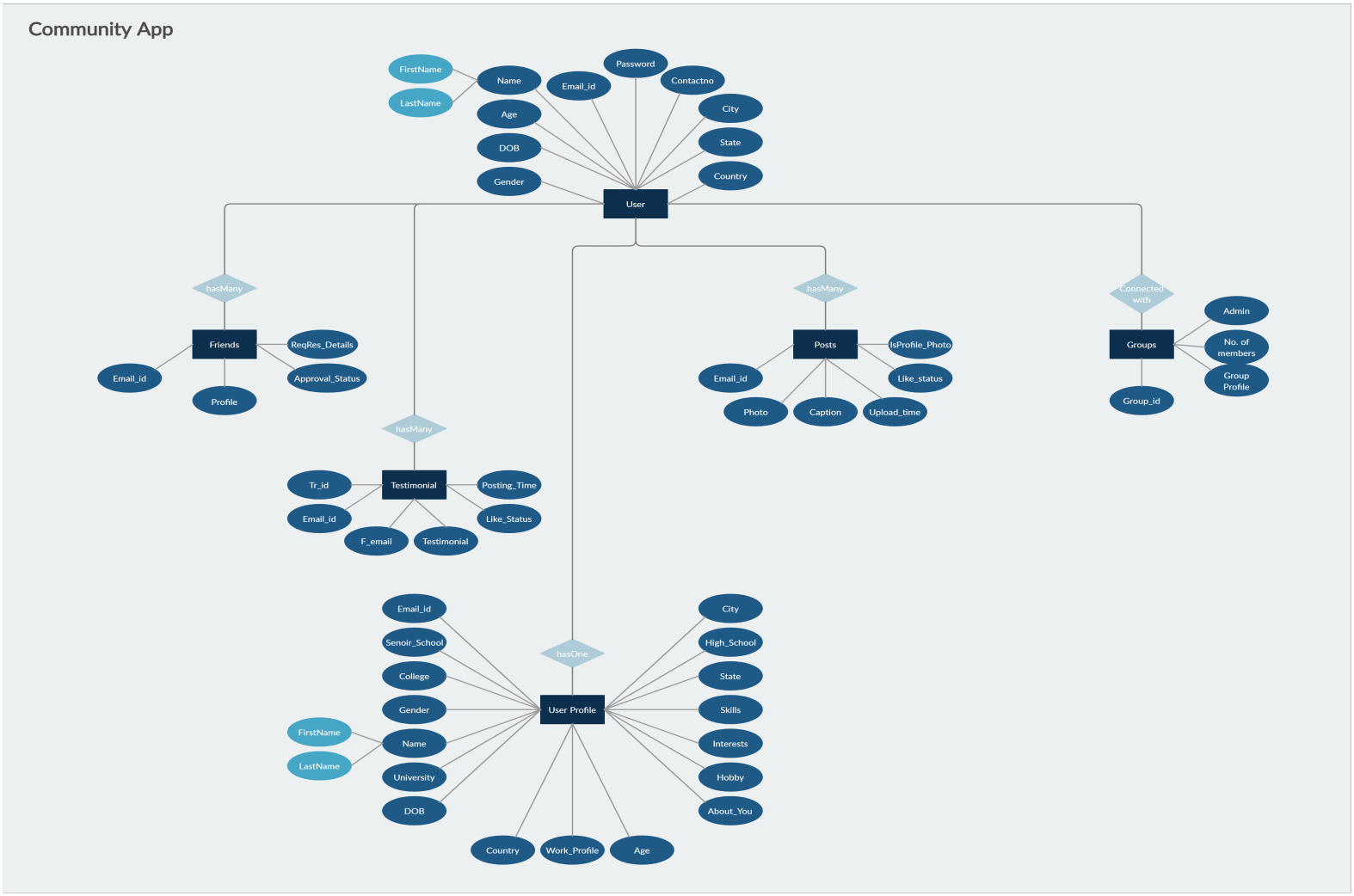


Figure 4.2: Entity Relationship Diagram

4.3 Use case Diagram

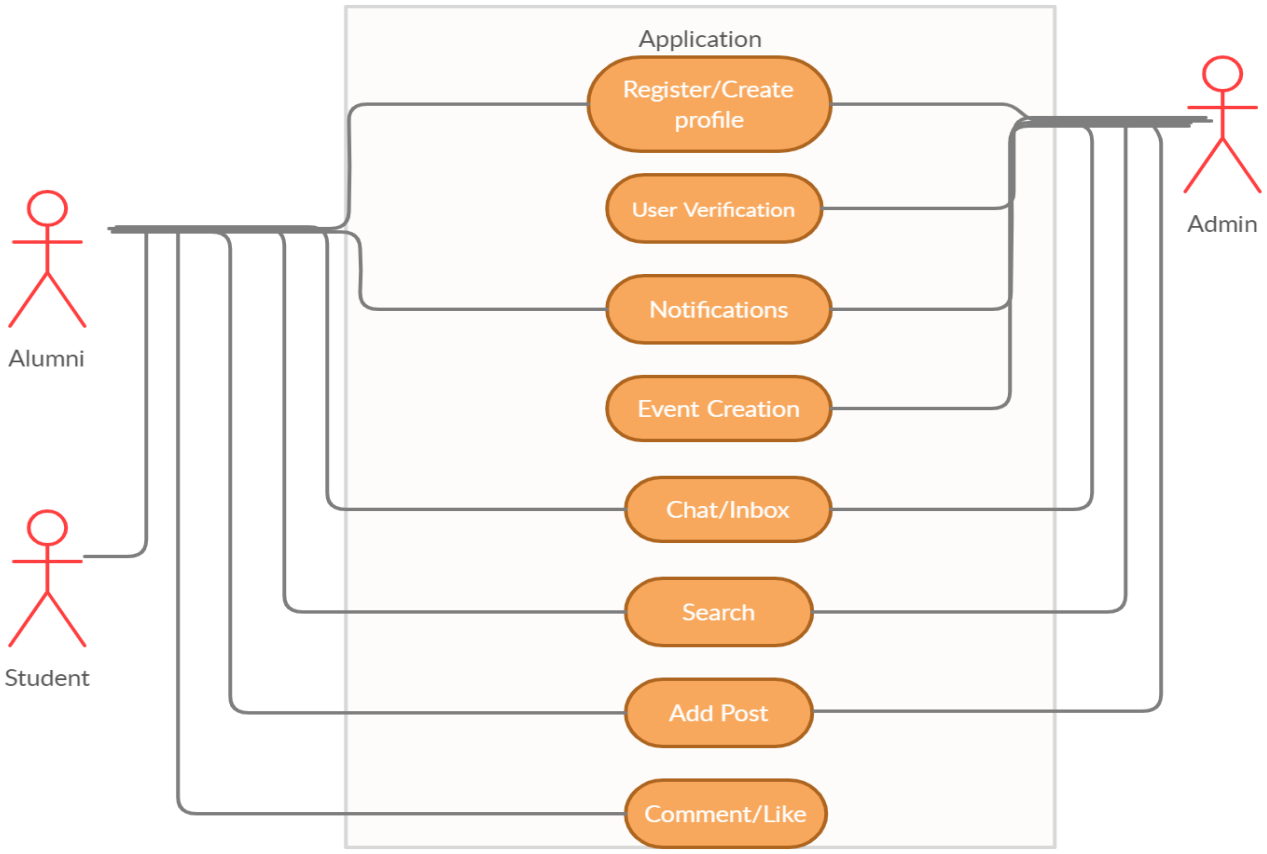


Figure 4.3: Use Case Diagram

4.4 Activity Diagram

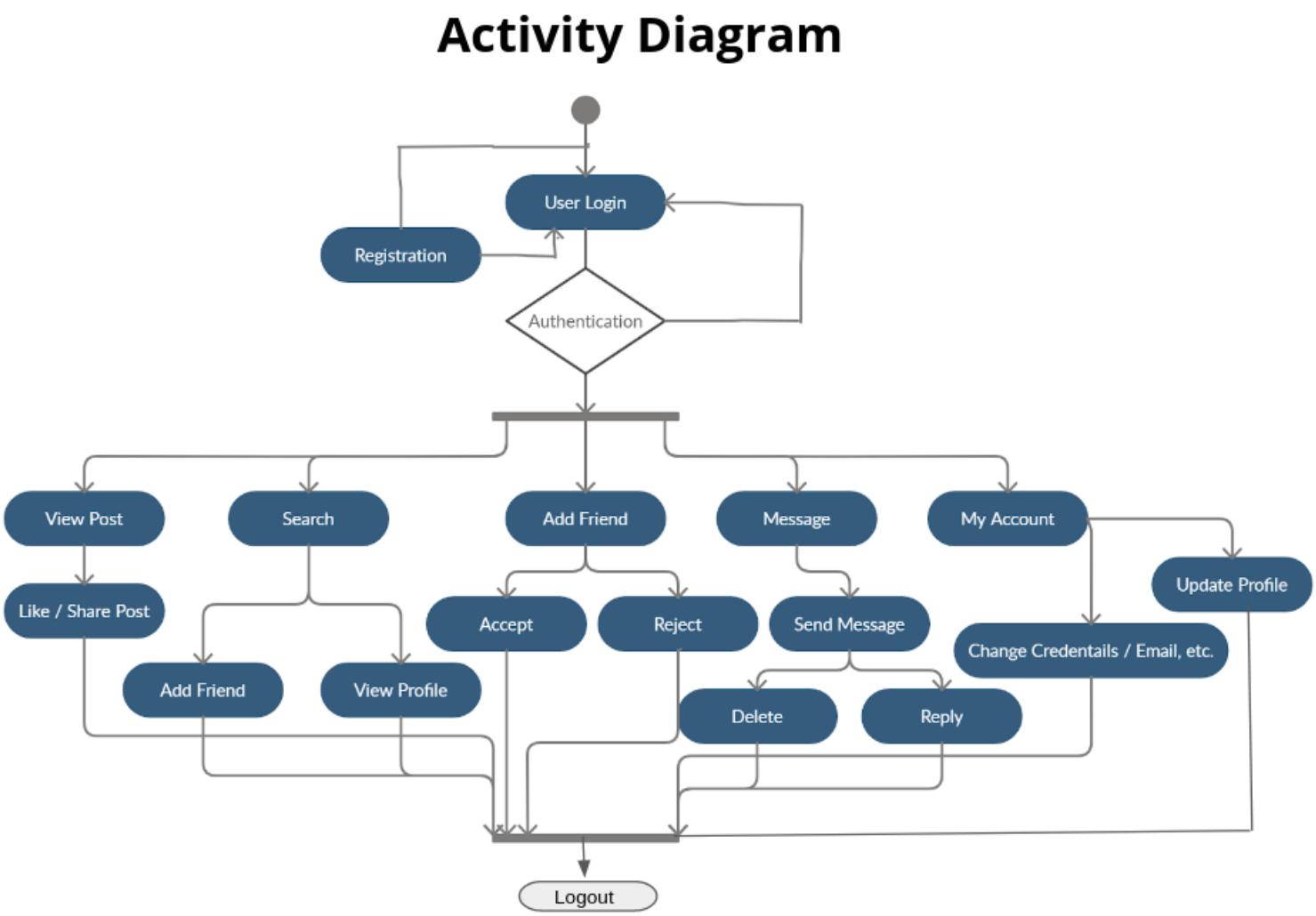


Figure 4.4: Activity Diagram

4.5 Sequence Diagram

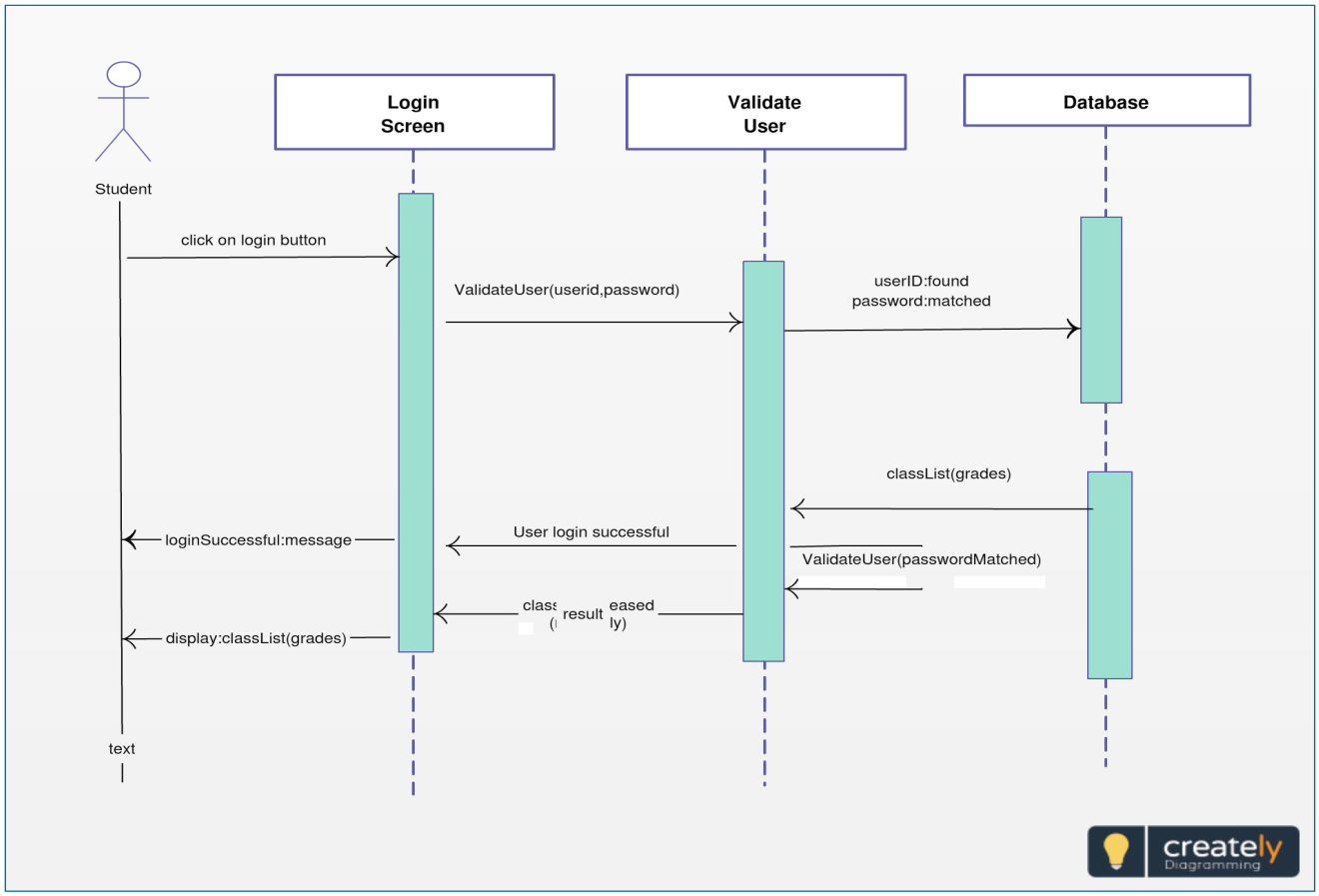


Figure 4.5: Sequence Diagram

Chapter 5

Project Plan

5.1 Project Estimate

5.1.1 Reconciled Estimates

The approximate cost structure for this project is given below.

The following assumptions are made:
1. We host this site.
2. Depending on the number of users we can dynamically increase or decrease the server.

5.1.2 Project Resources

- Stake holders: 4 Team Members, 1 Internal Guide, 1 External Guide.
- Database and Authentication: Firebase
- Development tools and frameworks: Django, Firebase
- Technologies: Python, HTML, CSS, JavaScript

5.2 Risk Management

5.2.1 Risk Identification

For risks identification project scope, requirements specifications and schedule document are reviewed. Answers to a certain questions revealed some risks. Each risk is categorized as per the categories mentioned the table in the next section.

1. Has the client formally committed to support the project?
Yes

2. Would end-users be enthusiastically committed to the project and the system/product to be built?
Yes

3. Are requirements fully understood by the software engineering team?
Yes

4. Will the system have a realtime application?
Yes

5.2.2 Risk Analysis

In software testing, risk analysis is the process of identifying risks in applications as well as prioritizing them to test. A risk is a potential for loss or damage to an organization from some threats. The key reason why we performed risk analysis during software testing is to better understand what can really go wrong with an application before it goes into production
Various risks identified for our project are :

Risk Analysis

ID	Risk Description	Likelihood of the risk occurring	Impact if the risk occurs
1.	Project purpose and need is not well-defined.	Low	Medium
2.	Project schedule is not clearly defined or understood	Low	Low
3.	Unplanned work that must be accommodated	Medium	Medium

Figure 5.1: Risk Study

Impact	Value	Description
High	> 10%	Unacceptable quality
Medium	5-10%	Some factors of the project have mediocre quality.
Low	< 5%	Barely noticeable degradation in quality. Low Impact on schedule or quality can be incorporated.

Figure 5.2: Risk Probability Definition

Probaility	Description	
High	Probability of occurrence is	> 75%
Medium	Probability of occurrence is	26 – 75%
Low	Probability of occurrence is	< 25%

Figure 5.3: Risk Impact Definition

5.3 Project Schedule

5.3.1 Project Task Set

1. Research:-
   Searching for IEEE and other publication papers related to Digital Immunization.
2. Software Requirement Specification:-
   For understanding customer requirements and analyzing objectives to be achieved.
3. Synopsis Formation
4. Designing of system architecture, gathering data for designing of models.
5. Implementation of system.
6. Testing of system by using different test cases.
7. Report.

Schedule of the project:
Start Date: 7th July 2020
End Date: May 2021
Duration: Approx 11 Months

5.3.2 Task Network

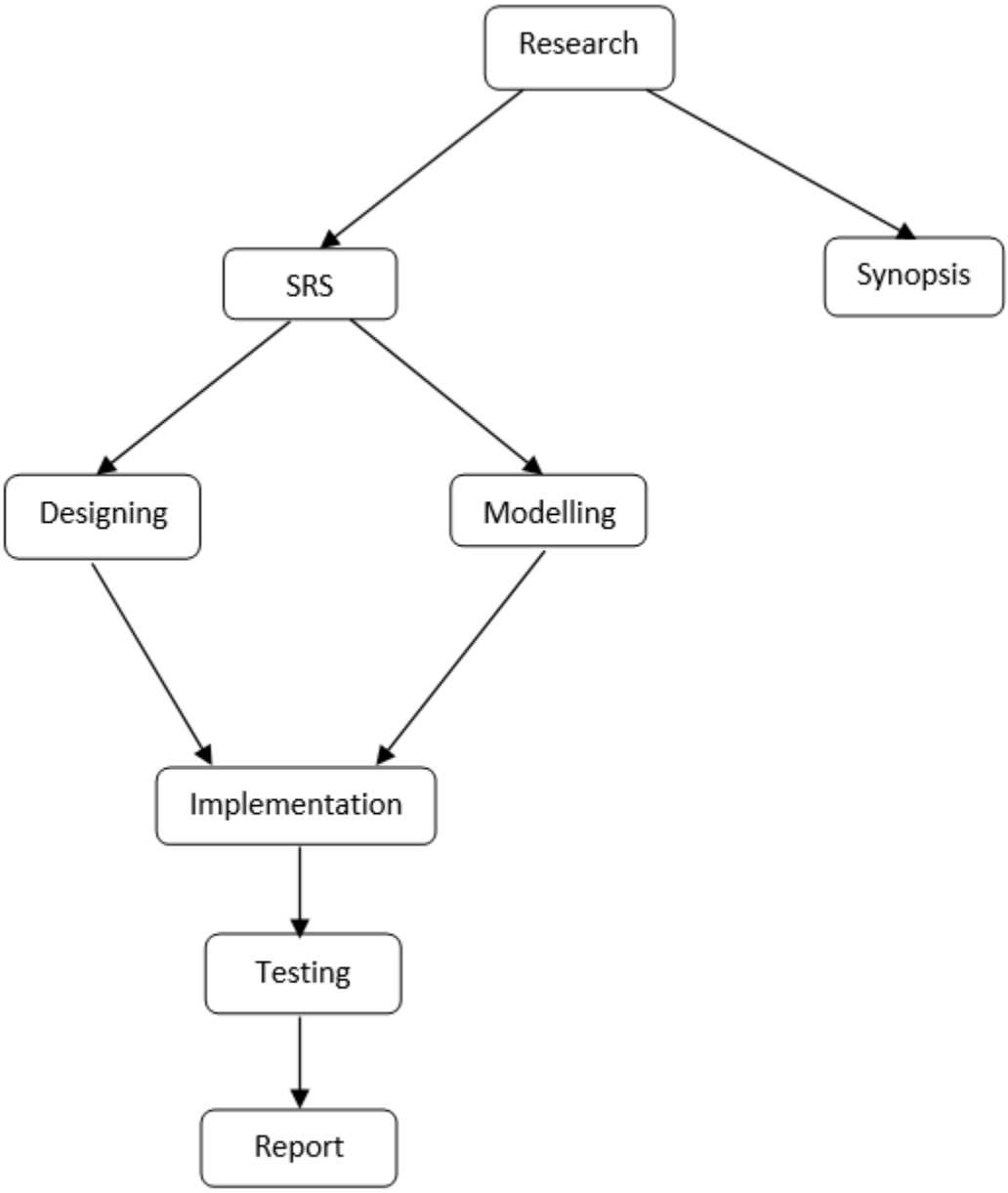


Figure 5.4: Task Network

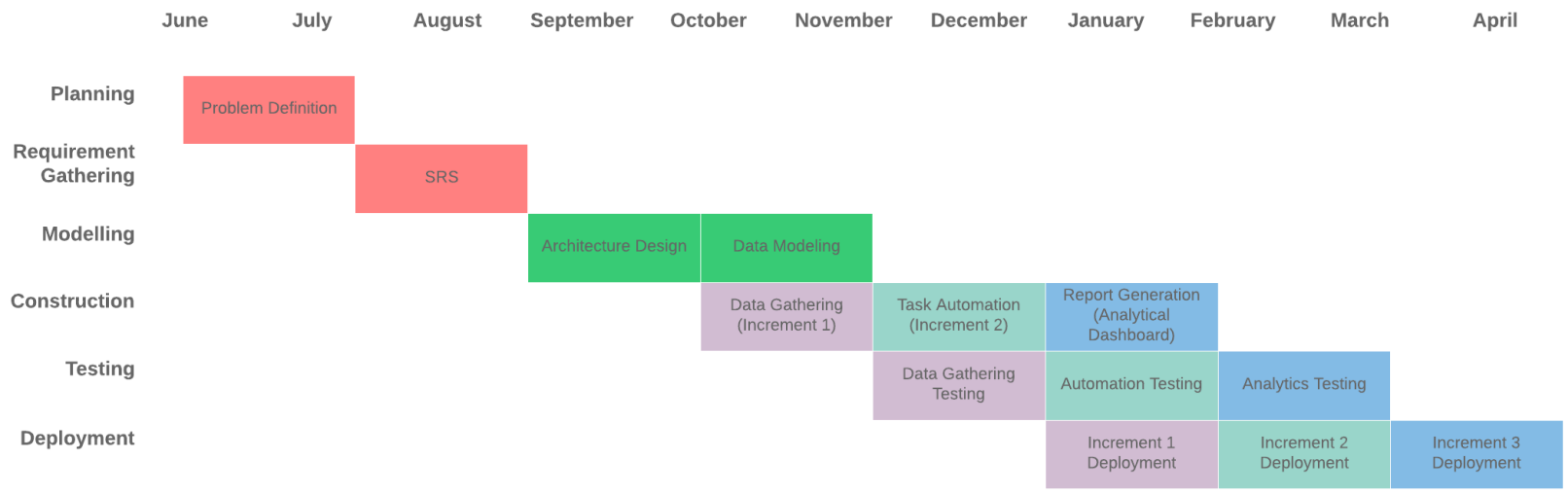


Figure 5.5: Time Line

5.3.3 Timeline Chart

5.4 Team Organization

5.4.1 Team structure

1. Neha Kolekar: Back-end, Data analysis and Documentation.
2. Akshita Shinde: Data analysis and Documentation.
3. Pradyumna Deshpande: Django, Front-end and Documentation.
4. Anuj Padmawar: Back-end and Documentation.

5.4.2 Management reporting and Communication

1. Weekly reporting to internal guide.
2. Continuous updating and reviewing of SRS and Development Processes.
3. Monthly reporting to external guide.
4. Expert guidance every month.

Chapter 6

Project Implementation

6.1 Overview of Project Modules

There are 4 major modules in this project namely,

1. Django Application
   Django is a Python-based free and open-source web framework that follows the model-template-views architectural pattern. A Django application is basically a Python package that is specifically intended for use in a Django projects. An application may use common Django conventions, such as having models ,tests ,urls ,and views submodules. With Django, you can take Web applications from concept to launch in a matter of hours. Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel.

2. Authentication
   Authentication is the act of proving an assertion, such as the identity of a computer system user. Here, Django comes with a user authentication system. It handles user accounts, Communities based platforms , and permission sessions. Briefly, authentication verifies whether a user is who they claim to be. Requests for protected resources by unauthenticated users always result in an authentication challenge.

3. Dahsboard
   A dashboard is a type of graphical user interface which presents various graphs and charts in the form of a report. It is the part where user will able the request the admin to post some data. He/She will be able make request to admin and if the admin allow it to post, all the users of community will be able to see that post. Post can be related to jobs, placements,

aptitude tests, coding competitions so in such way the community can interact with each others. User will be able to see other peoples post as well as his own.

4. User Profile

   A user profile is the module where basically the user can look into the user data and details. User data is warehoused into the firebase database from where it can be accessed for authentication and other required modules. In the user profile section the user will be able to see his data like name, IDs, profile picture, city, college, graduation year, Date of birth. Along with this user can also update his data such as name, surname, contact number. In the user profile the user can see all his posts he ever added into the community which can be helpful for further use and easy access to them.

6.2 Tools and Technologies Used

- OS: Linux
- Tools: Visual Studio Code, Firebase, Django application Server
- Programming Language Used: Python
- Scripting Language Used: Hypertext Markup Language, JavaScript, Cascading Style Sheets, Bootstrap-4

6.3 Algorithm Details

6.3.1 System Algorithm

1. Registration of User into his community.
2. After successful registration of user, he/she can directly login with the given credentials after getting profile approval from Super Admin.
3. User profile is generated.
4. On dashboard, user will be able to see the posts shared by admin.
5. In his profile section user will be able to see his data along with that, he will be able to update that data.
6. In the dashboard itself, user can make an request to admin to post specific data in the community feed. This posted data will be stored for further verification for the admin. Also the user can look for other users via the search bar provided in the dashboard section.
7. In admin profile, he will be able to see post submitted by all users and it's content. He will be able to approve whether to post the content sent by the user or not.
8. Admin can also see all the members of this particular community.
9. Based on the vaccine and user data, analysis is done.

6.4 Back-end Database-Firebase Screenshots

A Firebase project is the top-level entity for Firebase. In a project, you create Firebase apps by registering your iOS, Android, or web apps. After you register your apps with Firebase, you can add the Firebase SDKs for any number of Firebase products, like Analytics, Cloud Firestore, Performance Monitoring, or Remote Config. When you create a new Firebase project in the Firebase console, you're actually creating a Google Cloud project behind the scenes. You can think of a Google Cloud project as a virtual container for data, code, configuration, and services. A Firebase project is a Google Cloud project that has additional Firebase-specific configurations and services. You can even create a Google Cloud project first, then add Firebase to the project later.

Since a Firebase project is a Google Cloud project:

1. Projects that appear in the Firebase console also appear in the Google Cloud Console and Google APIs console.
2. Billing and permissions for projects are shared across Firebase and Google Cloud.
3. Unique identifiers for a project (like the project number and the project ID) are shared across Firebase and Google Cloud.
4. You can use products and APIs from both Firebase and Google Cloud in a project.
5. Deleting a project deletes it across Firebase and Google Cloud.

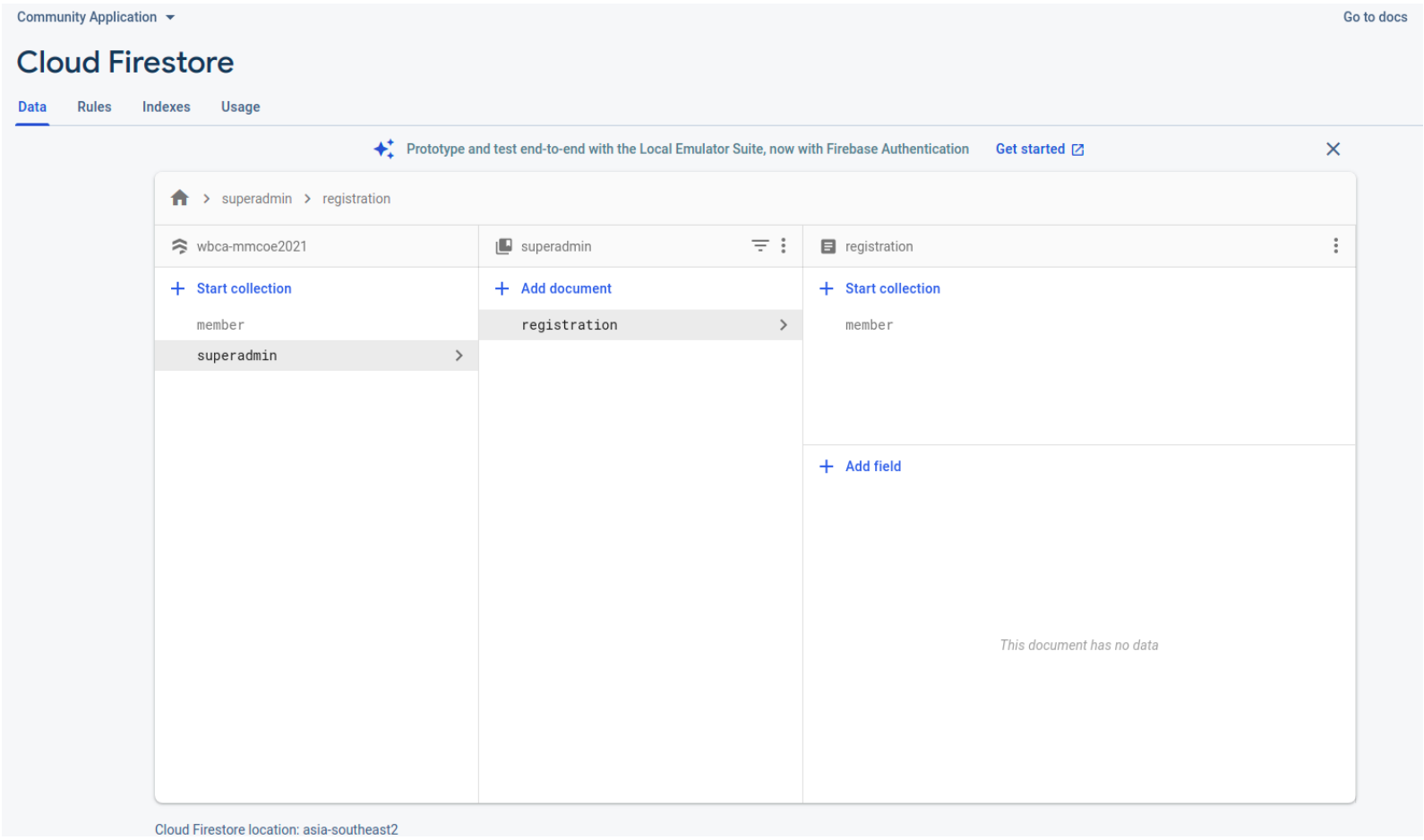


Figure 6.1: RESULT: Admin Hierarchy

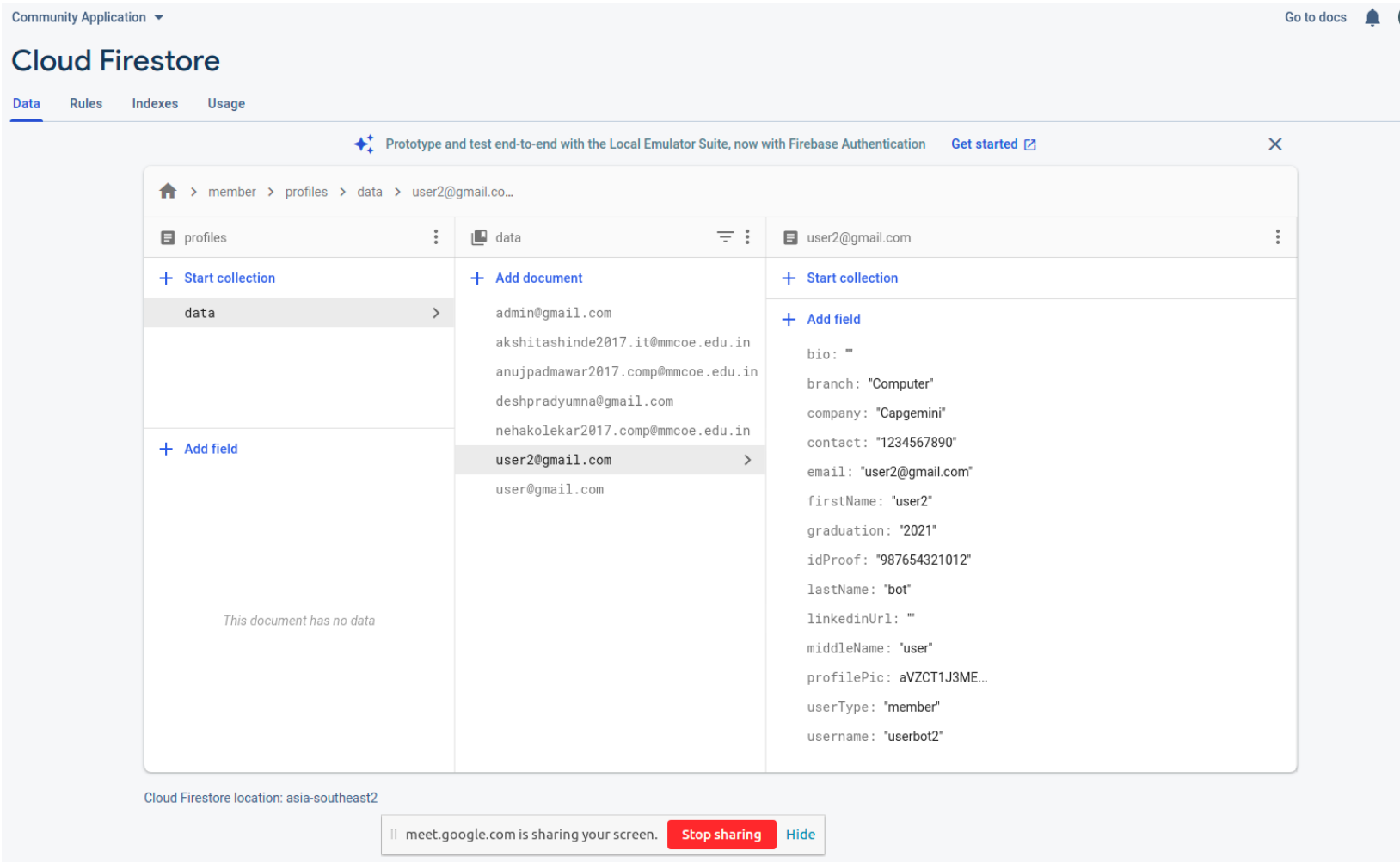


Figure 6.2: RESULT: Member Fields

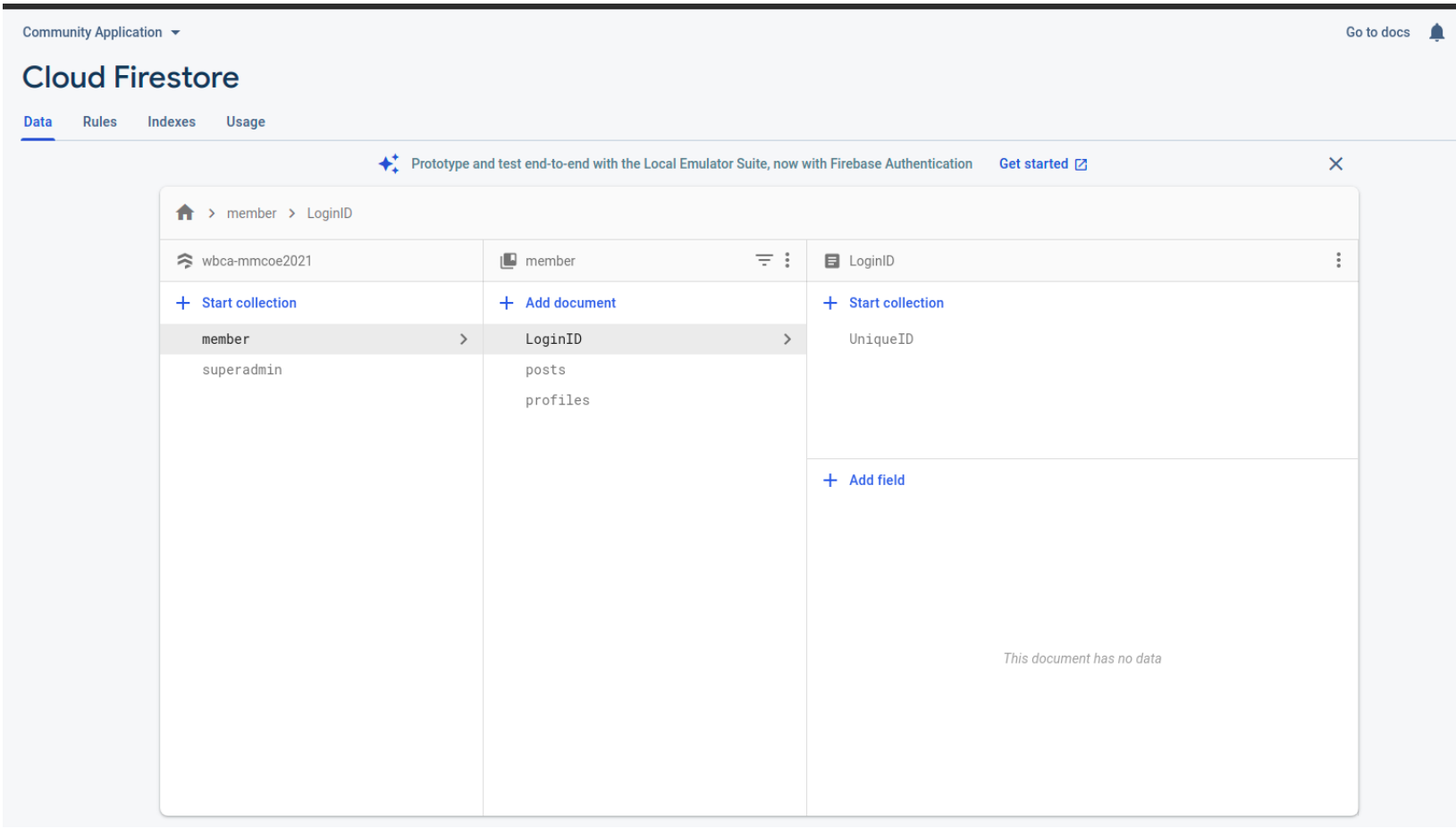


Figure 6.3: RESULT: Member Hierarchy

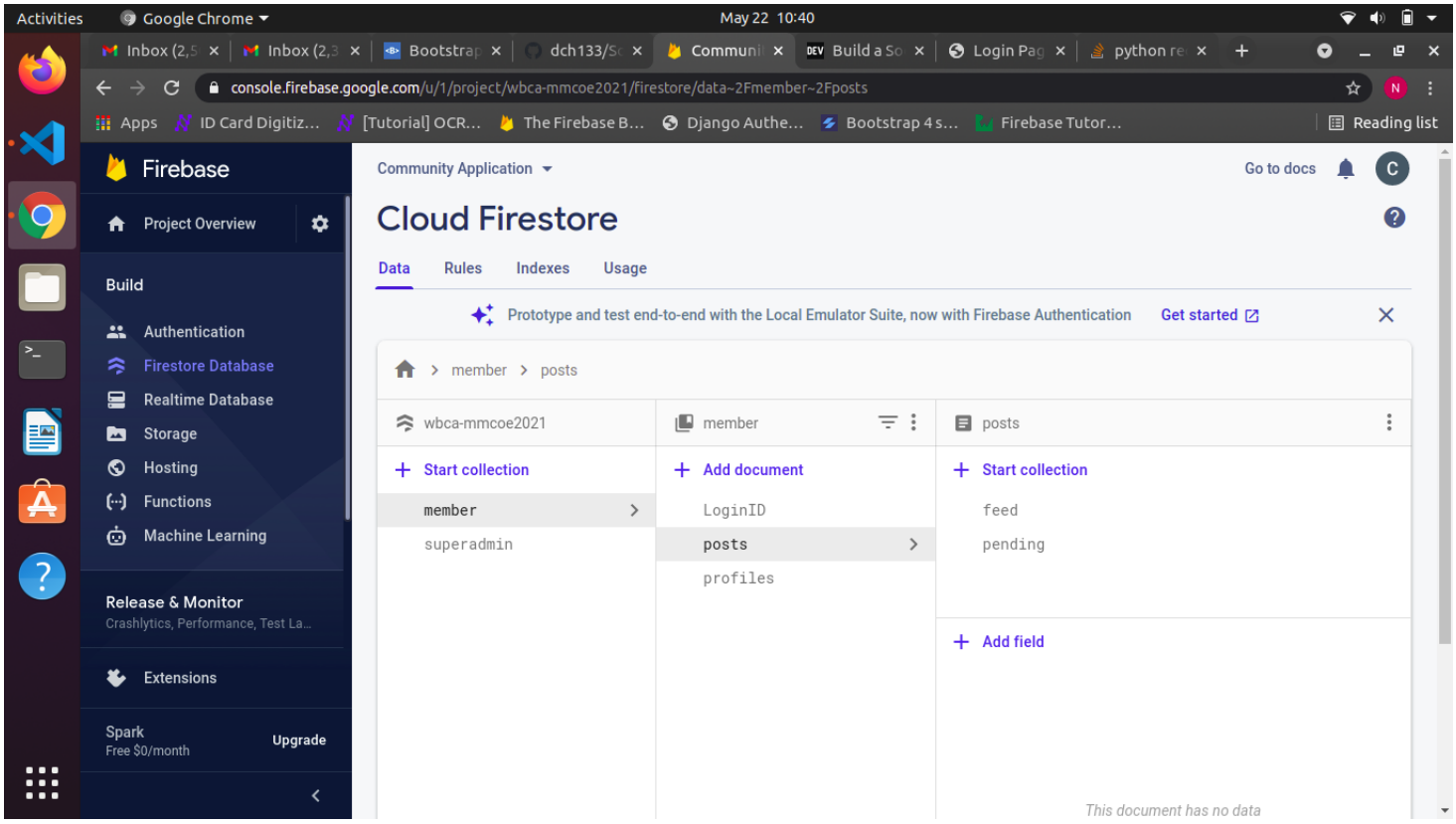


Figure 6.4: RESULT: Post Hierarchy

Chapter 7

Software Testing

7.1 Type of testing

Software testing an important activity carried out to check whether the actual results match the expected results and to ensure that the software system is defect free. It involves execution of a software component or system component to evaluate one or more properties of interest.
Software testing also helped us to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools.

- Manual Testing:
  It is a type of Software Testing where Testers manually execute test cases without using any automation tools. Manual Testing is the most primitive of all testing types and helps find bugs in the software system.
  Any new application must be manually tested before its testing can be automated.tem We manually tested various aspects of our project like:

  1. While registration, we manually tested whether the entered mobile number is a 10-digit number or not.
  2. For ID proof as AADHAR card, we manually tested if the entered number is a 12-digit number or not.
  3. We manually tested for updations in firebase after making additions in the code
  4. We manually tested for navigations through all pages to check whether they navigate to the desired pages or not.
  5. We manually tested for the dynamic displaying of username after he/she logs in.

6. We manually tested for the retrieval of Username and Bio as mentioned in the database at the time of registering.
7. We manually tested for viewing the same application on different devices using Bootstrap-4.
8. We manually tested and ensured if the correct password had been entered the second time after it had been entered once while registering.

Manual testing requires more effort but is necessary to check automation feasibility.Manual Testing does not require knowledge of any testing tool.

- Unit Automation Testing:
  Automated unit testing is a method of testing software. Units (small sections) of the code are rigorously checked to ensure they work correctly. The goal of automated unit testing is to demonstrate that each part of a larger software development project works as intended . It means using an automation tool to execute your test case suite. The automation software can also enter test data into the System Under Test, compare expected and actual results and generate detailed test reports.
  Using a unit test automation tool,like JUnit, PyUnit, it's possible to record this test suite and re-play it as required. Once the test suite is automated, no human intervention is required. This improved ROI of Test Automation.
  A few automated testcases are:

  1. Empty Username/Email Check
  2. Empty Password Field Check

  The goal of Automation is to reduce the number of test cases to be run manually and not to eliminate Manual Testing altogether.

In this project, we have used both manual as well as automation testing. Automation testing was performed where-ever it had scope.

Automation Testing Tools: JUnit, PyUnit

JUnit:

JUnit is a simple unit testing framework for the Java programming language to write repeatable tests. JUnit has been important in the development of test-driven development.

PyUnit:

The Python unit testing framework, sometimes referred to as “PyUnit”, is a Python language version of JUnit. It works similar to JUint.

7.2 Test Cases and Test Results

Back-End Test Cases

Test 1: Check Credential and data validation in python

- This is most basic and fundamental data validation test case in the system.
- Here, We will verify the user's data such as phone number, ID number
- We also go for checking the the password of user two times as we will be asking the user to type the password again to verify it from user side.

Test ID	Input	Expected Output	Actual Output	Status
1.	Empty Email	Invalid Field Error	Invalid Field Error	Pass
2.	Empty Password Field	Invalid Field Error	Invalid Field Error	Pass
3.	Wrong Email	Invalid Credential warning	Invalid Credential warning	Pass
4.	Wrong Password	Invalid Credential warning	Invalid Credential warning	Pass
5.	Profile Button	Shows user profile	Shows userprofile	Pass

Figure 7.1: Test Cases

File Edit Format Run Options Window Help

```
from django.shortcuts import render
from django.contrib import auth
from django.views.generic import TemplateView
from firebase_admin import credentials,firestore,db,auth
import firebase_admin
import pyrebase
from django.contrib.auth.decorators import login_required
from .models import Profile, FriendRequest
import random
from django.contrib.auth import get_user_model

#import cv2
import os,argparse
#import pytesseract ,re
from PIL import Image
import csv
#import pandas as pd
# Create your views here.

User = get_user_model()


class LoginPageView(TemplateView):
        template_name = 'registration/login.html'

config = {
        'apiKey': "AIzaSyD-THXuPuvdtXFMBvy1-PJo-ueMWu0SJ-E",
        'authDomain': "wbca-mmcoe2021.firebaseapp.com",
        'projectId': "wbca-mmcoe2021",
        'databaseURL': "https://wbca-mmcoe2021-default-rtdb.firebaseio.com",
        'storageBucket': "wbca-mmcoe2021.appspot.com",
        'messagingSenderId': "210306099976",
        'appId': "1:210306099976:web:c35649974e76992848dabd",
        'measurementId': "G-RC62WM6F3N"

}
firebase = pyrebase.initialize_app(config)
authe = firebase.auth()

cwd = os.getcwd()
cred = credentials.Certificate(os.path.join(cwd, 'wbca-mmcoe2021-firebase-adminsdk-4mnlv-e359f7b1aa.json'))
firebase_admin.initialize_app(cred, {'databaseURL': "https://wbca-mmcoe2021-default-rtdb.firebaseio.com"})
fs = firestore.client()
```

Figure 7.2: Credential

```
# Create your views here

global res
res = {}
def loginAuth(request):
        email=request.POST.get('email')
        passw = request.POST.get("pass")
        userType = request.POST.get("usertype")
        print(userType)
        if not fs.collection(u'{}'.format(userType)).document(u'LoginID').collection(u'UniqueID').document(u'{}'.format(email)).get().exists:
                msg = "Please enter valid credentials"
                return render(request,"registration/login.html",{"messg":msg})
        try:
                user = authe.sign_in_with_email_and_password(email,passw)
        except:
                message="invalid credentials"
                return render(request,"registration/login.html",{"messg":message})

        session_id=user['idToken']
        request.session['uid']=str(session_id)
        idtoken=request.session.get('uid',None)
        print(idtoken)
        print(user['idToken'])
#       auth.get_account_info(user['idToken'])

        currentuser = authe.current_user

        email = currentuser['email']
        print(currentuser['email'])
        print(email)
        doc_ref = fs.collection(u'member').document(u'profiles').collection(u'data')
        doc = doc_ref.document(email)
        print(doc)
        res = doc.get().to_dict()
        print(res)
        #user_data = res.copy()

        res = get_user_data()
        return render(request, "dashboard.html",res)
```

Figure 7.3: Credential validation

Chapter 8

Results

8.1 Source Code

```
#FIREBASE_PATH = os.path.join(BASE_DIR, 'wbca-mmcoe2021-firebase-adminsdk-

#AUTH_USER_MODEL = 'firebase_auth.User'


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
        'loginpage',
    'userprofile',
    'groups',

]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
        #'firebase_auth.authentication.FirebaseAuthMiddleware'
]

ROOT_URLCONF = 'notifire.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'notifire.wsgi.application'
```

Figure 8.1: Source Code

8.2 Screenshots

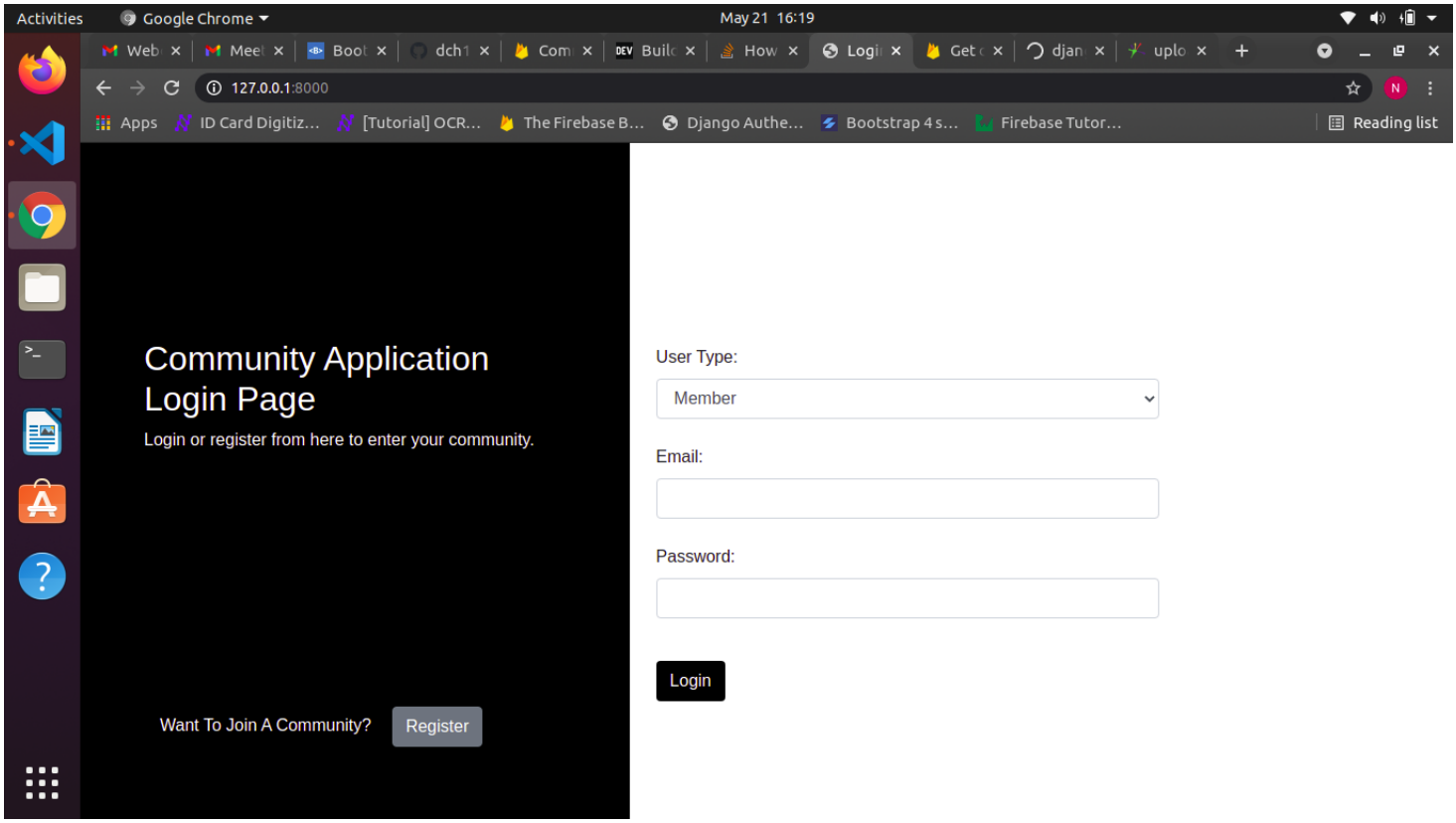


Figure 8.2: RESULT: Home Page

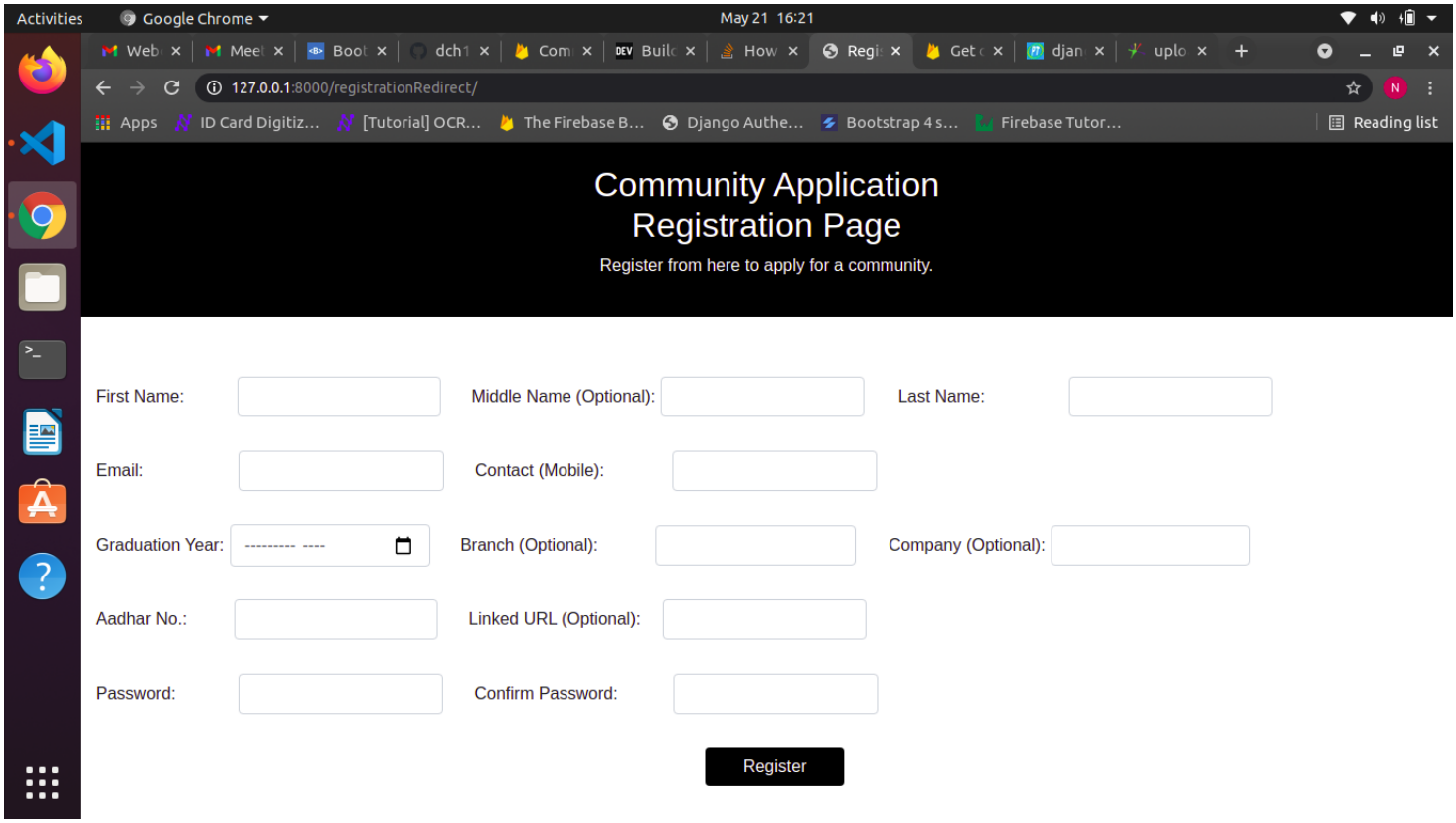


Figure 8.3: RESULT: Registration Page

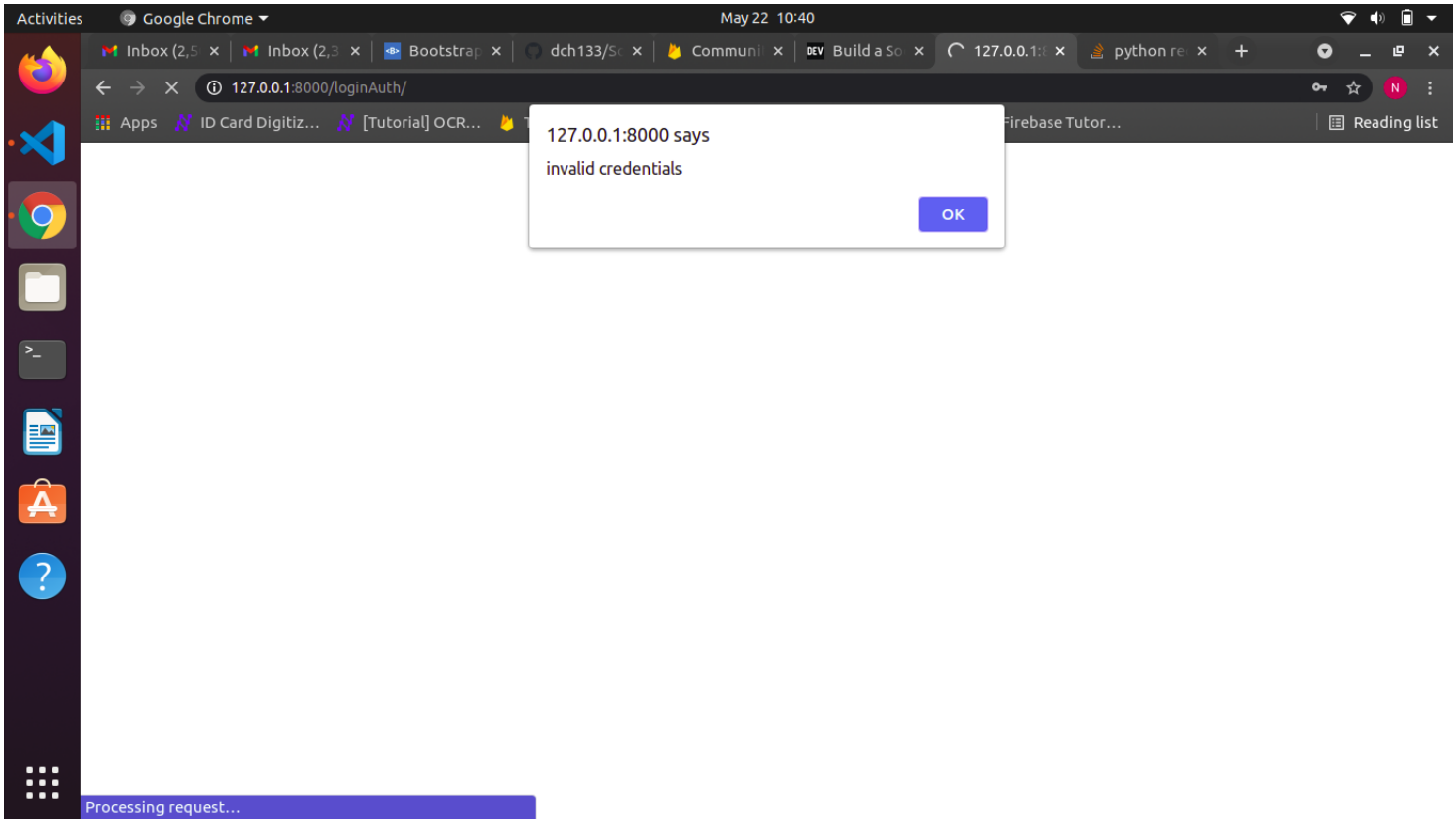


Figure 8.4: RESULT: Credential Check Page

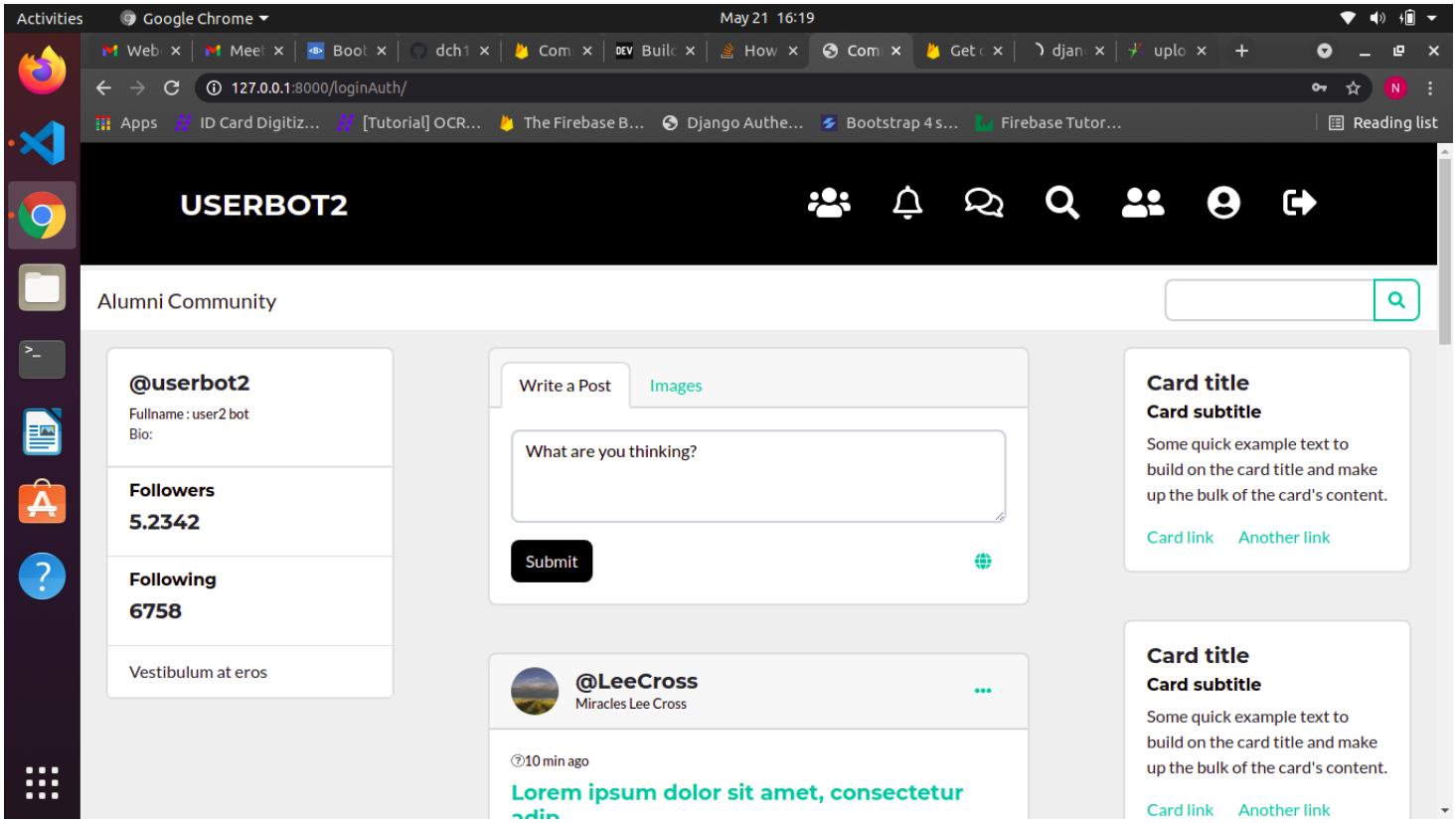


Figure 8.5: RESULT: Dashboard Of Web Application

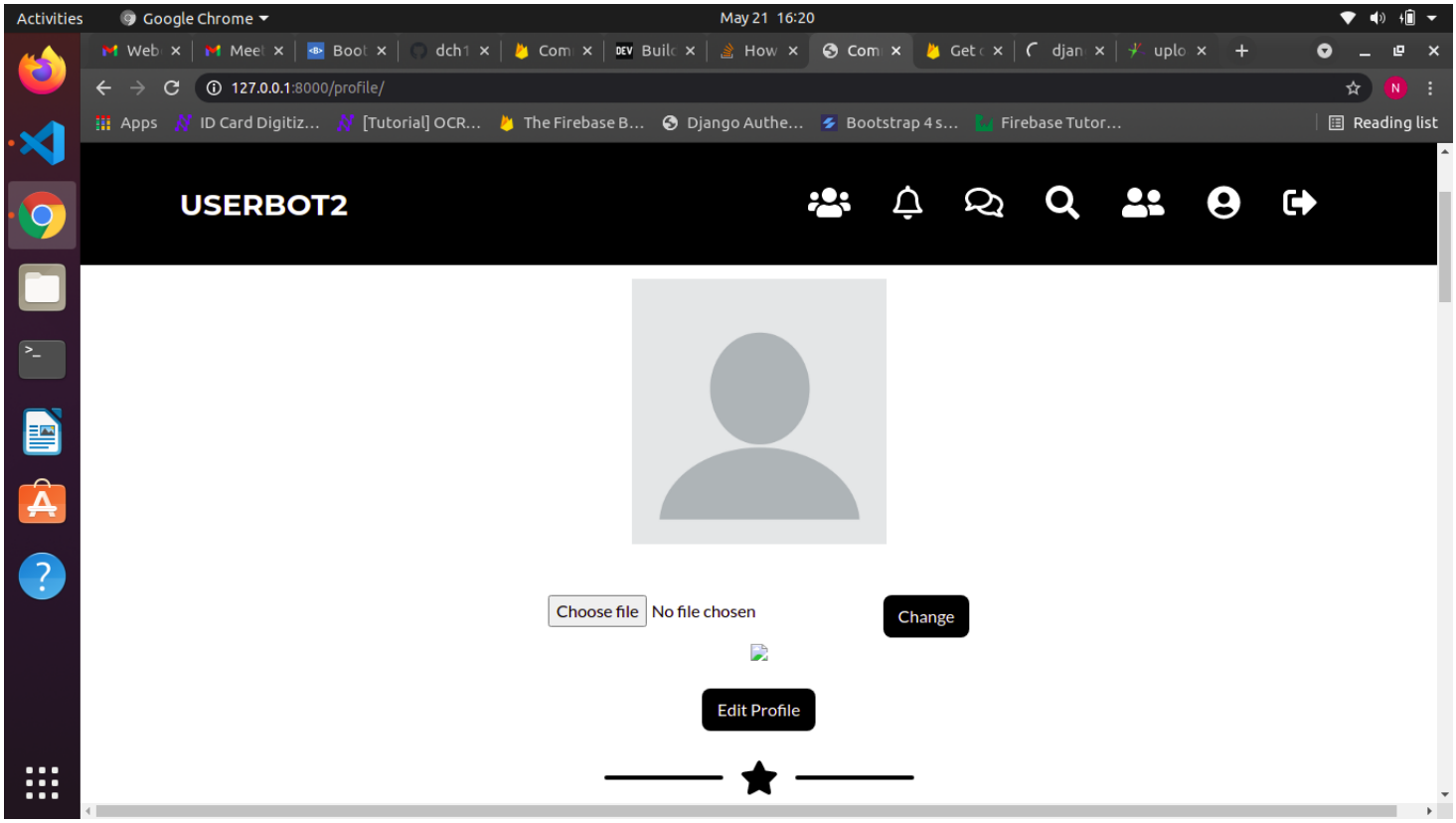


Figure 8.6: RESULT: User Profile

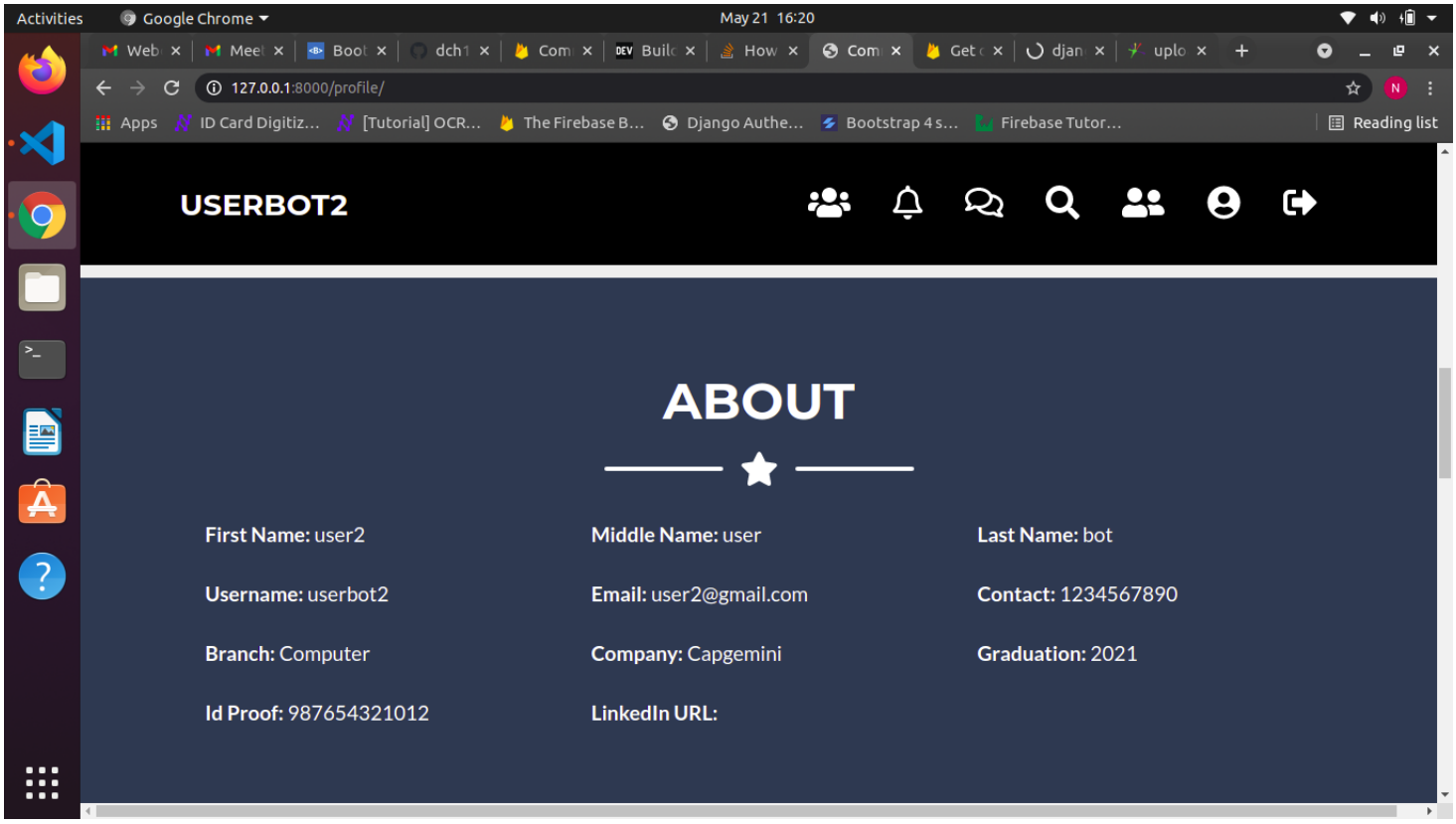


Figure 8.7: RESULT: About User

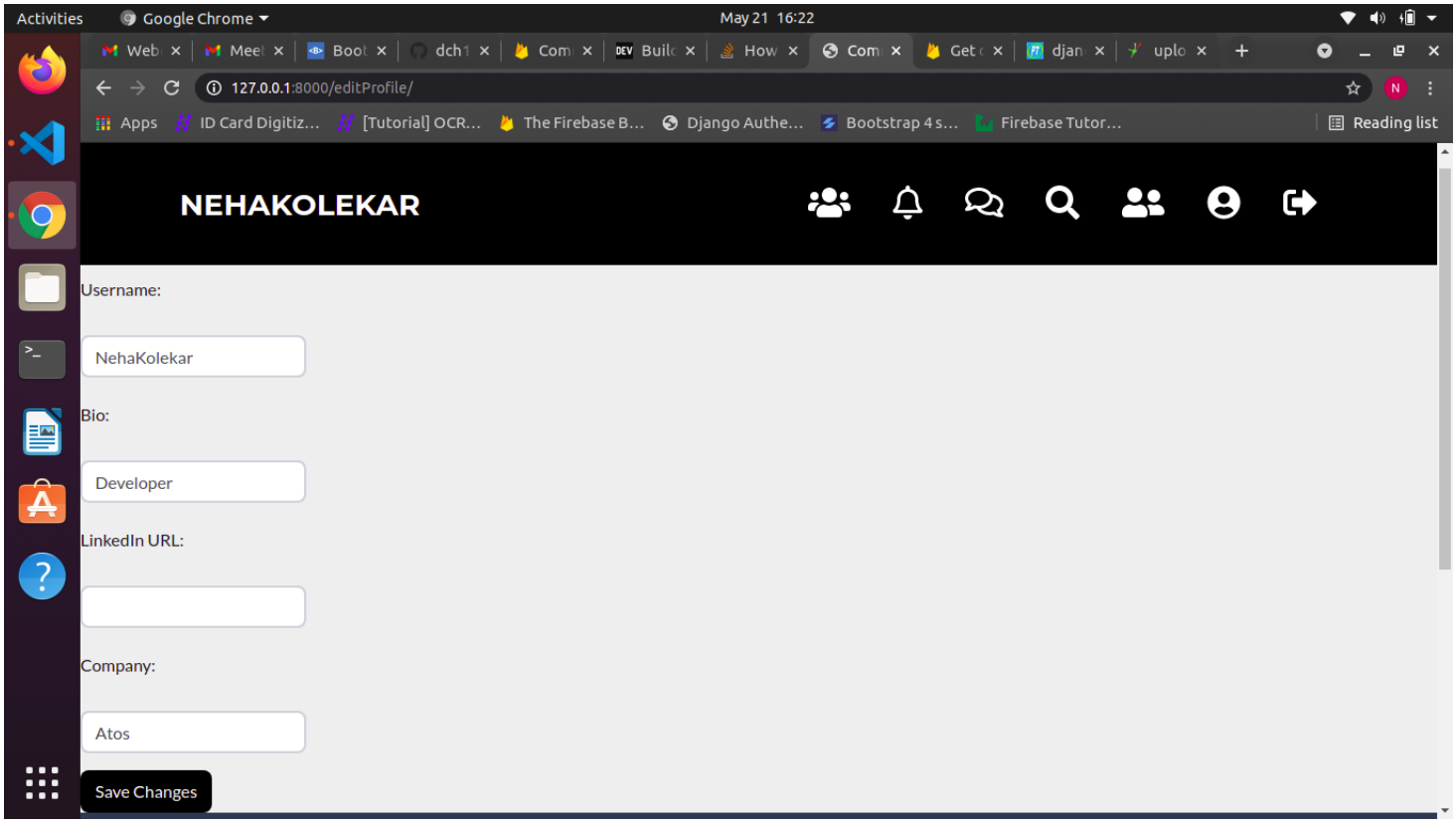


Figure 8.8: RESULT: Edit Profile

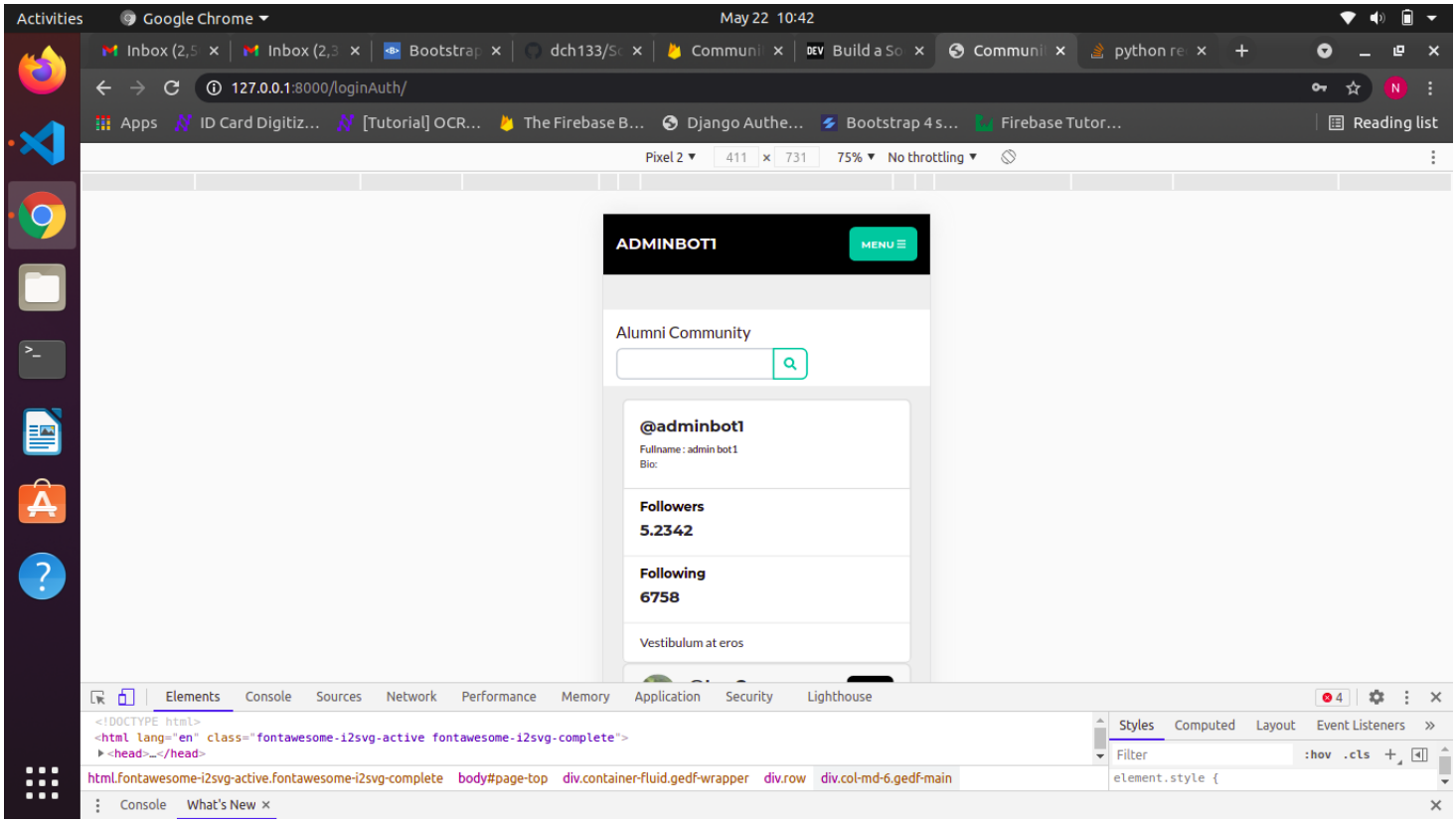


Figure 8.9: RESULT: Response View

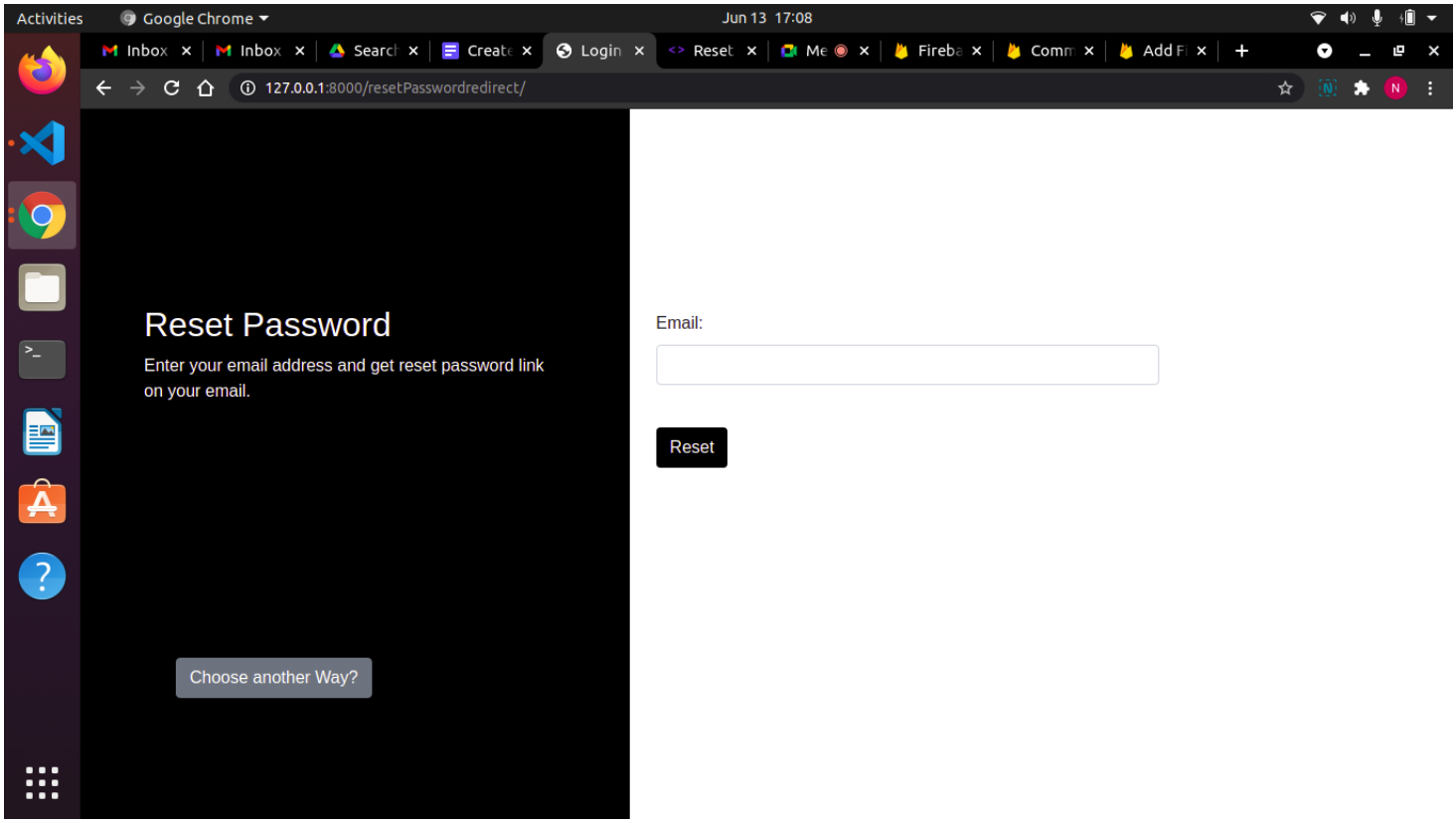


Figure 8.10: RESULT: Enter Email ID to Reset your password

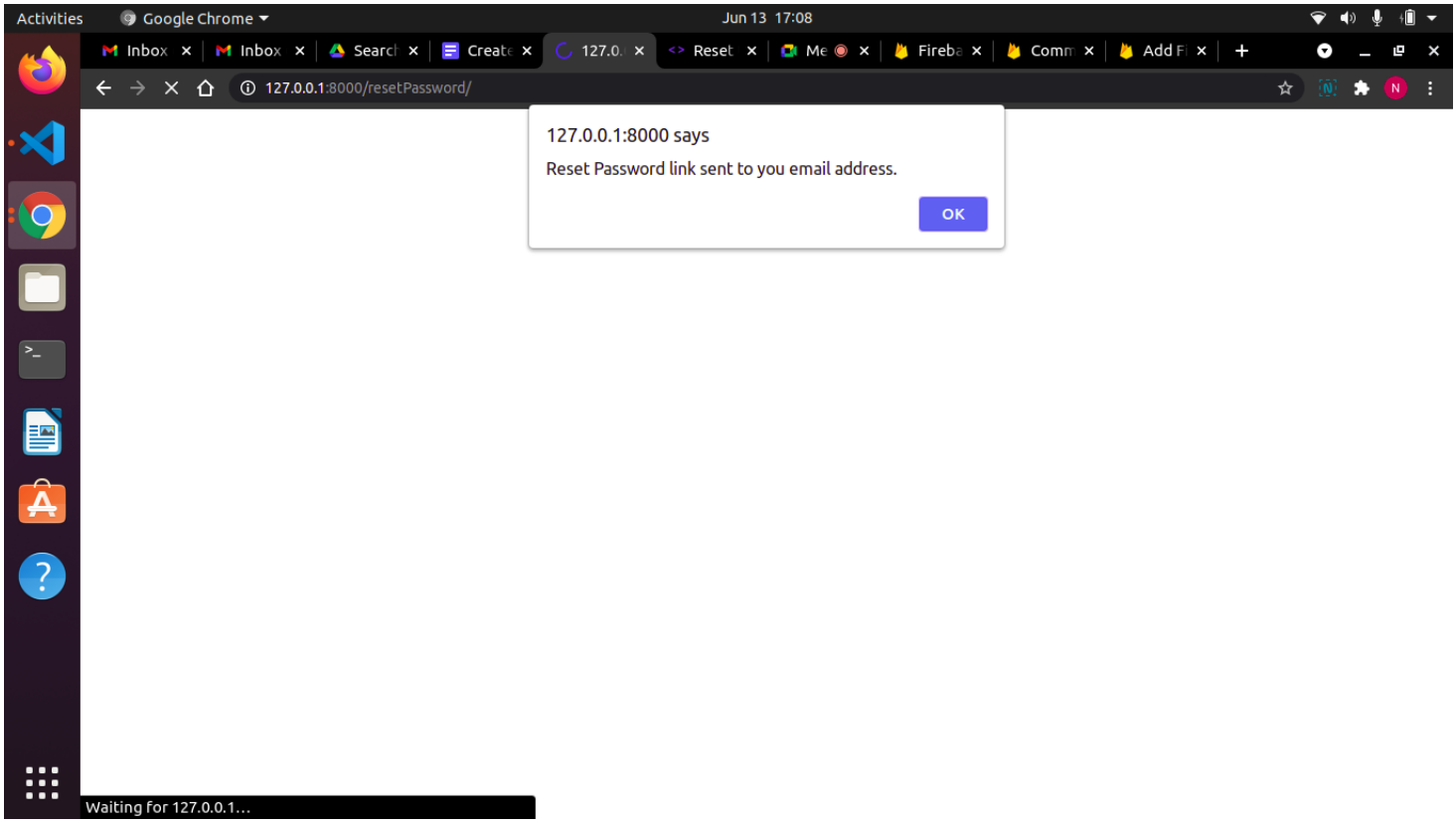


Figure 8.11: RESULT: Reset link sent to email ID

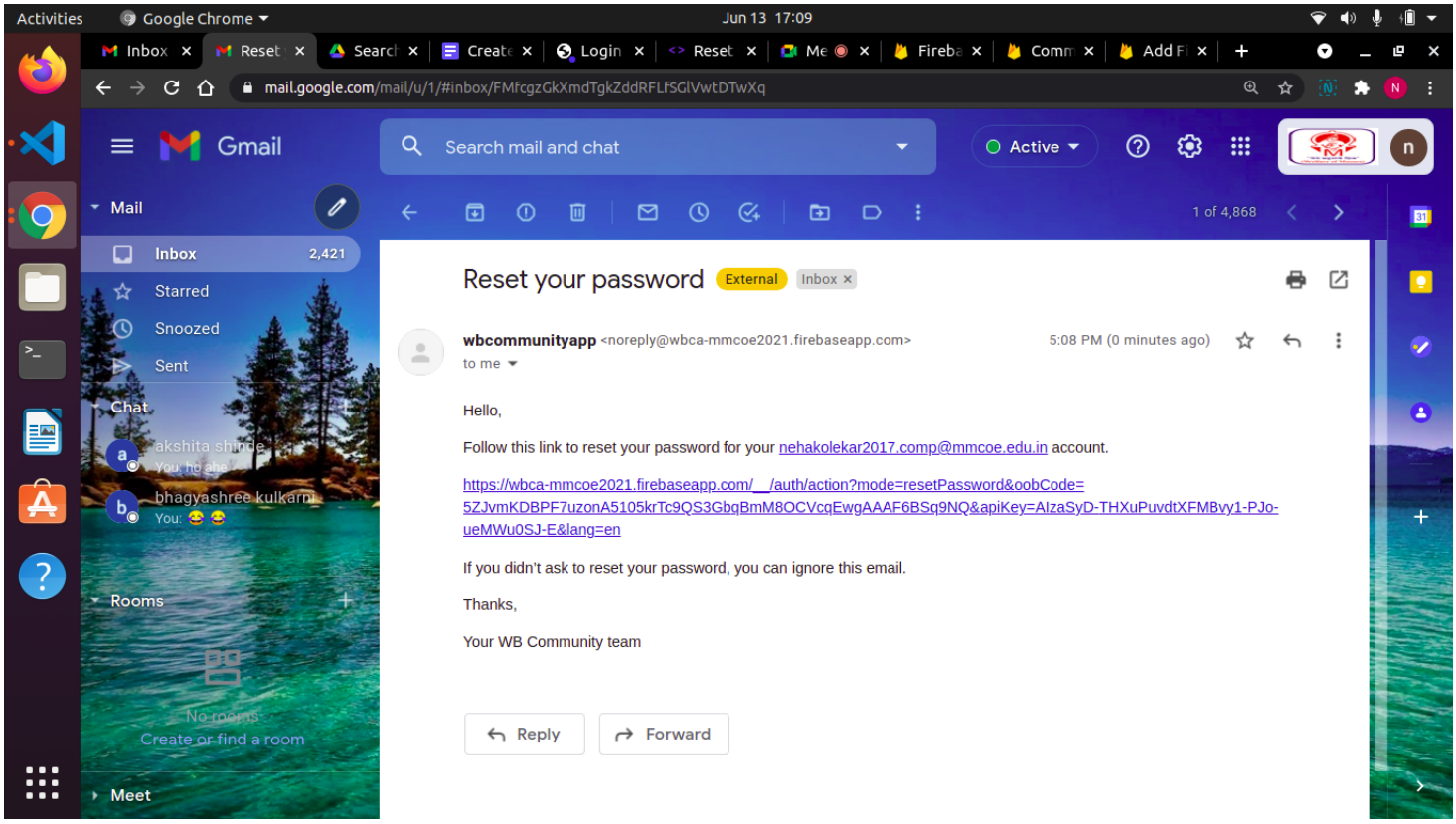


Figure 8.12: RESULT: Reset link received to email ID

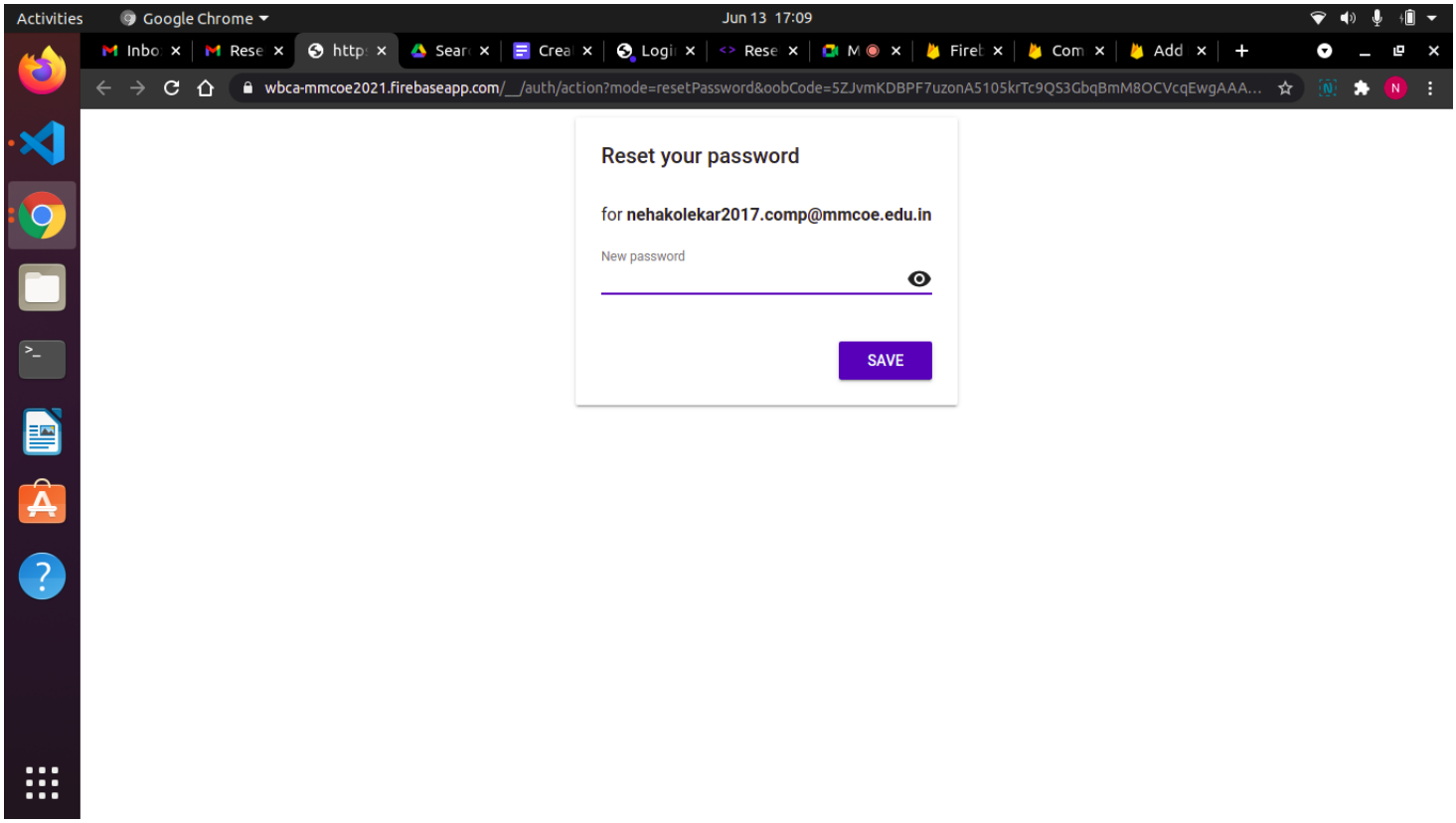


Figure 8.13: RESULT: New Password set

Chapter 9

Other Specifications

9.1 Specification

9.1.1 Advantages

- Keeping your community pre-launch private.
- Deciding which features will be enabled.
- Connect People and Maintain Business Relations.
- Identify key internal stakeholders for the community.
- Admin can monitor what will be posted in the feed Admin will see all the users of community to keep easy logs of members.

9.1.2 Limitations

- Here , we will be making the website in english language but we can convert it later to the other languages.
- Internet Access required.
- The users must be literate to use the system.

9.1.3 Applications

- Provide service to each group , matrimonial , jobs, sports, religion, caste with seperate registration form for each community.

- It can be again used for more advanced features like one to one chat , language detection , spam detection.

Chapter 10

Conclusion and Future Work

10.1 Conclusion

The proposed system can thus be used for connecting various people of Community with each other. The UI helps users to connect with each other easily and contact each other for any further help. Community development will lead people to become more responsible, develop healthy lifestyles, empower, and provide economic opportunities. Community work takes place in particular geographical areas, focusing on identifying their needs, issues and strategies.

10.2 Future Work

The user validation can be done with OTP in future work.
It can also be concentrated on a particular area in identifying the problems of the human beings or region. This application is useful in various areas like alumni, matrimonial sites, language exchange and job distribution.

Chapter 11

APPENDIX A

11.1 Feasibility Study

11.1.1 Operational Feasibility

The system is easy to use, portable, and has more features related to the other systems that are already available in the market.

11.1.2 Technical Feasibility

1) The system will run efficiently if provided with good internet connection.
2) No issues regarding legality will be faced as the libraries and softwares used are open source and free to use.

11.2 Problem Type

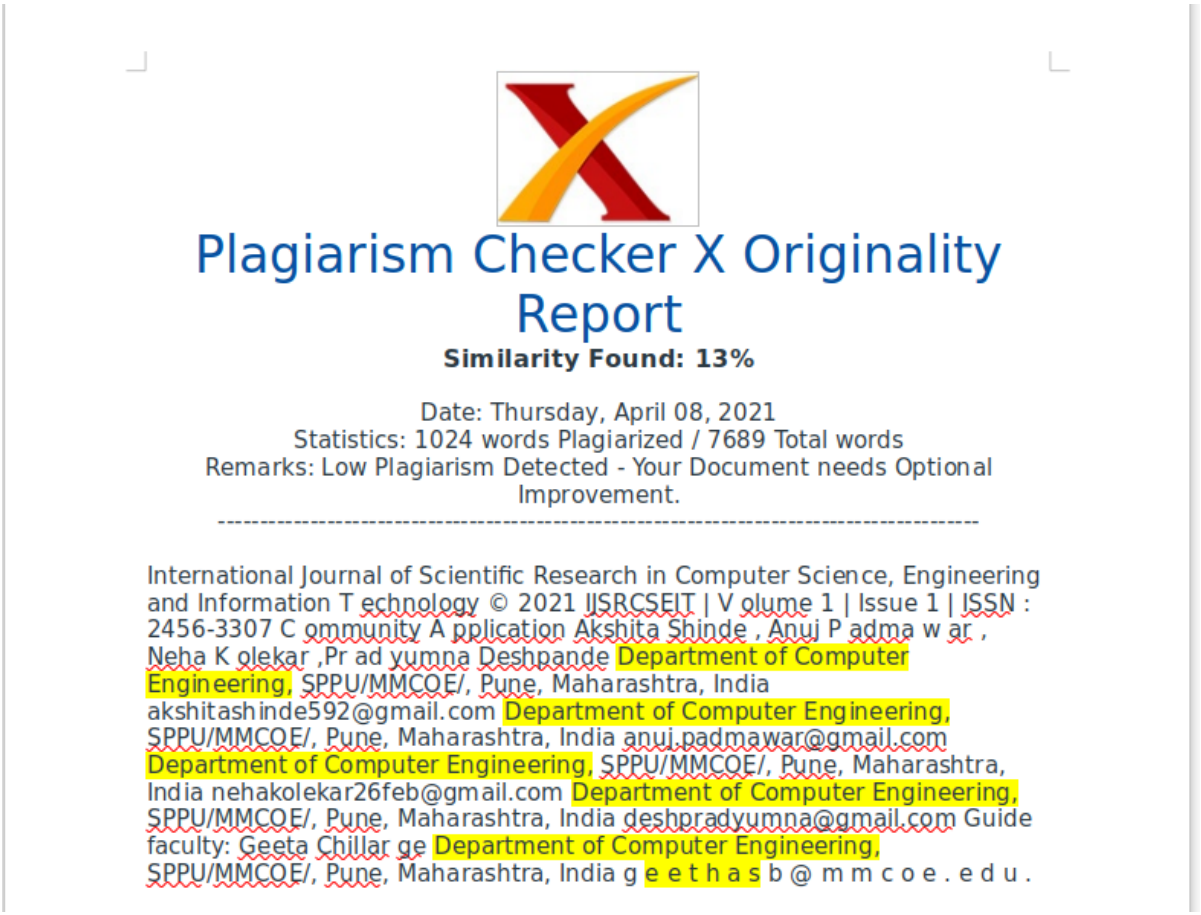
This problem gives a solution in polynomial time. Hence, this problem statement is P type problem.

Chapter 12

APPENDIX B

12.1 Paper Publication

Title: Community Application
Journal : IJSR CSEIT
Year : 2021
Type : UGC Approved
Comments: Accepted and Published



Plagiarism Checker X Originality Report

Similarity Found: 13%

Date: Thursday, April 08, 2021
Statistics: 1024 words Plagiarized / 7689 Total words
Remarks: Low Plagiarism Detected - Your Document needs Optional Improvement.

International Journal of Scientific Research in Computer Science, Engineering and Information T echnology © 2021 IJSRCSEIT | V olume 1 | Issue 1 | ISSN : 2456-3307 C ommunity A pplication Akshita Shinde , Anuj P adma w ar , Neha K olekar ,Pr ad yumna Deshpande Department of Computer Engineering, SPPU/MMCOE/, Pune, Maharashtra, India akshitashinde592@gmail.com Department of Computer Engineering, SPPU/MMCOE/, Pune, Maharashtra, India anuj.padmawar@gmail.com Department of Computer Engineering, SPPU/MMCOE/, Pune, Maharashtra, India nehakolekar26feb@gmail.com Department of Computer Engineering, SPPU/MMCOE/, Pune, Maharashtra, India deshpradyumna@gmail.com Guide faculty: Geeta Chillar ge Department of Computer Engineering, SPPU/MMCOE/, Pune, Maharashtra, India g e e t h a s b @ m m c o e . e d u .

Figure 12.1: Plagarism Check

Bibliography

[1] Flux: [Online] URL: https://facebook.github.io/flux/docs/in-depth-overview.html Accessed April 4 2017 *Application Architecture for Building User Interfaces.*

[2] 6 Web Development Stacks to Try in 2017. 2017. Lozinsky, Yuriy. Web Document. *¡https://webinerds.com/6-web-development-stacks-try-2017/¿. Accessed 16 April 2018.*

[3] Django Documentation – FAQ: General. 2018. Django. Web Document. ¡https://docs.djangoproject.com/en/2.0/faq/general/¿. Accessed 15 April 2018.

[4] Docs.djangoproject.com. 2020. Django. Accessed on 26 April 2020 [online] Available at: ¡ https://docs.djangoproject.com/en/3.0/¿

[5] Lokhande, P. S., Aslam, F., Hawa, N., Munir, J., Gulamgaus, M. (2015). Efficient way of Web Development using Python and Flask. *International Journal of Advanced Research in Computer Science, 54-57.*

[6] Kuhlman, D. (2011). A Python Book: Beginning Python, Advanced Python, and Python Exercises. *Platypus Global Media.*

[7] Jovanovic, eljko Jagodic, Dijana ujicic, Dejan an ic , Sinis a, Java Spring Boot est EB Service Integration with Java Artificial Intelligence Weka Framework 2017 INTE NATIONAL SCIENTIFIC CONFERENCE, GABROVO

[8] Mohamed Abdalla Mokar, Sallam Osman Fageeri, Saif Eldin Fatton "Using Firebase Cloud Messaging to Control Mobile Community Application", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN :2456-3307, Volume 7 Issue 3, pp. 37-43, May-June 2021

[9] Prof. B Nithya Ramesh, Aashay R Amballi, Vivekananda PYTHON Department Mahanta. WEB of "DJANGO THE FRAMEWORK" 2019 Master of Computer Applications, NHCE, Bangalore, India May 2019.

[10] Jyoti Shetty, Deepika Dash, Akshaya Kumar Joish, Guruprasad, "Review Paper on Web Frameworks, Databases and Web Stacks", 2020, International Research Journal of Engineering and Technology (IRJET), Apr 2020, e-ISSN:2395-0056

[11] S.L. Kavya, Dr. S. Sarathambekai "Python Libraries and Packages for Web Development- A Survey" .2019. International Journal of Innovative Research in Technology May 2019 ISSN: 2349-6002.

[12] Joel Vainikka, "Full-stack web development using Django REST framework and React". 2018 Metropolia University of Applied Sciences. (16 May 2018)