

**RV COLLEGE OF ENGINEERING ®**  
**BENGALURU-560059**  
**(Autonomous Institution Affiliated to VTU, Belagavi)**



**“Student Career Portal”**  
**Report**  
**Database Design Laboratory Project**  
**(18CS53)**  
**Submitted By**  
Ankit Kumar Singh (1RV18IS007)  
Ayush Kumar(1RV18IS009)



**Under the Guidance of**  
**Dr. Padmashree T.**  
**Asst. Professor**

**in partial fulfillment for the award of degree of**  
**Bachelor of Engineering**  
**in**  
**INFORMATION SCIENCE AND ENGINEERING**  
**2020-21**

**RV COLLEGE OF ENGINEERING®, BENGALURU - 560059**  
**(Autonomous Institution Affiliated to VTU, Belagavi)**

**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**



**CERTIFICATE**

Certified that the Mini Project work entitled ‘Student Career Portal’ has been carried out as a part of Database Design Laboratory(18CS53) in partial fulfillment for the award of degree of **Bachelor of Engineering in Information Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year **2020-2021** by **Ankit Kumar Singh (1RV18IS007), Ayush Kumar (1RV18IS009)**, who are bonafide students of **RV College of Engineering®**, Bengaluru. It is certified that all the corrections/suggestions indicated for the internal assessment have been incorporated in the report deposited in the departmental library. The report has been approved as it satisfies the academic requirements in respect of work prescribed by the institution for the said degree.

**Dr. Padmashree T.**

**Asst. Professor**

Department of ISE,  
RVCE, Bengaluru-59

**Dr Sagar B M**

**Head of the Department**

Department of ISE,  
RVCE, Bengaluru-59

Name of the Examiners

Signature with Date

1. \_\_\_\_\_

\_\_\_\_\_

2. \_\_\_\_\_

\_\_\_\_\_

**RV COLLEGE OF ENGINEERING®, BENGALURU - 560059**  
**(Autonomous Institution Affiliated to VTU, Belagavi)**

**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**

**DECLARATION**

We Ankit Kumar Singh, Ayush Kumar are students of Fifth Semester B.E Department of Information Science and Engineering, **RV College of Engineering ®**, bearing USN: **1RV18IS007, 1RV18IS009**, hereby declare that the project titled “**Student Career Portal**” has been carried out as a part of Database Design (18CS53) by us and submitted in partial fulfillment of the program requirements for the award of degree in Bachelor of Engineering in Information Science and Engineering of the **Visvesvaraya Technological University, Belagavi** during the year **2020-2021**.

Further we declare that the content of the report has not been submitted previously by anybody for the award of any degree or diploma to any other University.

<b>Place:</b> Bengaluru	<b>Name</b>	<b>Signature with Date</b>
<b>Date:</b>	<b>Ankit Kr Singh</b>	
	<b>Ayush Kumar</b>	

## TABLE OF CONTENTS

<b>CHAPTER 1</b>	<b>6</b>
INTRODUCTION	6
1.1 Terminology	6
1.2 Purpose	6
1.3 Motivation	6
1.4 Problem Statement	7
1.5 Objective	7
1.6 Scope and Relevance	7
<b>CHAPTER 2</b>	<b>8</b>
REQUIREMENT SPECIFICATION	8
2.1 Specific Requirements	8
2.1.1 Functional Requirements	9
2.1.2 Non-Functional Requirements	10
2.2.2 Hardware Requirements	12
2.2.3 Software Requirements	13
<b>CHAPTER 3</b>	<b>14</b>
DESIGN	14
3.1 E-R Diagram	15
3.1.1 Schema Representation	16
3.2 UML Diagrams	17
3.2.1 Class Diagram	18
3.2.2 Sequence Diagram	19
3.2.3 Activity Diagram	21
3.3 Front End Design	22
<b>CHAPTER 4</b>	<b>25</b>
IMPLEMENTATION DETAILS	25
4.1 Database implementation	25
4.1.1 Table Creation	26
4.1.2 Table Population	29
4.1.3 Query Execution and Output	29
4.2 Front End implementation	31
4.2.1 Form Creation	28
4.2.2 Connectivity to the Database	28
4.2.3 Report generation	28
4.2.4 Security features	28

<b>CHAPTER 5</b>	<b>35</b>
TESTING AND RESULTS	36
5.1 Database Testing	36
5.1.1 Test cases	36
5.2 Front End Testing	37
5.2.1 Test cases	37
5.3 System Testing	38
5.3.1 Test cases	39
<b>CHAPTER 6</b>	<b>40</b>
CONCLUSION	40
6.1 Limitations	40
6.2 Future Enhancements	40
REFERENCES	41
<b>APPENDIX A- CODE SNIPPETS</b>	<b>42</b>
<b>APPENDIX B –SCREENSHOTS</b>	<b>45</b>

# CHAPTER 1

## Introduction

### 1.1 Terminology

The terminologies used in the following document are given by:

- **HTML:** Hypertext Markup Language is the standard markup language to design documents which are to be displayed in the web browser.
- **CSS:** Cascading Style sheets is a language used to style or change the presentation of a document written in a markup language.
- **JAVASCRIPT:** JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm.
- **Django:** Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.
- **PostgresQL:** PostgreSQL is a powerful, open source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.

### 1.2 Purpose

In nearly all educational institutes placement record holds a major criteria for analysing quality of education and performance of an Institute. These institutes also act as a bridge between industries and students and bring these two stakeholders together. Having a robust platform for proper interaction between them will make recruitment streamlined. Better recruitment improves utilization of human resources and contributes to the nation's economy.

### 1.3 Motivation

Proposed portal is meant as a replacement for an already existing portal. The career portal that currently exists deals with problems like lack of adequate functionalities, improper management, major bugs in the design as well as improperly working features. Though present, the above problems have made it difficult for the students to use it as their goto platform for placement activities. Also, the current platform is only for placement purposes and isn't applicable for internships.

Keeping in mind the above problems, the new portal is being built to provide much more versatility and flexibility in terms of features and better efficiency and performance. It is an open source project with an active developer team to support it. It is developed to run on Web Browsers as of now, but can be turned into a Progressive Web App (PWA) as well.

## 1.4 Problem Statement

The portal is being designed for the department, faculties and students, to keep them updated about the internship/placement statistics as well as the upcoming opportunities. The portal will help students to efficiently manage their time and will provide them with an all-in-one hub for internship/placement activities. This portal will help the students and placement department to efficiently manage applications for companies and keep track of offers made to the students.

## 1.5 Objective

The portal is being designed for the department, faculties and students, to keep them updated about the internship/placement statistics as well as the upcoming opportunities. The portal will help students to efficiently manage their time and will provide them with an all-in-one hub for internship/placement activities. This portal will help the students and placement department to efficiently manage applications for companies and keep track of offers made to the students.

The plan will include, but is not restricted to, a summary of functionalities, features of the portal, the structure of involved software, the workflow we will adopt, societal concerns regarding the Portal and the summary of technical trends we will use in the portal.

Proposed system will be built with Django as backend and Postgresql RDBMS. It will provide role based access to Students, SPCs and Admins to efficiently manage applications and recruitment of student records. It will also help in generating reports for department wise statistics. The portal will be useful to :

- **Students :** For instant update on placement activities and eligibility criteria.
- **Placement Coordinators:** Managing applications for various offers made by different companies.
- **Placement Department:** Handling placement and user records.
- **Faculties:** Getting details of department wise placement statistics and reports.

## 1.6 Scope and Relevance

The portal will be used to carry out internship and placement activities throughout the campus. Students will be able to keep track of companies which are currently visiting or are going to visit the campus and will be able to apply for the same. Apart from the ongoing session, the portal will also contain statistics of previous placement sessions as well. As an admin, one will be able to manage all the users registered as students and will have privileges to add or remove companies from the portal as per the need.

Faculty members will be able to monitor the progress of their students regarding the same by viewing the statistics of the ongoing session. Councilors, if the need persists, will be able to contact the student regarding the same.

# **CHAPTER 2**

## **Requirement Specifications**

### **2.1 Specific Requirements**

Signup/Login:

- New user registration : Signup system for users with google oauth
- Multiple login system for students, SPCs , Faculties and Admin

Profile:

- Edit profile : Students needs to add the required details
- Contact details : email, address, phone number to be associated with user
- Educational details: educational details to be associated with user

Applications:

- New: Fill up a new application
- View: View the existing applications
- Cancel: Cancel an application previously applied to

Statistics:

- View stats of previously held placement sessions
- Filter the stats by year ,company or department

Admin:

- Functionality to add SPCs and Faculties
- Admin should be able to add new companies and offers
- Functionality to edit applications details by SPCs and faculties

Others:

- Students should be able to list upcoming companies offers
- Students should be able to get information about their application progress
- Updates to students via sms / email
- Student should be able to filter companies according to their eligibility criteria

## **2.2 Functional Requirements**

### **2.2.1 Authentication**

#### **2.2.1.1 Description and Priority**

All users using the software must be authenticated. The software will support three different user accounts. The Administrator account will have access to all data in the database and all features. faculties and placement coordinators accounts will have limited access to the database, as they can only interact with data of their respective students. The Administrator will be able to create new Administrator, Professor and Mentor accounts and can also remove them from the database as well.

#### **2.2.1.2 Stimulus/Response Sequences**

Each user accessing the application will authenticate with google Oauth via their college email address.

- On successful completion of authentication users will login to the portal and can access various other sections.
- On unsuccessful attempts users will be prompted to authenticate again.
- Admin users can access a different restricted section to manage the database.
- user can press logout to log out of the session.

#### **2.2.1.3 Functional Requirements**

REQ-1-1: Software should be able to maintain separate record of different kind of users

REQ-1-2: Software should preserve records of all users in the database.

REQ-1-3: Once successfully authenticated users should be asked to login again until session expires.

REQ-1-4: Users should be able to logout and their session token must get invalidated.

REQ-1-5: Software should display only details of their profile to the user.

REQ-1-6: Software should only allow administrators to access the restricted section.

REQ-1-7: Coordinators should have limited access to the database.

## 2.2.2 User Profile

### 2.2.2.1 Description and Priority

All students using the portal should have a profile to represent their educational and contact details. These details should be associated with their application when applying to the offers provided by companies.

### 2.2.2.2 Stimulus/Response Sequences

Authenticated student users will have to update their profile to fill up their contact and education details. Contact details include email address, phone number, address. Educational details include 10th , 12th percentage, semester wise SGPA, CGPA, resume etc.

### 2.2.2.3 Functional Requirements

REQ-2-1: Personal details of students should only be editable by the user or administrator.

REQ-2-2: User details must be validated before entering in the database.

REQ-2-3: Students should be able to update their profile as per their interest.

REQ-2-4: Students details should be associated with applications sent to company offers.

## 2.2.3 User Privileges

### 2.2.3.1 Description and Priority

Users will have three different types. Students will have access to their profiles and application only. Coordinators can manage applications and offers. Administrators will be able to access all data on the database.

### 2.2.3.2 Stimulus/Response Sequences

Coordinators and administrators can access the admin interface. coordinators can manage offers and applications made by students for the assigned departments. Administrators can assign faculty and coordinators roles to users.

### 2.2.3.3 Functional Requirements

REQ-3-1: Students will only be able to update their profile and applications.

REQ-3-2: Coordinators can manage offers and applications made by students.

REQ-3-3: Administrators can assign faculty and coordinators roles to users.

## 2.2.4 Offers and Applications

### 2.2.4.1 Description and Priority

A record of companies visiting the campus will be maintained. Administrators will add these records and their offers as per constraints provided by the company. Students can apply for various positions as per their eligibility criteria.

### 2.2.4.2 Stimulus/Response Sequences

When a company arrives for recruitment they will provide their details to the placement department about their eligibility criteria, selection process etc. These details will be fed into the database and students will receive notification regarding the offer. Eligible students will be able to apply for the company and send their application. Coordinators can review applications and update as per requirement.

### 2.2.4.3 Functional Requirements

REQ-4-1: Administrators will be able to add company details and offers.

REQ-4-1: Eligible Students can apply for roles as per their interest.

REQ-4-2: Coordinators can review applications and update their status.

## 2.2.5 Report and Statistics

### 2.2.5.1 Description and Priority

Over the recruitment process overall statistics of department wise data and charts needs to be generated to visualize the performance of students. These data can be grouped according to department wise for easier understanding.

### 2.2.5.2 Stimulus/Response Sequences

Users can generate reports on aggregate of various data through a dashboard to visualize and understand the statistics.

### 2.2.5.3 Functional Requirements

REQ-5-1: Users should be able to see charts from the dashboard.

REQ-5-2: Faculties should be able to download reports for department wise placement records.

## **2.3 NON FUNCTIONAL REQUIREMENT**

### **2.3.1 Performance Requirements**

- Portal should be able to serve 100+ concurrent user requests.
- Portal should be accessible anytime, anywhere.
- Portal can have a maintenance period of 6 hours.
- Portal should be easy to navigate and responsive to all screen sizes.
- Average response time should not be more than 2500 ms.

### **2.3.2 Safety Requirements**

- Periodic backup of database to be made to avoid loss of data.
- Regular monitoring of the servers and database is required.
- Periodic maintenance of software to be done.

### **2.3.3 Security Requirements**

- Users will be authenticated with google Oauth, software have access to users google email address , name and profile picture.
- User's profiles will be shared with 3rd party companies.
- HTTPS secure connection will be established for communication.
- Portal will have Cross site scripting (XSS) protection. XSS attacks allow a user to inject client side scripts into the browsers of other users.
- Portal will have Cross site request forgery (CSRF) protection. CSRF attacks allow a malicious user to execute actions using the credentials of another user without that user's knowledge or consent.
- Portal will have SQL injection protection. SQL injection is a type of attack where a malicious user is able to execute arbitrary SQL code on a database.
- Portal will have Clickjacking protection. Clickjacking is a type of attack where a malicious site wraps another site in a frame.

### **2.3.4 Software Quality Attributes**

- Software should have a clean and easy to understand codebase.
- Software should have well written documentation.
- Software should be written in well commented, modularized components.
- Software should be scalable.
- Software should be easy to debug and add new features.

### **2.3.5 Business Rules**

Initially, only the admin for the portal, ie., the placement cell of the college, will be able to add the maintainers of the portal who will further add the SPC's and the necessary faculty members in the portal. These SPC's and teachers will further ask the required batch of students to enroll themselves on the portal. The portal provides privileges to SPC's to add and remove the Companies and the subsequent offers made by them. Also, in the application being filled by the students, if any discrepancies are found, only the admin will be allowed to correct them.

Students on the other hand will not have the above mentioned privileges. On registration, they will be asked to complete their profile by filling in certain details necessary with respect to internships/placements. Apart from that, they will be able to see placement statistics and apply for the eligible positions.

## **2.4 Hardware Requirements**

- F1 micro instance Virtual Machine (Google Cloud Platform)
- 2 GB RAM
- 1.8 Ghz or faster processor
- 3 TB Disk
- AWS S3 bucket
- GCP Cloud storage

## **2.5 Software Requirement**

- Django
- Chartjs, numpy, matplotlib, Seaborn
- Postgresql RDBMS
- LucidChart
- Google Docs
- Git version control
- Docker
- Ubuntu 18.6 Operating system
- Nginx
- Draw.io
- RabbitMQ | Redis
- Celery & Celery beat
- SMTP mailing service

# CHAPTER 3

## Design

The following are the notable entities and their attributes which will be used in the project:

- **User:** Username, Password, Email, Privileges
- **Student:** Personal Details(Name, DOB, Gender), USN, Academic Details{GPA(CGPA, SGPA), Branch, Admission Mode}, Contact Details(Address, Contact No.)
- **Faculty:** Name, Email, Phone No., Branch
- **Company:** Name, Description, Website
- **Offer:** Type, Timeline(Application Date, Exam Date), Package, Eligibility(Branch, CGPA Cut-Off), Description

For the portal open source PostgreSQL RDBMS is used. PostgreSQL is a highly stable database management system and a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness. PostgreSQL runs on all major operating systems, including Linux, UNIX and Windows. It supports text, images, sounds, and video, and includes programming interfaces for C / C++, Java, Perl, Python, Ruby, Tcl and Open Database Connectivity (ODBC).

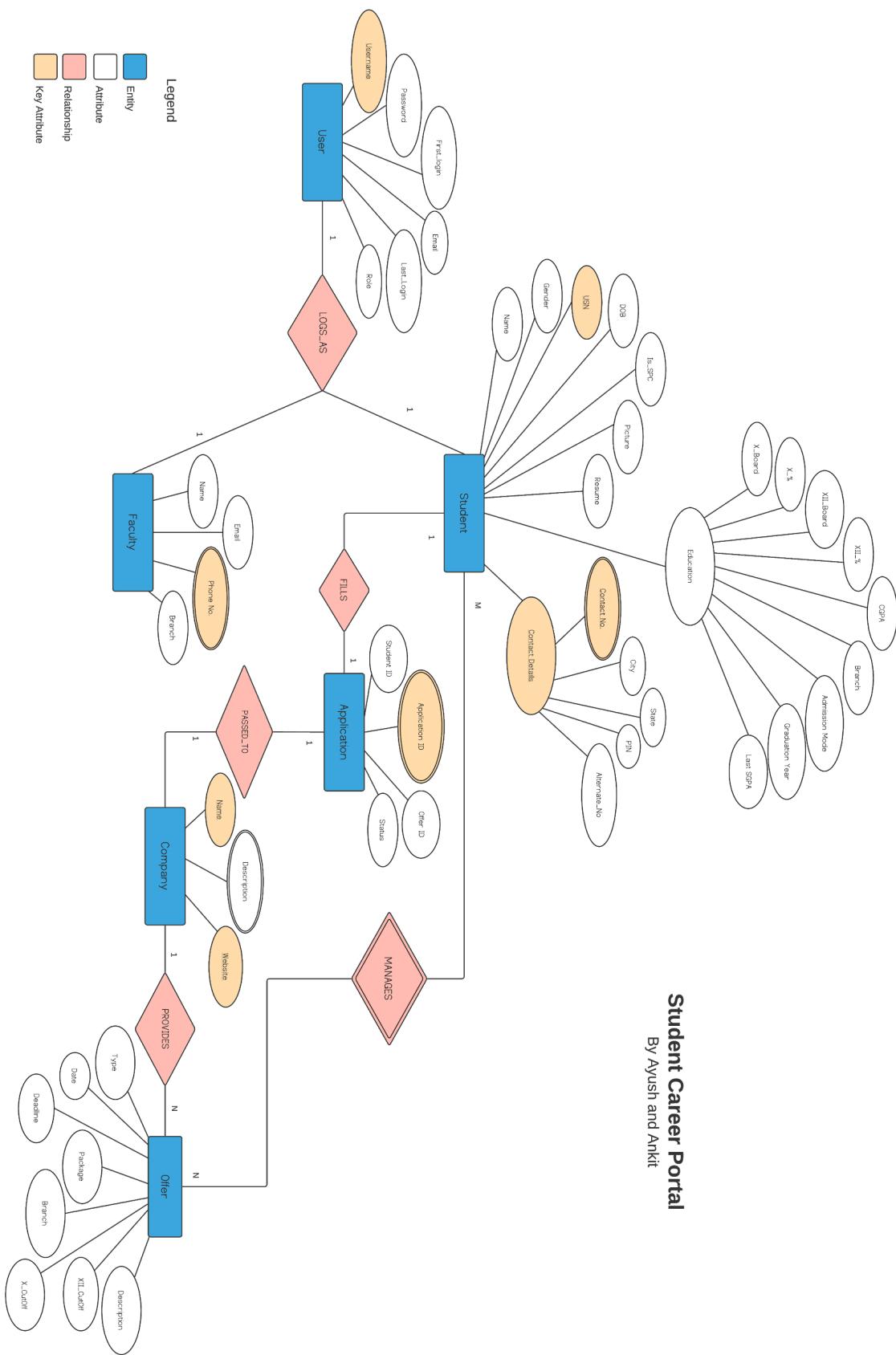
Following are some of the main reasons for using PostgreSQL in the project:

1. Django provides a number of data types that will only work with PostgreSQL.
2. Django has `django.contrib.postgres` to make database operations on PostgreSQL.
3. If you are building an application with maps or you are storing geographical data, you need to use PostgreSQL, as GeoDjango is only fully compatible with PostgreSQL.
4. PostgreSQL has the richest set of features that are supported by Django.

### 3.1 Entity-Relationship Diagram

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research.

Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

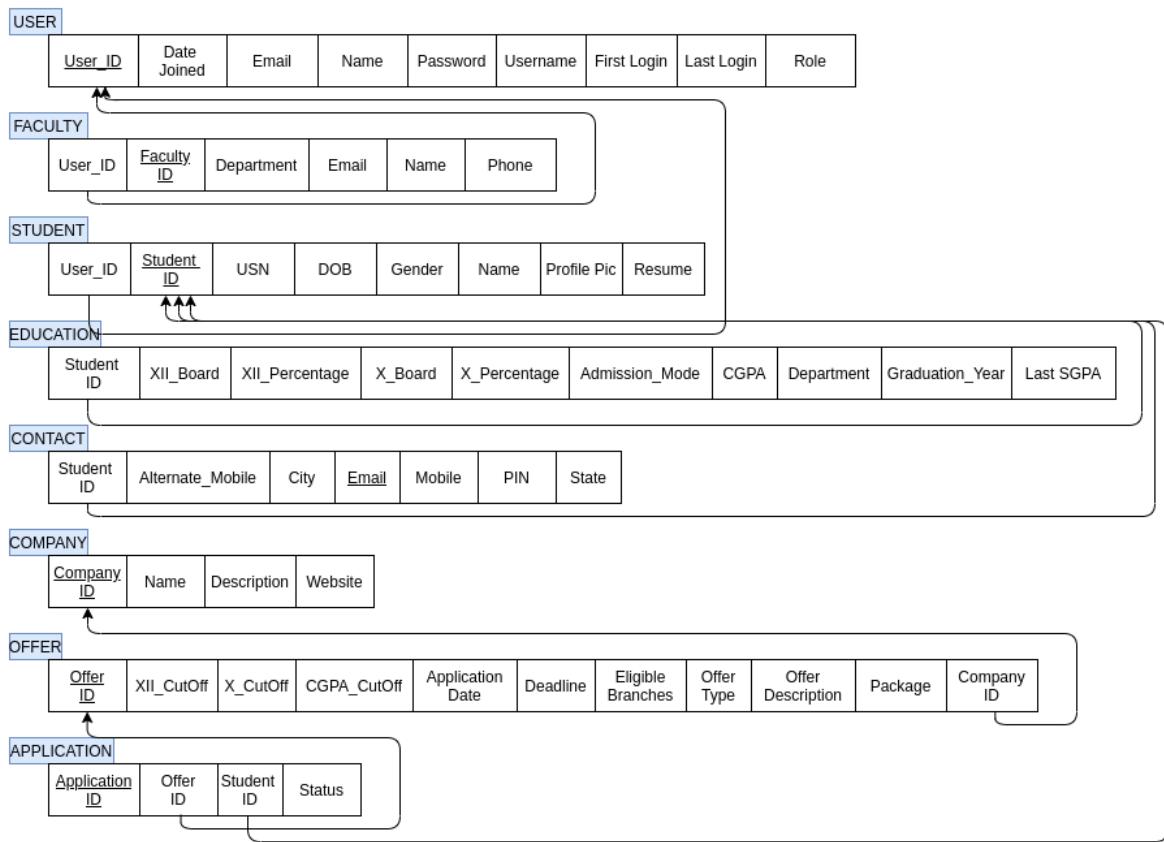


**Fig. 1 E-R Diagram**

### 3.1.1 Database Schema

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It's the database designers who design the schema to help programmers understand the database and make it useful.



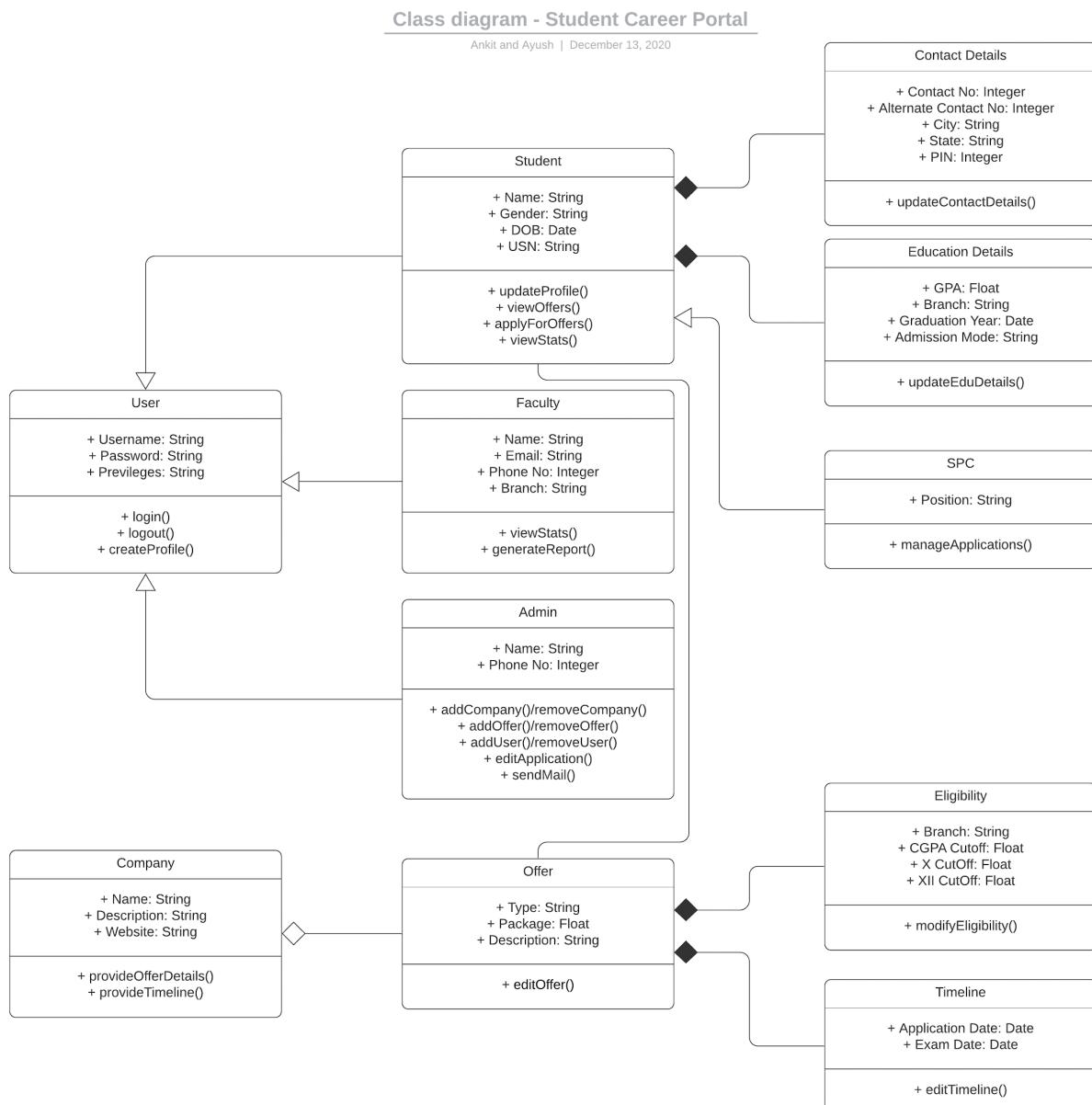
**Fig. 2 Database Schema**

## 3.2 UML Diagrams

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

### 3.2.1 Class Diagram

In software engineering, a class diagram in the Unified Modelling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.



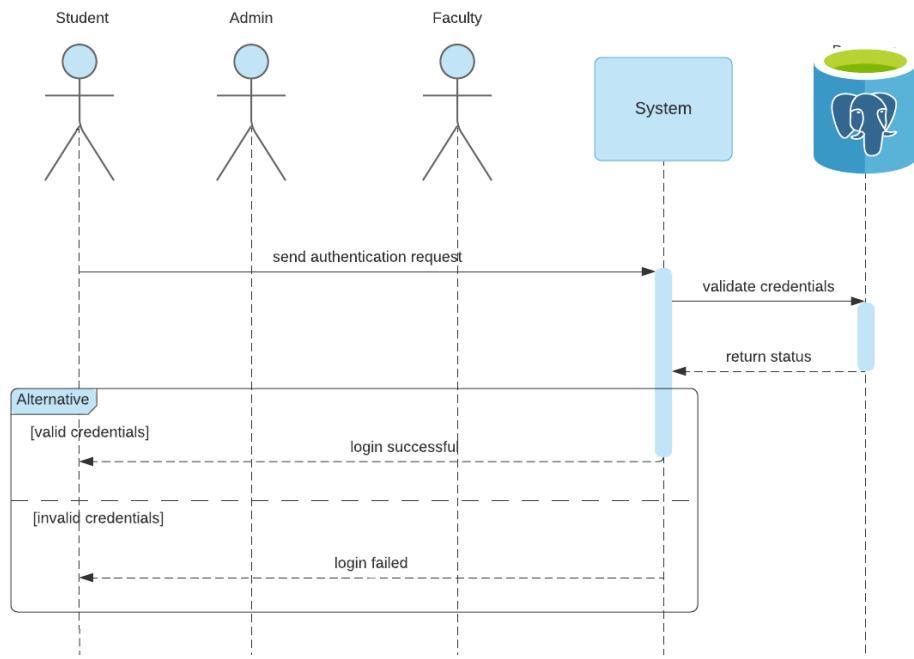
**Fig. 3 Class Diagram**

### 3.2.2 Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

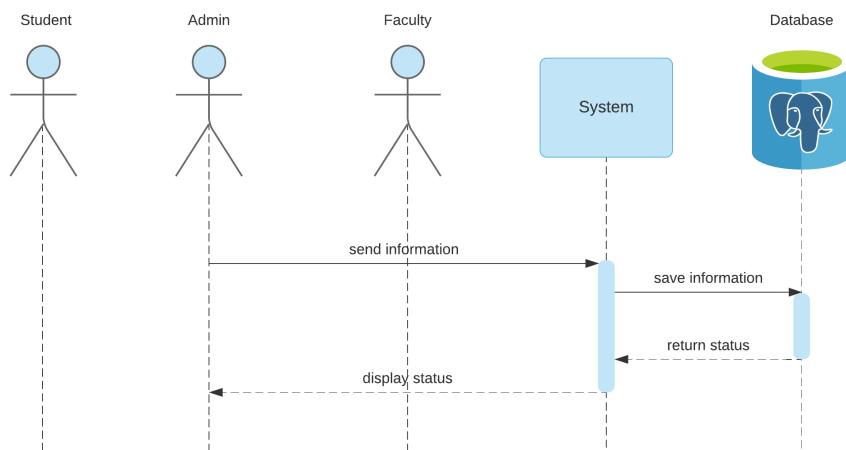
The dynamic behavior of the system is modeled by figuring out the interactions between the classes involved in each principal action.

#### 1. Login



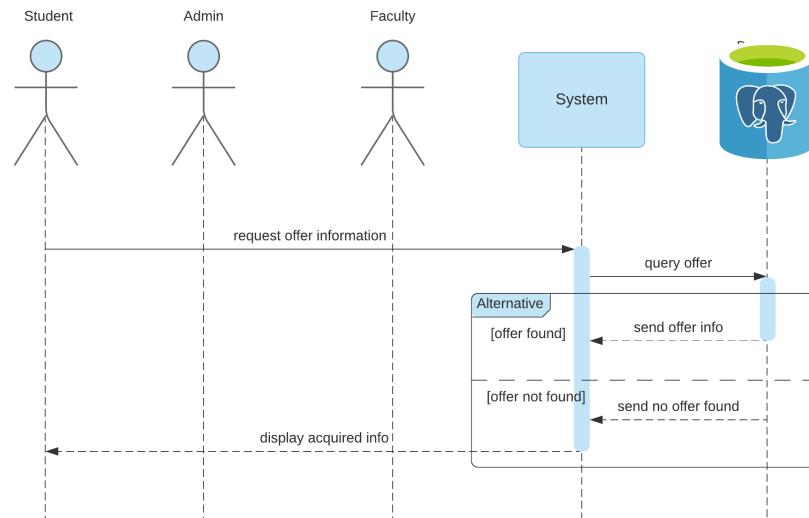
**Fig. 4 Login Sequence Diagram**

#### 2. Add Offers



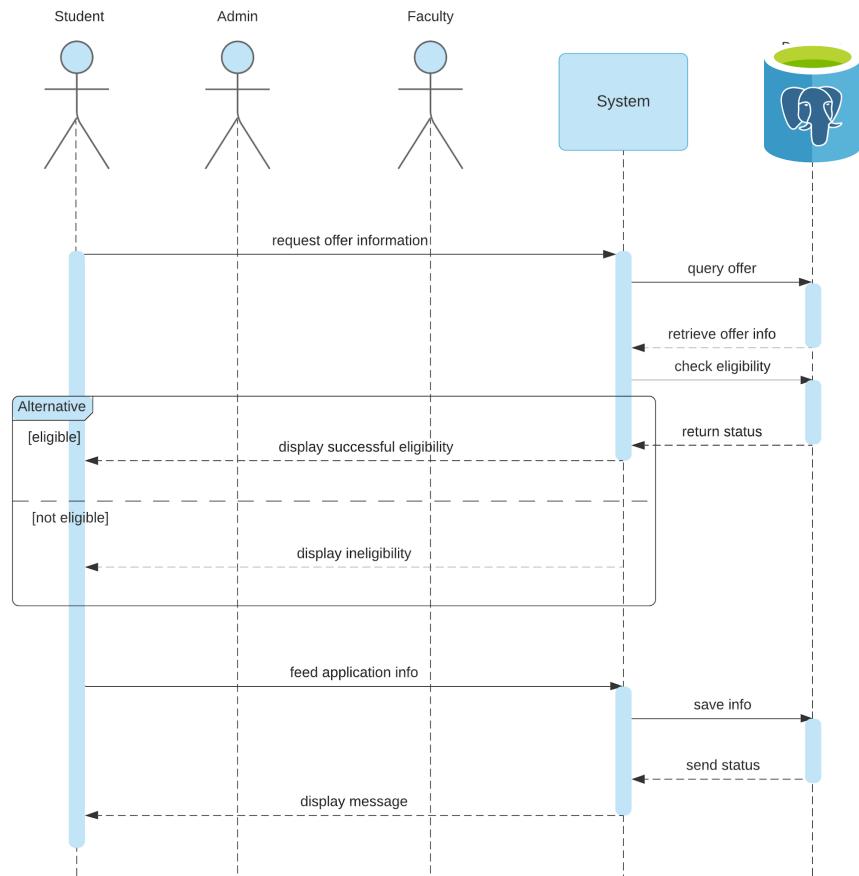
**Fig. 5 Adding Offers Sequence Diagram**

### 3. View Offers



**Fig. 6 Viewing Offers Sequence Diagram**

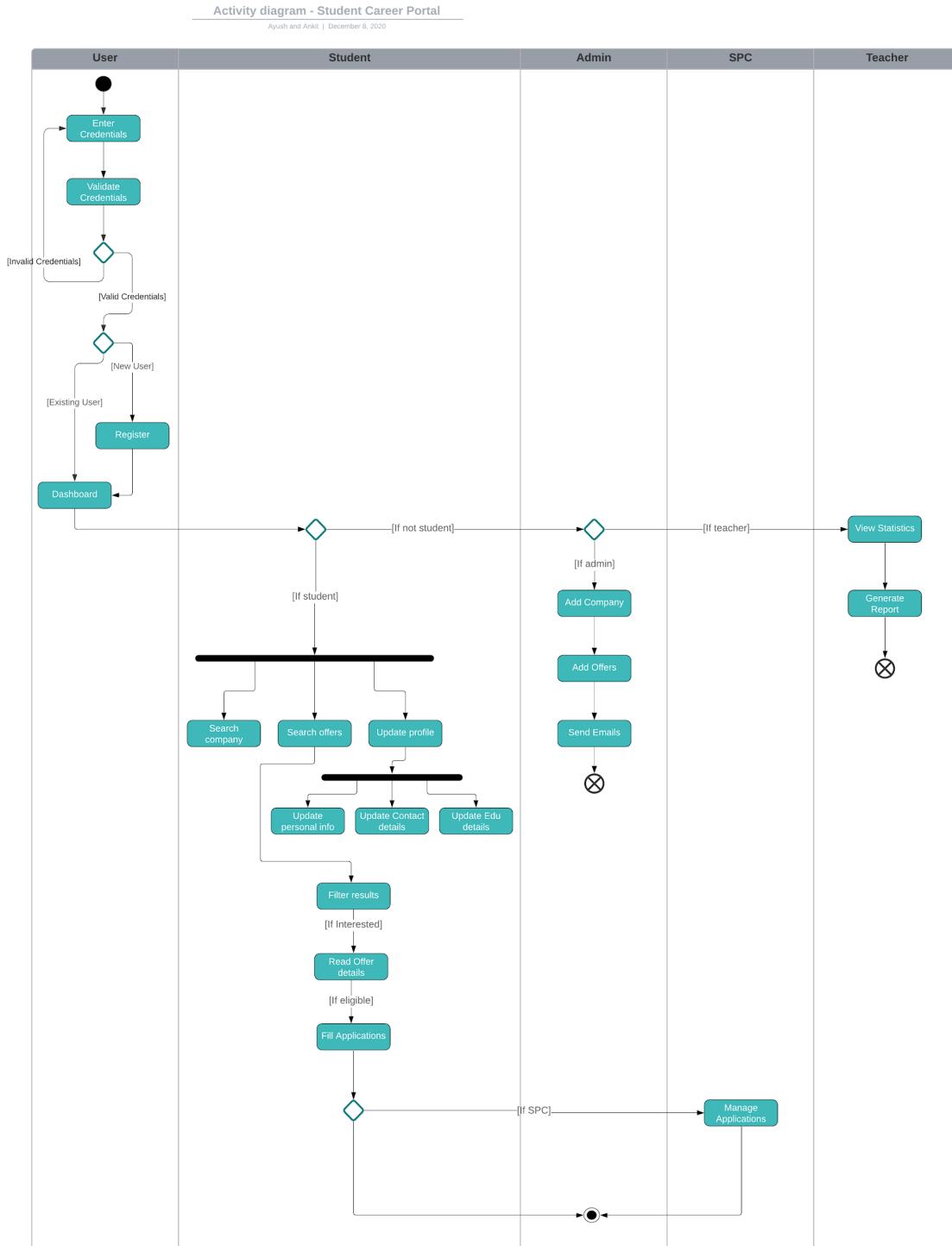
### 4. Fill Application



**Fig. 7 Filling Application Sequence Diagram**

### 3.2.3 Activity Diagram

Activity diagram is defined as a UML diagram that focuses on the execution and flow of the behaviour of a system instead of implementation. It is also called object-oriented flowchart. Activity diagrams consist of activities that are made up of actions which apply to behavioural modelling technology.



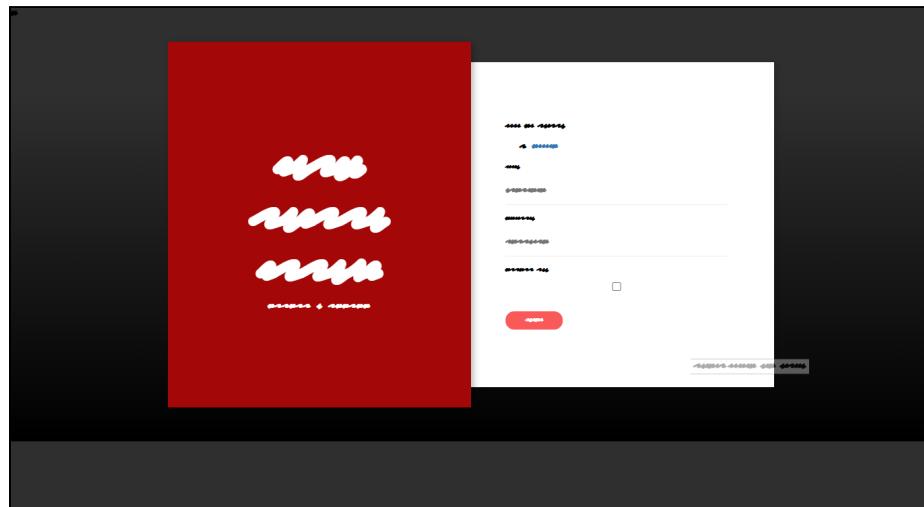
**Fig. 8 Activity Diagram**

### 3.3 Front-End Design

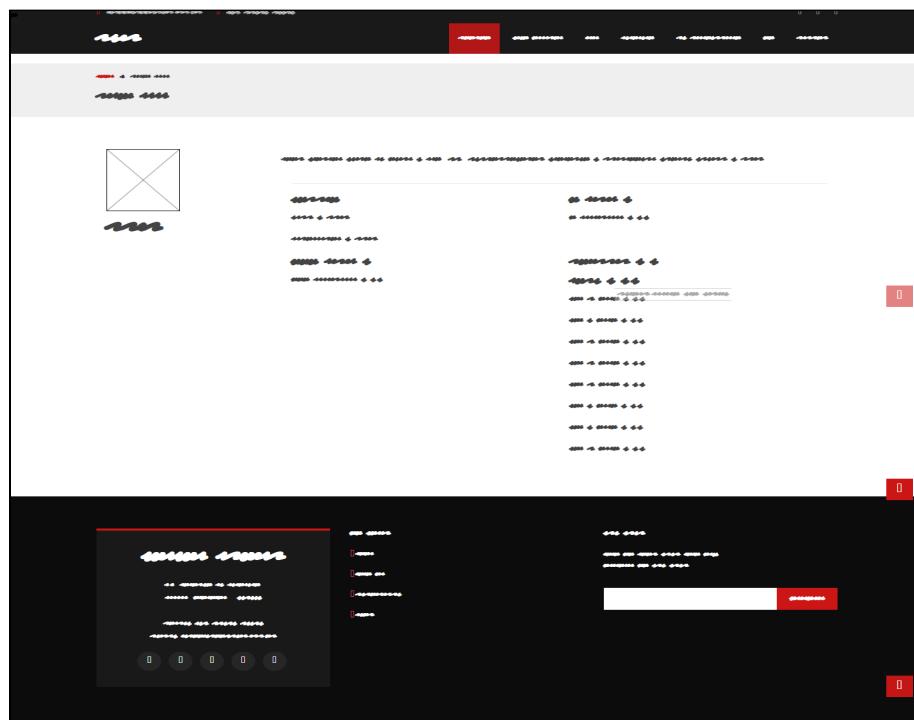
**Wireframing** is a way to design a website service at the structural level. A wireframe is commonly used to lay out content and functionality on a page which takes into account user needs and user journeys. Wireframes are used early in the development process to establish the basic structure of a page before visual design and content is added.

The aim of a wireframe is to provide a visual understanding of a page early in a project to get stakeholder and project team approval before the creative phase gets under way. Following are the wireframes associated with our website for different pages.

#### 1. Login Page



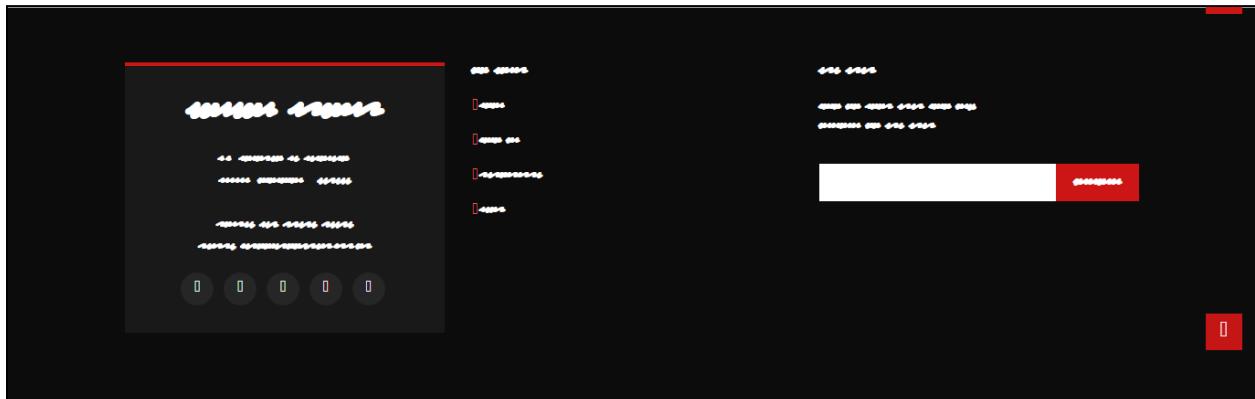
#### 2. Student Dashboard Page



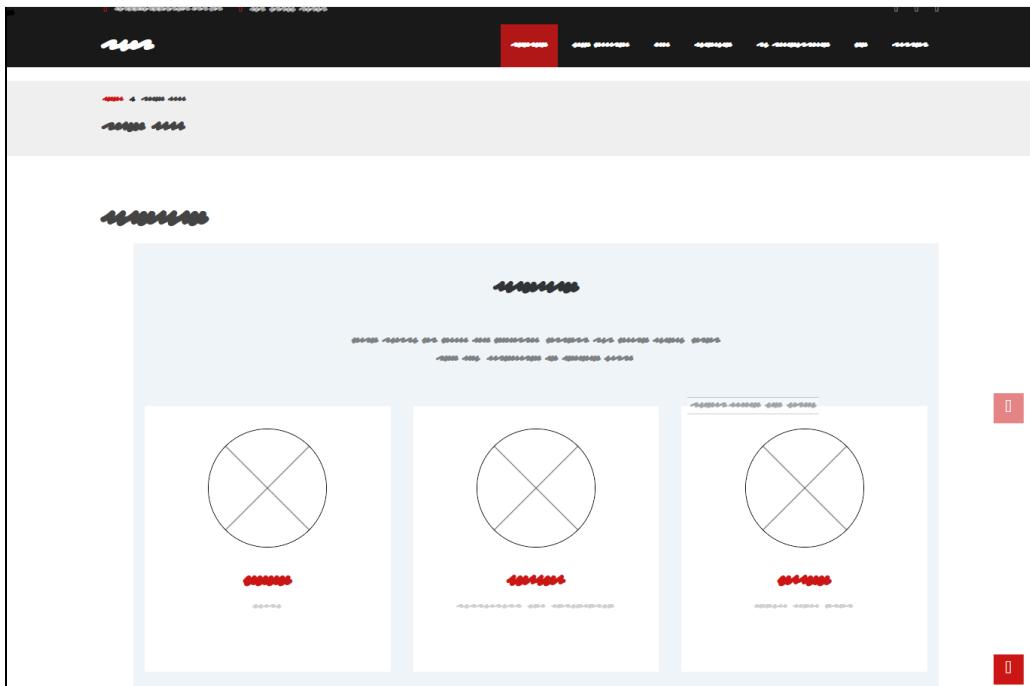
### 3. Landing Page



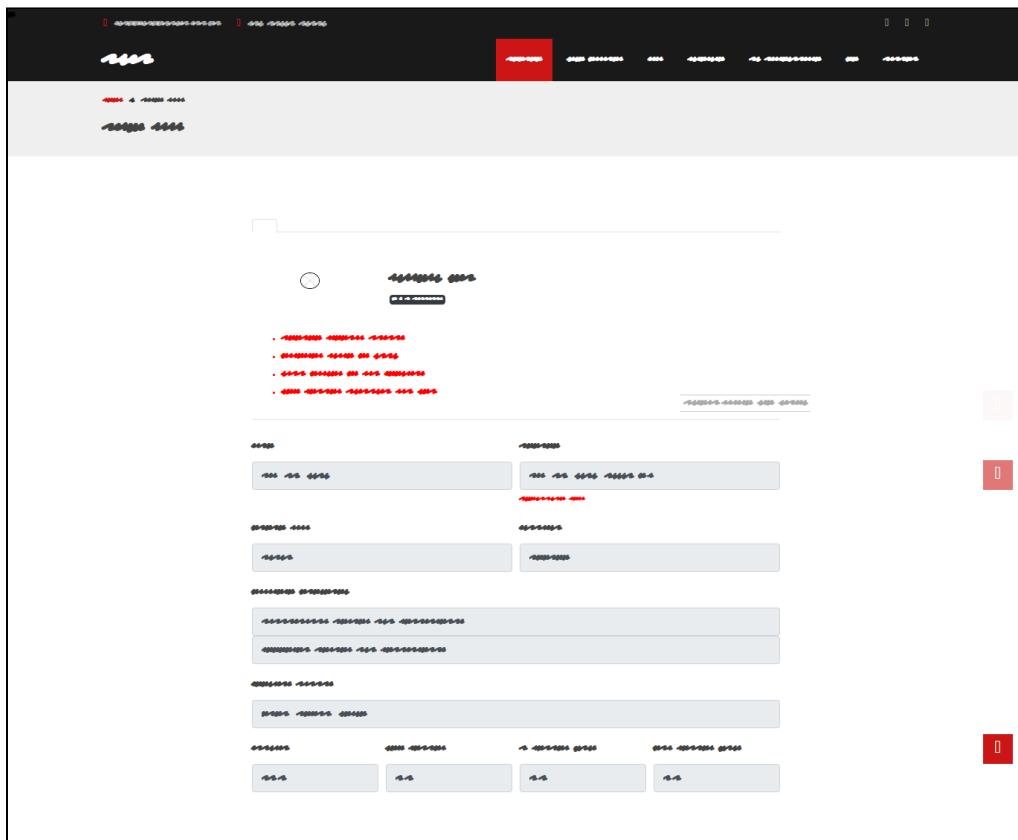
### 4. Footer



## 5. Companies Page



## 6. Offer Details Page



# CHAPTER 4

## Implementation Details

### 4.1 Database Implementation

#### 4.1.1 Table Creation

The Following tables have been created:

app_application	app_faculty
app_company	app_offer
app_contact	app_offer_eligible_branches
app_department	app_spc
app_departmentgroupemail	app_student
app_education	app_user

The creation of tables using SQLite Queries is given below.

```
.....
app_application
.....



CREATE TABLE IF NOT EXISTS "app_application"
(
    "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
    "status" varchar(10) NOT NULL,
    "student_id" integer NOT NULL REFERENCES "app_student" ("id") DEFERRABLE INITIALLY
DEFERRED,
    "offer_id" integer NOT NULL REFERENCES "app_offer" ("id") DEFERRABLE INITIALLY
DEFERRED);

.....
app_company
......



CREATE TABLE IF NOT EXISTS "app_company" ("id" integer NOT NULL PRIMARY KEY
AUTOINCREMENT,
    "name" varchar(100) NOT NULL,
    "description" text NOT NULL,
    "website" varchar(200) NOT NULL,
    "category" varchar(40) NOT NULL,
    "logo" varchar(100) NOT NULL);
```

```

.....
app_contact
.....

CREATE TABLE IF NOT EXISTS "app_contact" ("id" integer NOT NULL PRIMARY KEY
AUTOINCREMENT,
"city" varchar(50) NOT NULL,
"state" varchar(50) NOT NULL,
"pin" integer NULL,
"mobile" varchar(10) NOT NULL,
"alternate_mobile" varchar(10) NOT NULL,
"email" varchar(254) NULL,
"user_id" integer NOT NULL UNIQUE REFERENCES "app_student" ("id") DEFERRABLE
INITIALLY DEFERRED);

.....
app_department
.....

CREATE TABLE IF NOT EXISTS "app_department" ("id" integer NOT NULL PRIMARY KEY
AUTOINCREMENT,
"name" varchar(50) NOT NULL,
"description" text NOT NULL,
"group_email" varchar(254) NOT NULL);

.....
app_departmentgroupemail
......



CREATE TABLE IF NOT EXISTS "app_departmentgroupemail" ("id" integer NOT NULL
PRIMARY KEY AUTOINCREMENT,
"email" varchar(254) NOT NULL UNIQUE,
"graduation_year" integer NOT NULL,
"department_id" integer NOT NULL REFERENCES "app_department" ("id") DEFERRABLE
INITIALLY DEFERRED);

.....
app_education
......



CREATE TABLE IF NOT EXISTS "app_education" ("id" integer NOT NULL PRIMARY KEY
AUTOINCREMENT,
"admission_mode" varchar(10) NOT NULL,
"X_board" varchar(20) NOT NULL,
"XII_board" varchar(20) NOT NULL,
"X_percentage" real NOT NULL,
"XII_percentage" real NOT NULL,
"sem_1_sgpa" real NULL,
"sem_2_sgpa" real NULL,
"sem_3_sgpa" real NULL,

```



```

.....
app_offer_eligible_branches
.....



CREATE TABLE IF NOT EXISTS "app_offer_eligible_branches"
(
    "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
    "offer_id" integer NOT NULL REFERENCES "app_offer" ("id") DEFERRABLE INITIALLY DEFERRED,
    "department_id" integer NOT NULL REFERENCES "app_department" ("id") DEFERRABLE INITIALLY DEFERRED);
.....



app_student
.....



CREATE TABLE IF NOT EXISTS "app_student" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
    "name" varchar(100) NOT NULL,
    "USN" varchar(20) NULL,
    "gender" varchar(10) NOT NULL,
    "date_of_birth" date NOT NULL,
    "resume" varchar(100) NULL,
    "is_eligible" bool NOT NULL,
    "user_id" integer NOT NULL UNIQUE REFERENCES "app_user" ("id") DEFERRABLE INITIALLY DEFERRED,
    "profile_pic" varchar(100) NOT NULL);
.....



app_user
.....



CREATE TABLE IF NOT EXISTS "app_user" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
    "password" varchar(128) NOT NULL,
    "last_login" datetime NULL,
    "is_superuser" bool NOT NULL,
    "username" varchar(150) NOT NULL UNIQUE,
    "first_name" varchar(150) NOT NULL,
    "last_name" varchar(150) NOT NULL,
    "email" varchar(254) NOT NULL,
    "is_staff" bool NOT NULL,
    "is_active" bool NOT NULL,
    "date_joined" datetime NOT NULL,
    "user_type" varchar(10) NOT NULL);

```

#### 4.1.2 Table Population

```
/* Inserting into app_company table */
INSERT INTO app_company VALUES (1, 'SAP', 'Sap labs', 'www.sap.com',
'Software', '' ) ;

/* Inserting into app_department table */
INSERT INTO app_department VALUES (1, 'ISE', 'Information Science and
Engg', 'hodise@evce.edu.in' ) ;

/* Inserting into app_application table */
INSERT INTO app_application VALUES (1, 1, 1, 'PROCESSING' ) ;
```

#### 4.1.3 Query Execution and Output

1. List all the eligible student-offer pairs along with some other info

```
select s.name,app_company.name, offer.id, offer.cgpa_cutoff
from app_student as s , app_offer as offer,app_company ,app_education
as edu
where s.is_eligible
      and s.id = edu.user_id
      and edu.cgpa > offer.cgpa_cutoff
      and offer.company_id = app_company.id ;
```

The screenshot shows a database query results window. At the top, there is a SQL query with numbered lines 1 through 6. Below the query is a table with four columns: name, name, id, and cgpa\_cutoff. The table contains 10 rows of data, each corresponding to a student record from the app\_student table.

	name	name	id	cgpa_cutoff
1	Ayush Kumar	Sabrina Long	1	0.0
2	Ayush Kumar	Alyssa McDonald MD	2	0.0
3	Ayush Kumar	Danielle Jefferson MD	3	0.0
4	Ayush Kumar	Paul Benson	4	0.0
5	Ayush Kumar	Andre Johnson	5	0.0
6	Ayush Kumar	Kristen Johnson	6	0.0
7	Ayush Kumar	Amber Williams	7	0.0
8	Ayush Kumar	Amber Peters	8	0.0
9	Ayush Kumar	SAP LABS	9	0.0
10	Ayush Kumar	Richard Martin	10	9.0

**Fig. 1 Output for Query 1**

2. List the average package offered by companies whose offers have been accepted

```
select avg(offer.package)
from app_offer as offer , app_application
where app_application.status="ACCEPTED" ;
```

```
1 select avg(offer.package)
2 from app_offer as offer , app_application
3 where app_application.status="ACCEPTED" ;
4
5 avg(offer.package)
6 57.0
```

**Fig. 2 Output for Query 2**

3. List all accepted Open-Dream category offers

```
select s.name , app_company.name, offer.offer_type ,appl.status
from app_student as s , app_offer as offer,  app_company,app_application as
appl where appl.status = 'ACCEPTED'
      and appl.offer_id = offer.id
      and offer.offer_type="OPEN"
      and appl.student_id = s.id
      and app_company.id = offer.company_id;
```

```
1 select s.name , app_company.name, offer.offer_type ,appl.status
2 from app_student as s , app_offer as offer,  app_company,app_application as
appl where appl.status = 'ACCEPTED'
3      and appl.offer_id = offer.id
4      and offer.offer_type="OPEN"
5      and appl.student_id = s.id
6      and app_company.id = offer.company_id;
7
8   name    name    offer_type  status
9 1 Zach    Danielle Jefferson MD OPEN    ACCEPTED
10 2 Ayush Kumar SAP LABS           OPEN    ACCEPTED
```

**Fig. 3 Output for Query 3**

4. Populate app\_company table and retrieve the pushed value

```
insert into app_company
values (22, 'SAP', 'Sap labs', 'www.sap.com', 'Software', '') ;
select * from app_company where id=22;
```

```
1 INSERT INTO app_company VALUES (22, 'SAP', 'Sap labs', 'www.sap.com', 'Software', '');
2
3 SELECT * FROM app_company WHERE id=22;
4
5   id  name  description  website  logo  category
6 1 22  SAP  Sap labs  www.sap.com Software
```

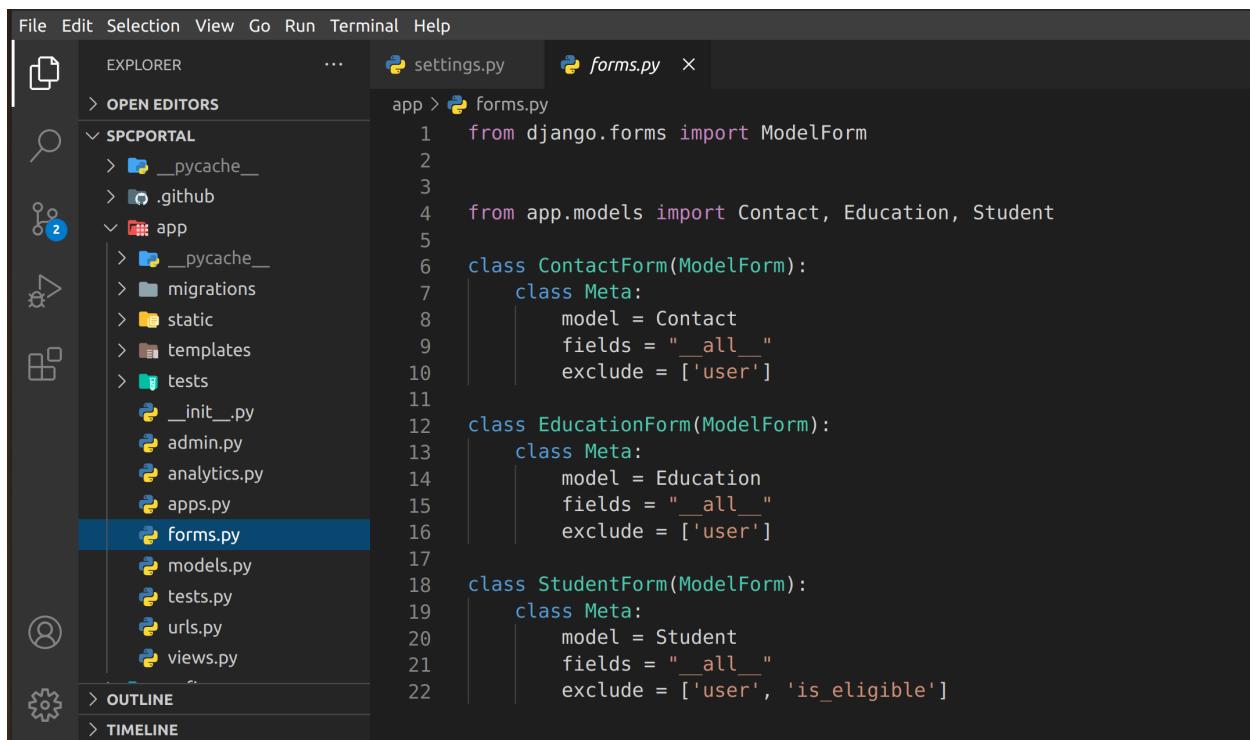
**Fig. 4 Output for Query 4**

## 4.2 Front-End Implementation

### 4.2.1 Form Creation

After navigating to the **Edit Profile** section, three different forms are displayed. In order to complete the profile, a student needs to enter several of his details. These are acquired via a set of forms, dedicated entirely to those details. The constituent of this set are:

- **Student Form** - It asks for basic details for a student like Name, USN, Gender, etc..
- **Contact Form** - It asks for contact details of a student like Contact No., Pin Code, City, State, etc..
- **Education Form** - It asks for educational details like Mode of Admission, Department, Performance in School, College Grades, etc..



The screenshot shows a code editor interface with a dark theme. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. On the left is an Explorer sidebar with icons for file operations like Open Editors, Find, Search, and Outline. The main area displays two files: 'settings.py' and 'forms.py'. The 'forms.py' file is open and contains the following Python code:

```
1  from django.forms import ModelForm
2
3
4  from app.models import Contact, Education, Student
5
6  class ContactForm(ModelForm):
7      class Meta:
8          model = Contact
9          fields = "__all__"
10         exclude = ['user']
11
12 class EducationForm(ModelForm):
13     class Meta:
14         model = Education
15         fields = "__all__"
16         exclude = ['user']
17
18 class StudentForm(ModelForm):
19     class Meta:
20         model = Student
21         fields = "__all__"
22         exclude = ['user', 'is_eligible']
```

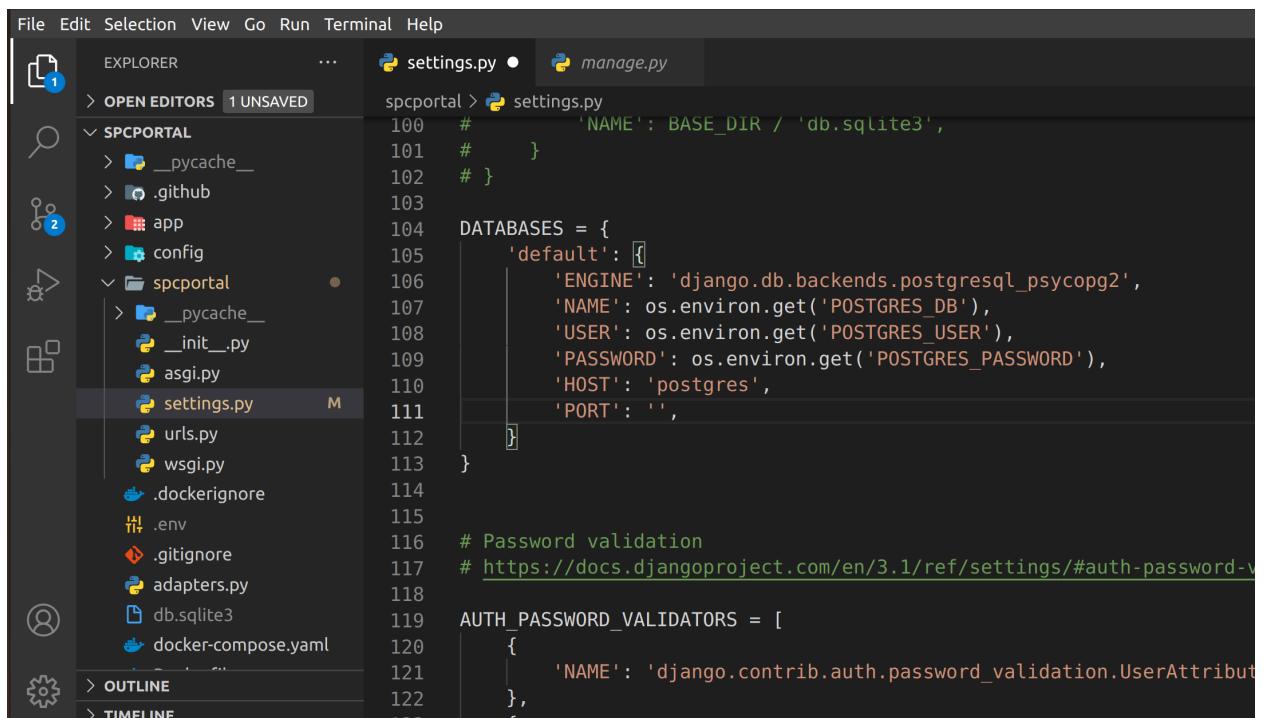
Fig. 1 Implementation of Forms

## 4.2.2 Connectivity with Database

For the portal open source PostgreSQL RDBMS is used. PostgreSQL is a highly stable database management system and a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness. PostgreSQL runs on all major operating systems, including Linux, UNIX and Windows. It supports text, images, sounds, and video, and includes programming interfaces for C / C++, Java, Perl, Python, Ruby, Tcl and Open Database Connectivity (ODBC).

Following are some of the main reasons for using PostgreSQL in the project:

1. Django provides a number of data types that will only work with PostgreSQL.
2. Django has django.contrib.postgres to make database operations on PostgreSQL.
3. If you are building an application with maps or you are storing geographical data, you need to use PostgreSQL, as GeoDjango is only fully compatible with PostgreSQL.
4. PostgreSQL has the richest set of features that are supported by Django.



```
File Edit Selection View Go Run Terminal Help
EXPLORER ... settings.py ● manage.py
> OPEN EDITORS 1 UNSAVED
SPCPORTAL
> __pycache__
> .github
> app
> config
spcportal
> __pycache__
> __init__.py
> asgi.py
> settings.py M
> urls.py
> wsgi.py
.dockerignore
.env
.gitignore
adapters.py
db.sqlite3
docker-compose.yaml
> OUTLINE
> TIMELINE

100 #             'NAME': BASE_DIR / 'db.sqlite3',
101 #         }
102 #
103
104 DATABASES = {
105     'default': [
106         {
107             'ENGINE': 'django.db.backends.postgresql_psycopg2',
108             'NAME': os.environ.get('POSTGRES_DB'),
109             'USER': os.environ.get('POSTGRES_USER'),
110             'PASSWORD': os.environ.get('POSTGRES_PASSWORD'),
111             'HOST': 'postgres',
112             'PORT': '',
113         }
114     ]
115
116     # Password validation
117     # https://docs.djangoproject.com/en/3.1/ref/settings/#auth-password-
118
119 AUTH_PASSWORD_VALIDATORS = [
120     {
121         'NAME': 'django.contrib.auth.password_validation.UserAttribut
122     },
123 ]
```

**Fig. 2 Connectivity with Database**

### 4.2.3 Security Features

This section provides an overview of the system security and requirements that must be followed during implementation.

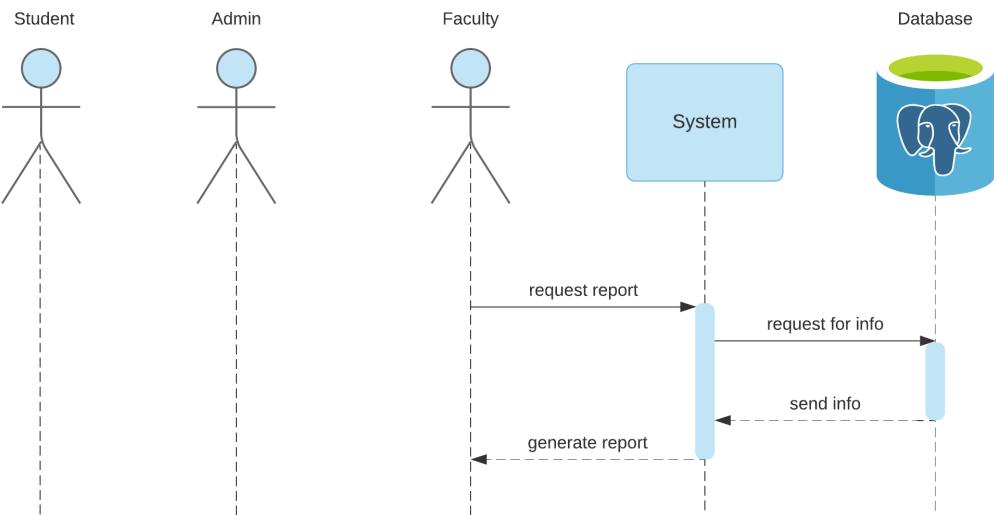
- Users will be authenticated with google Oauth, software has access to users google email address, name, and profile picture.
- User profiles will be shared with 3rd party companies.
- HTTPS secure connection will be established for communication.
- Portal will have Cross-site scripting (XSS) protection. XSS attacks allow a user to inject client-side scripts into the browsers of other users.
- Portal will have Cross-site request forgery (CSRF) protection. CSRF attacks allow a malicious user to execute actions using the credentials of another user without that user's knowledge or consent.
- Portal will have SQL injection protection. SQL injection is a type of attack where a malicious user is able to execute arbitrary SQL code on a database.
- Portal will have Clickjacking protection. Clickjacking is a type of attack where a malicious site wraps another site in a frame.

```
File Edit Selection View Go Run Terminal Help
Python settings.py ×
spcportal > Python settings.py
55 |     'allauth.socialaccount',      # <-- 
56 |     'allauth.socialaccount.providers.google',    # <-- 
57 |
58 ]
59
60 MIDDLEWARE = [
61     'django.middleware.security.SecurityMiddleware',
62     'django.contrib.sessions.middleware.SessionMiddleware',
63     'django.middleware.common.CommonMiddleware',
64     'django.middleware.csrf.CsrfViewMiddleware',
65     'django.contrib.auth.middleware.AuthenticationMiddleware',
66     'django.contrib.messages.middleware.MessageMiddleware',
67     'django.middleware.clickjacking.XFrameOptionsMiddleware',
68 ]
69
70 ROOT_URLCONF = 'spcportal.urls'
71
72 TEMPLATES = [
73     {
```

**Fig. 3 Implementation of security features in Django**

#### 4.2.4 Report Generation

The following sequence shows how the report generation action works.



**Fig. 4 Action: Report Generation**

```
File Edit Selection View Go Run Terminal Help
EXPLORER      ...
OPEN EDITORS
SPCPORTAL
> __pycache__
> .github
app
> __pycache__
> migrations
> static
> templates
> tests
> __init__.py
> admin.py
> analytics.py
> apps.py
> forms.py
> models.py
> tests.py
> urls.py
> views.py
config
spcportal
.dockignore
.env
.gitignore
OUTLINE
TIMELINE

app > analytics.py
10
11
12 def pie_dream_open_offer(*args):
13     a = Offer.objects.all().values('offer_type').annotate(package = Coalesce(Sum('package'), 0))
14     a = list(a)
15     dataset = {}
16
17     dataset['labels'] = list(map(lambda x : x['offer_type'], a))
18     dataset['data'] = list(map(lambda x : x['package'], a))
19     dataset['backgroundColor'] = list(map(color, a))
20     chart = {}
21
22     options = {
23         'title': {
24             'display': 'true',
25             'text': 'Total CTC breakup (in LPA) '
26         }
27     }
28
29     chart = {
30         'type':'pie',
31         'data':{
32             'datasets': [dataset],
33             'labels': dataset['labels']
34         },
35         'options':options
36     }
37
38     return chart
```

The screenshot shows a code editor with Python code in the file 'analytics.py'. The code defines a function 'pie\_dream\_open\_offer' that retrieves data from a database, processes it into a pie chart format, and returns the chart object. The code uses Django's ORM and chart.js library.

**Fig. 5 Report Generation Implementation**

# CHAPTER 5

## Testing and Results

### 5.1 Database Testing

Database Testing is a type of software testing that checks the schema, tables, triggers etc. of the database under test. It involves creating complex queries for performing the load or stress test on the database and checking its responsiveness. It checks integrity and consistency of data.

#### 5.1.1 Test Cases

Test Case Id	Testcase	Expected Output	Actual Output	Pass/Fail
TSFD01	User details are entered into valid database	Username, first name, last name, password hashed with md5 updated	Username, first name, last name, password hashed with md5 updated	Pass
TSFD02	User details are tried to be entered into an invalid database	Error message: Database unavailable	Database does not exist	Pass
TSFD03	Valid user login password is entered	User logs in and a pop-up gets displayed on the top right corner	User logs in and a pop-up is displayed on the top right corner	Pass
TSFD04	Invalid user login password is entered	Invalid username/password	Invalid username/password	Pass

TSFD05	Create User in user table in empty database	User record in database gets created	User record in database is created	Pass
TSFD06	Create User Record	User record not found	User record not found	Pass
TSFD07	Delete User Record	Corresponding Student, contact and education gets deleted	Corresponding Student, contact and education gets deleted	Pass
TSFD08	User creates an offer without a company record	record not created	record not created	Pass
TSFD09	User filters for a non-existing Company	No such company exists	No such company exists	Pass
TSFD010	User searches for entry not present in database	No such item exists	No such item exists	Pass

## 5.2 Front-End Testing

Front-end Testing is testing or verifying the frontend functionality, GUI, and Usability. The main aim of Frontend testing is to make sure that every user is well-protected from bugs. Creating a frontend testing plan helps you to know the devices, browsers, and systems which your project needs to cover.

### 5.2.1 Test Cases

<b>Test Scenario ID</b>	Login 1	<b>Test Case ID</b>	Login-1A
<b>Test case description</b>	testing login feature	<b>Test Priority</b>	High
<b>prerequisites</b>	None	<b>Post requisites</b>	None

Sl. no	Action	Inputs	Expected output	Actual output	test browser	test result
1	Sign in with google using ayushkumar.is18@rvce.edu.in		login success	login success	Firefox	Pass
2	Sign in with google using ayushkumar.is18@gmail.com		Login failed	login failed	Firefox	Pass
3	Sign in with username	username: admin password : admin	Login success	login success	Firefox	Pass
4	Sign in with username	username: admin password : password	Login failed	login failed	Firefox	Pass
5	Click on logout button		Logged out	Logged out	Firefox	Pass
6	Visit /dashboard		redirect to login page	redirect to login page	Firefox	Pass

### 5.2.3 System Testing

System testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system.

<b>Test Scenario ID</b>	Student Features	<b>Test Case ID</b>	Student-1A
<b>Test case description</b>	testing user functionality of student	<b>Test Priority</b>	High
<b>prerequisites</b>	None	<b>Post requisites</b>	None

<b>Sl. no</b>	<b>Action</b>	<b>Expected output</b>	<b>Actual output</b>	<b>test browser</b>	<b>test result</b>
1	Update student profile	Update successful	Update successful	Firefox	Pass
2	Update Contact details	Update successful	Update successful	Firefox	Pass
3	Update Education details	Update successful	Update successful	Firefox	Pass
4	Apply for Eligible Applications	Apply successful	Apply successful	Firefox	Pass
5	Apply for Non Eligible Applications	restricted	restricted	Firefox	Pass
6	Click on Companies tab	display company list	display company list	Firefox	Pass

<b>Test Scenario ID</b>	Admin Features	<b>Test Case ID</b>	Admin-1A
<b>Test case description</b>	testing user functionality of admin	<b>Test Priority</b>	High
<b>prerequisites</b>	None	<b>Post requisites</b>	None

<b>Sl. no</b>	<b>Action</b>	<b>Inputs</b>	<b>Expected output</b>	<b>Actual output</b>	<b>test browser</b>	<b>test result</b>
1	login at /admin	username = admin password = admin	login success	login success	Firefox	Pass
2	Add company details		added company	added company	Firefox	Pass
3	Add offer details		added offer	added offer	Firefox	Pass
4	Add spc role to student		added student role	added student role	Firefox	Pass
5	visit /dashboard		display statistics	display stats	Firefox	Pass
6	click on send email button in offers		send emails	send emails	Firefox	Pass

# CHAPTER 6

## Conclusion

### 6.1 Limitations

Following are the limitations which have been identified in the project so far.

- **Report Generation:** The website does not allow students to generate reports of the statistics.
- **Login:** Login via Google OAuth is only available for students so far, students can't use their personal ID's .
- **Forgotten Password:** If the admin has forgotten his or her password, there is no way to retrieve it, and he/she will lose access to all the data in the portal.
- **Invalid Email-ID:** We have no way of tracking whether or not the email address entered by the user is for an existing student or an alumni.
- **Forms:** The forms collecting the information are not validated for each field. Also, the layout is pretty basic.

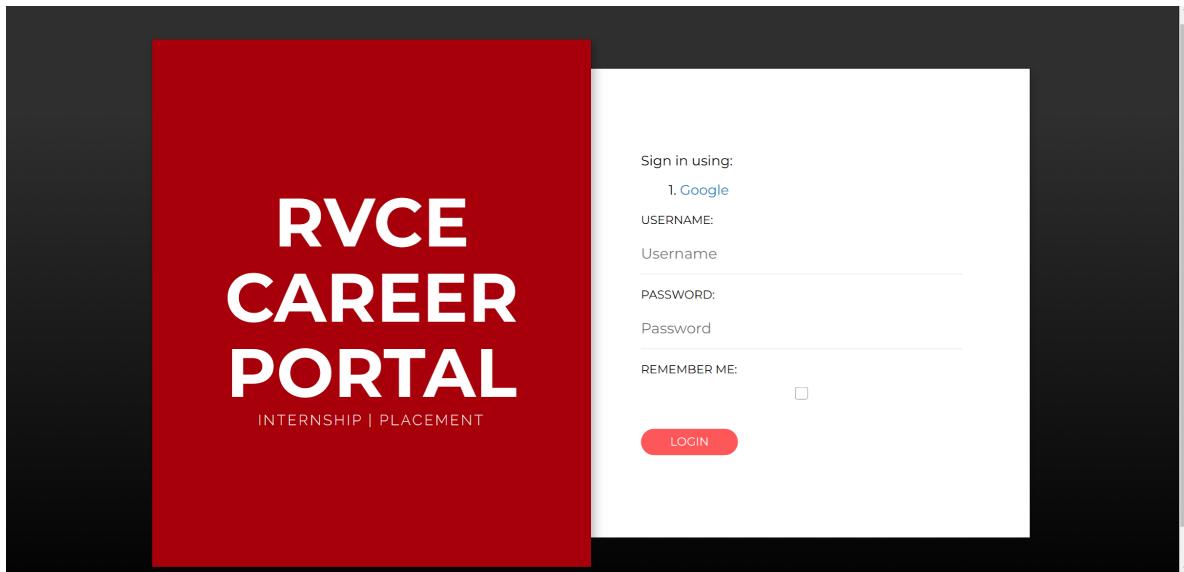
### 6.2 Future Enhancements

Following are some of the points to be looked upon in future.

- **Use of better technology:** The use of better technology in order to understand the likes and dislikes of the users, according to their needs.
- **Report Generation for Students:** Even students will be able to generate reports of past statistics.
- **Better UI:** The use of modern technologies like ReactJS or AngularJS to improve the UI and make it more efficient to use as well as better looks.
- **Usage of API's:** Usage of API's for example Google Calendar to remind students before the exam date or something similar.
- **PWA:** Convert the webpage to PWA that can be installed in phone as cross platform mobile app

## REFERENCES

- [How to use PostgreSQL with Django | EDB](#)
- [Native apps and plugins with our assets](#)
- [What is Front End Testing? Tools & Frameworks](#)
- [Software Testing | Database Testing](#)
- [What is System Testing? Types & Definition with Example](#)
- [Nagios - The Industry Standard In IT Infrastructure Monitoring](#)
- [pytest: helps you write better programs — pytest documentation](#)
- [Docker Compose with NginX, Django, Gunicorn and multiple Postgres databases](#)
- [Deployment checklist](#)
- [Creating and starting a VM instance | Compute Engine Documentation](#)
- [Django documentation | Django documentation](#)



## APPENDIX A- CODE SNIPPETS

A screenshot of a code editor (Visual Studio Code) displaying the file "settings.py" from a Django project named "spcportal". The code defines various settings like BASE\_DIR, SECRET\_KEY, DEBUG, and INSTALLED\_APPS. The code editor shows syntax highlighting for Python and includes a sidebar with file navigation and status bar information.

Code Snippet 1

```

112     # Password validation
113     # https://docs.djangoproject.com/en/3.1/ref/settings/#auth-password-validators
114
115     AUTH_PASSWORD_VALIDATORS = [
116         {
117             'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
118         },
119         {
120             'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
121         },
122         {
123             'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
124         },
125         {
126             'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
127         },
128     ]
129
130
131     # Internationalization
132     # https://docs.djangoproject.com/en/3.1/topics/i18n/
133
134     LANGUAGE_CODE = 'en-us'
135
135     TIME_ZONE = 'Asia/Kolkata'
136
137     USE_I18N = True
138
139     USE_L10N = True
140
141     USE_TZ = True
142
143
144     CRISPY_TEMPLATE_PACK = 'bootstrap4'
145

```

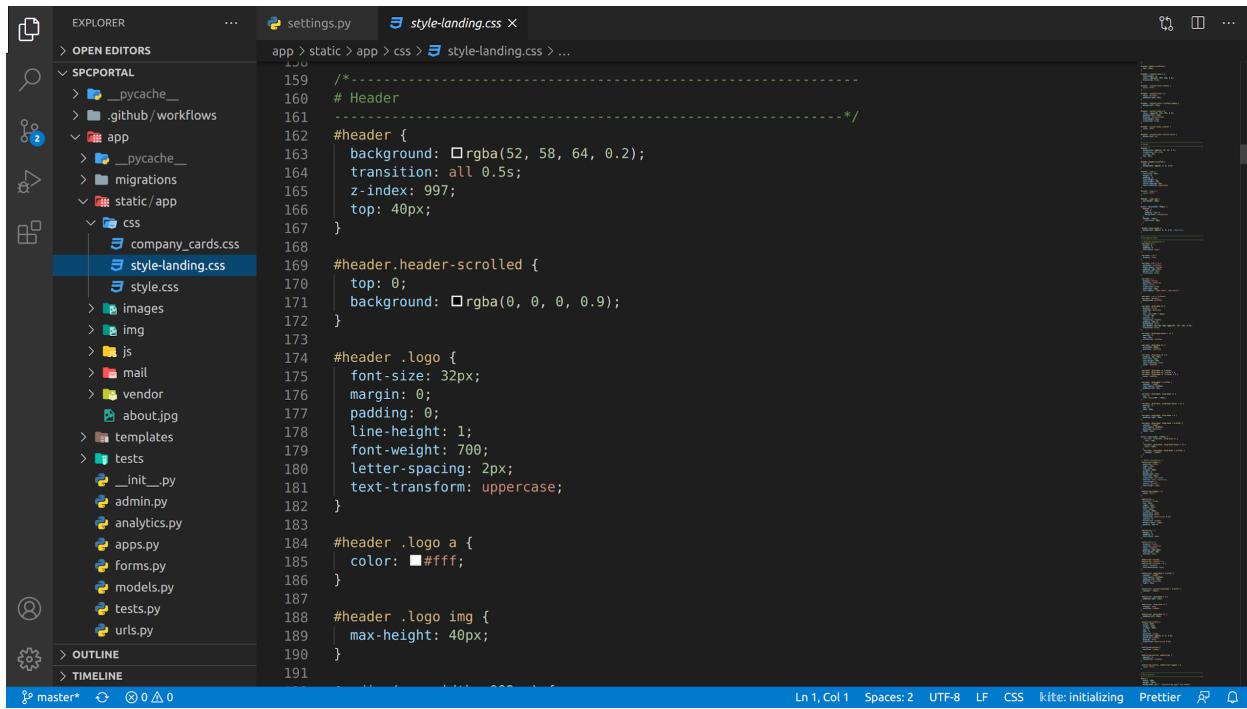
**Code Snippet 2**

```

184     PWA_APP_NAME = 'My App'
185     PWA_APP_DESCRIPTION = "My app description"
186     PWA_APP_THEME_COLOR = '#0A0302'
187     PWA_APP_BACKGROUND_COLOR = '#ffffff'
188     PWA_APP_DISPLAY = 'fullscreen'
189     PWA_APP_SCOPE = '/'
190     PWA_APP_ORIENTATION = 'any'
191     PWA_APP_START_URL = '/'
192     PWA_APP_STATUS_BAR_COLOR = 'default'
193     PWA_APP_ICONS = [
194         {
195             'src': '/static/images/my_app_icon.png',
196             'sizes': '160x160'
197         }
198     ]
199     PWA_APP_ICONS_APPLE = [
200         {
201             'src': '/static/images/my_apple_icon.png',
202             'sizes': '160x160'
203         }
204     ]
205     PWA_APP_SPLASH_SCREEN = [
206         {
207             'src': '/static/images/icons/splash-640x1136.png',
208             'media': '(device-width: 320px) and (device-height: 568px) and (-webkit-device-pixel-ratio: 2)'
209         }
210     ]
211     PWA_APP_DIR = 'ltr'
212     PWA_APP_LANG = 'en-US'
213
214     EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
215     # EMAIL_BACKEND = 'django.core.mail.backends.console.EmailBackend'
216
217

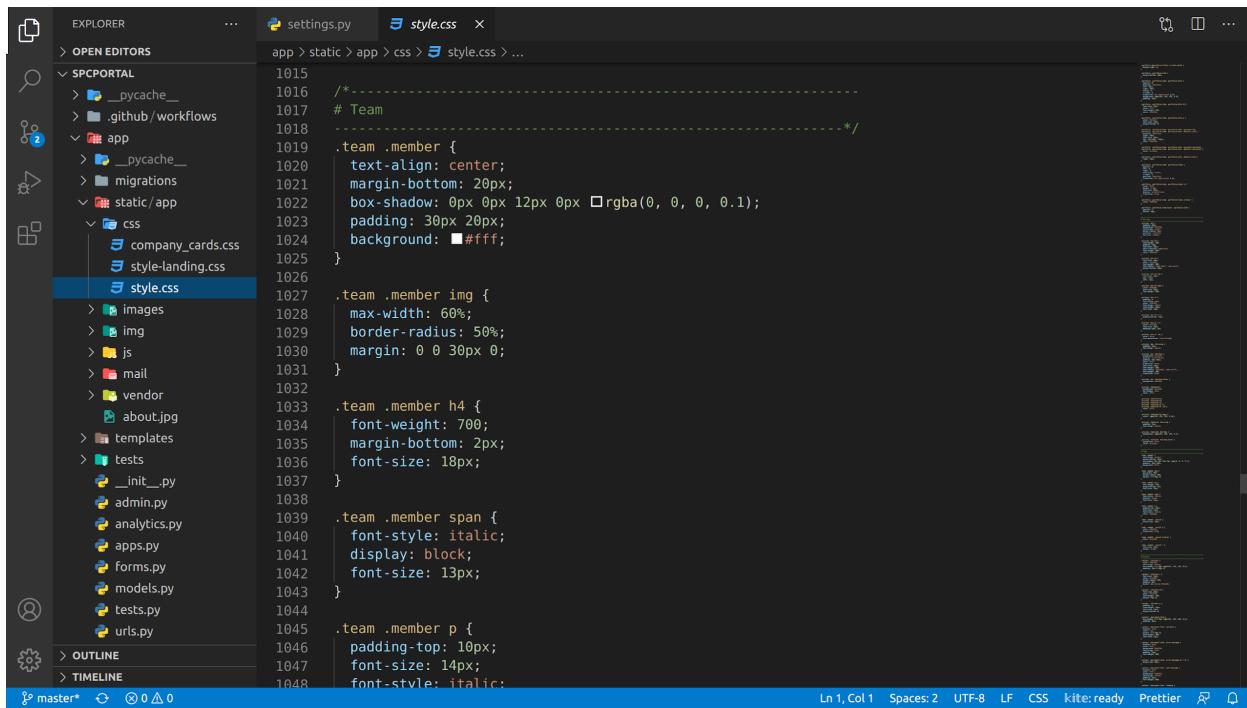
```

**Code Snippet 3**



```
app > static > app > css > style-landing.css > ...
159  /*
160  * Header
161  */
162 #header {
163   background: #rgba(52, 58, 64, 0.2);
164   transition: all 0.5s;
165   z-index: 997;
166   top: 40px;
167 }
168 #header.header-scrolled {
169   top: 0;
170   background: #rgba(0, 0, 0, 0.9);
171 }
172 #header .logo {
173   font-size: 32px;
174   margin: 0;
175   padding: 0;
176   line-height: 1;
177   font-weight: 700;
178   letter-spacing: 2px;
179   text-transform: uppercase;
180 }
181 #header .logo a {
182   color: #fff;
183 }
184 #header .logo img {
185   max-height: 40px;
186 }
187
188
189
190
191
```

Code Snippet 4



```
app > static > app > css > style.css > ...
1015 /*
1016 * Team
1017 */
1018 .team .member {
1019   text-align: center;
1020   margin-bottom: 20px;
1021   box-shadow: 0px 0px 12px 0px #rgba(0, 0, 0, 0.1);
1022   padding: 30px 20px;
1023   background: #fff;
1024 }
1025
1026 .team .member img {
1027   max-width: 60%;
1028   border-radius: 50%;
1029   margin: 0 0 30px 0;
1030 }
1031
1032 .team .member h4 {
1033   font-weight: 700;
1034   margin-bottom: 2px;
1035   font-size: 18px;
1036 }
1037
1038 .team .member span {
1039   font-style: italic;
1040   display: block;
1041   font-size: 13px;
1042 }
1043
1044 .team .member p {
1045   padding-top: 10px;
1046   font-size: 14px;
1047   font-style: italic;
1048 }
```

Code Snippet 5

```

1  !(function($) {
2    "use strict";
3
4    // Preloader
5    $(window).on('load', function() {
6      if ($('#preloader').length) {
7        $('#preloader').delay(100).fadeOut('slow', function() {
8          $(this).remove();
9        });
10     });
11   });
12
13 // Smooth scroll for the navigation menu and links with .scrollto classes
14 var scrolltoOffset = $('#header').outerHeight() - 1;
15 $(document).on('click', '.nav-menu a, .mobile-nav a, .scrollto', function(e) {
16   if (location.pathname.replace(/^\//, '') == this.pathname.replace(/^\//, '') & location.hostname
17     var target = $(this.hash);
18     if (target.length) {
19       e.preventDefault();
20
21       var scrollto = target.offset().top - scrolltoOffset;
22
23       if ($(this).attr("href") == '#header') {
24         scrollto = 0;
25       }
26
27       $('html, body').animate({
28         scrollTop: scrollto
29       }, 1500, 'easeInOutExpo');
30
31       if ($(this).parents('.nav-menu, .mobile-nav').length) {
32         $('.nav-menu .active, .mobile-nav .active').removeClass('active');
33         $(this).closest('li').addClass('active');
34     }

```

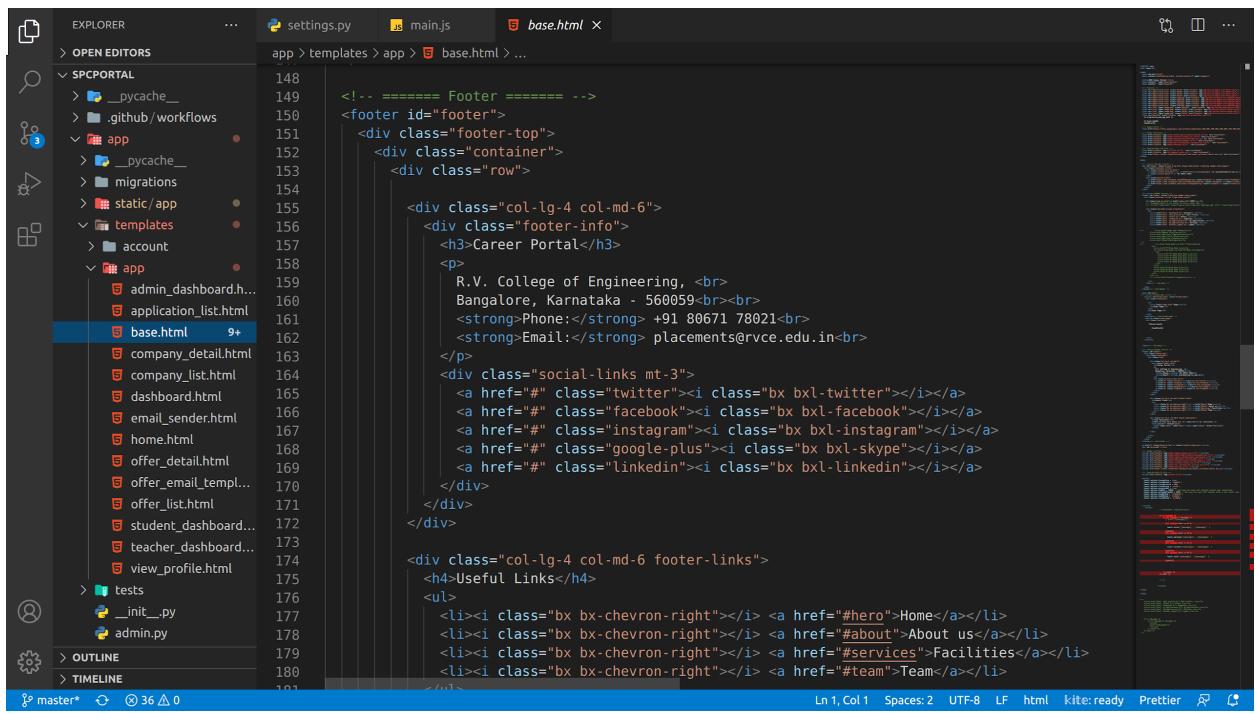
**Code Snippet 6**

```

1  [% load pwa %]
2  {% load static %}
3
4  <!DOCTYPE html>
5  <html lang="en">
6
7  <head>
8    <meta charset="utf-8">
9    <meta content="width=device-width, initial-scale=1.0" name="viewport">
10
11   <title>RVCE Career Portal</title>
12   <meta content="" name="description">
13   <meta content="" name="keywords">
14
15   <!-- Favicons -->
16   <link rel="apple-touch-icon" sizes="57x57" href="{%static "app/img/favicon/apple-icon-57x57.png" %}>
17   <link rel="apple-touch-icon" sizes="60x60" href="{%static "app/img/favicon/apple-icon-60x60.png" %}>
18   <link rel="apple-touch-icon" sizes="72x72" href="{%static "app/img/favicon/apple-icon-72x72.png" %}>
19   <link rel="apple-touch-icon" sizes="76x76" href="{%static "app/img/favicon/apple-icon-76x76.png" %}>
20   <link rel="apple-touch-icon" sizes="114x114" href="{%static "app/img/favicon/apple-icon-114x114.png" %}>
21   <link rel="apple-touch-icon" sizes="120x120" href="{%static "app/img/favicon/apple-icon-120x120.png" %}>
22   <link rel="apple-touch-icon" sizes="144x144" href="{%static "app/img/favicon/apple-icon-144x144.png" %}>
23   <link rel="apple-touch-icon" sizes="152x152" href="{%static "app/img/favicon/apple-icon-152x152.png" %}>
24   <link rel="apple-touch-icon" sizes="180x180" href="{%static "app/img/favicon/apple-icon-180x180.png" %}>
25   <link rel="icon" type="image/png" sizes="192x192" href="{%static "app/img/favicon/android-icon-192x192.png" %}>
26   <link rel="icon" type="image/png" sizes="32x32" href="{%static "app/img/favicon/favicon-32x32.png" %}>
27   <link rel="icon" type="image/png" sizes="96x96" href="{%static "app/img/favicon/favicon-96x96.png" %}>
28   <link rel="icon" type="image/png" sizes="16x16" href="{%static "app/img/favicon/favicon-16x16.png" %}>
29   <link rel="manifest" href="{%static "app/img/favicon/manifest.json" %}">
30
31   {% progressive_web_app_meta %}
32
33   {% block head%}
34   {%endblock%}

```

**Code Snippet 7**



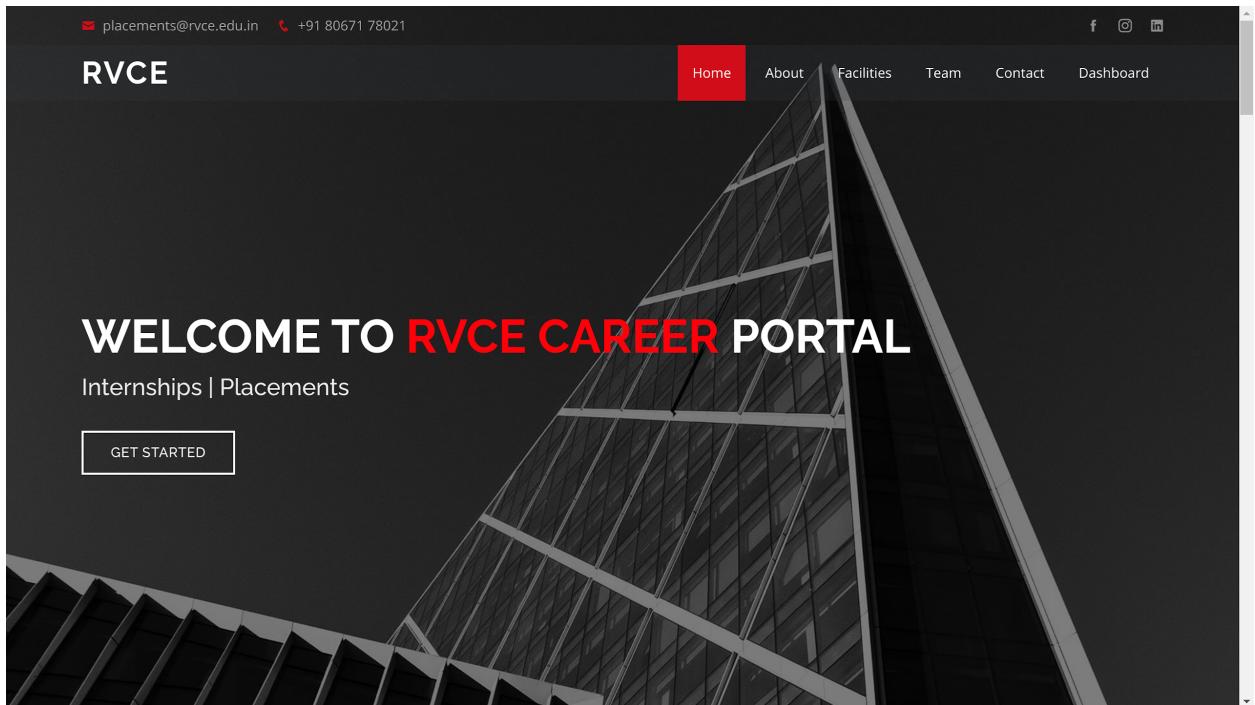
```

148      <!-- ===== Footer ===== -->
149      <footer id="footer">
150          <div class="footer-top">
151              <div class="container">
152                  <div class="row">
153
154          <div class="col-lg-4 col-md-6">
155              <div class="footer-info">
156                  <h3>Career Portal</h3>
157                  <p>
158                      R.V. College of Engineering, <br>
159                      Bangalore, Karnataka - 560059<br><br>
160                      <strong>Phone:</strong> +91 80671 78021<br>
161                      <strong>Email:</strong> placements@rvce.edu.in<br>
162
163                  </p>
164                  <div class="social-links mt-3">
165                      <a href="#" class="bx bxl-twitter"><i class="bx bxl-twitter"></i></a>
166                      <a href="#" class="bx bxl-facebook"><i class="bx bxl-facebook"></i></a>
167                      <a href="#" class="bx bxl-instagram"><i class="bx bxl-instagram"></i></a>
168                      <a href="#" class="bx bxl-google-plus"><i class="bx bxl-google-plus"></i></a>
169                      <a href="#" class="bx bxl-skype"><i class="bx bxl-skype"></i></a>
170                      <a href="#" class="bx bxl-linkedin"><i class="bx bxl-linkedin"></i></a>
171
172                  </div>
173
174          <div class="col-lg-4 col-md-6 footer-links">
175              <h4>Useful Links</h4>
176              <ul>
177                  <li><i class="bx bx-chevron-right"></i> <a href="#hero">Home</a></li>
178                  <li><i class="bx bx-chevron-right"></i> <a href="#about">About us</a></li>
179                  <li><i class="bx bx-chevron-right"></i> <a href="#services">Facilities</a></li>
180                  <li><i class="bx bx-chevron-right"></i> <a href="#team">Team</a></li>
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201

```

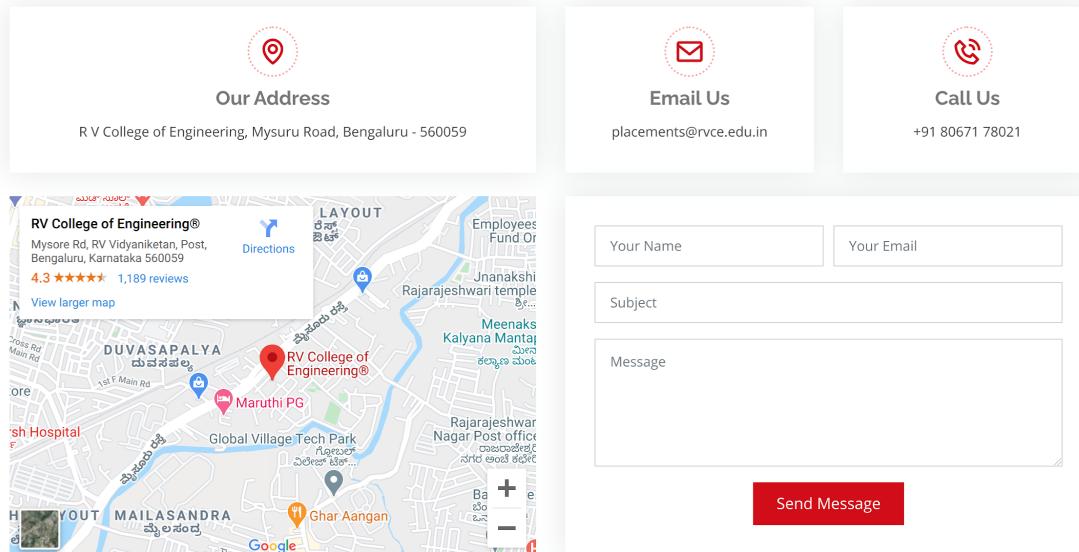
**Code Snippet 8**

## APPENDIX B –SCREENSHOTS



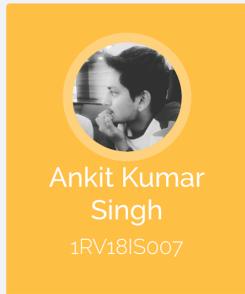
**ScreenShot 1**

## CONTACT



ScreenShot 2

ScreenShot 3



*Just an ordinary guy..*

Name	:	Ankit Kumar Singh	Gender	:	MALE
Date of Birth	:	June 11, 1999	City	:	Jaipur
State	:	Rajasthan	Pincode	:	302017
mobile	:	8769448546	Phone	:	8741993361
Email	:	ankitkumars.is18@rvce.edu.in			

**COMEDK**  
Rank : 459  
Department : Information Science and Engineering

X Board : CBSE  
X Percentage : 95.0

XII Board : CBSE  
XII Percentage : 93.6

Backlogs : 0  
CGPA : 9.22

#### ScreenShot 4

#### STUDENT FORM

Name\*

Ankit Kumar Singh

USN\*

1RV18IS007

Gender

MALE

Date of birth\*

1999-06-11

Profile pic\*

Currently: profile\_pic/ankitz007.jpeg  
Change:

No file chosen

Resume

Currently: resume/resume\_ankit.pdf  Clear

Change:

#### CONTACT FORM

City

Jaipur

State

Rajasthan

Pin

302017

Mobile

8769448546

Alternate mobile

8741993361

Email

ankitkumars.is18@rvce.edu.in

#### EDUCATION FORM

Bio

Just an ordinary guy..

Sem\*

5

Admission mode\*

COMEDK

Admission mode rank

#### ScreenShot 5

**RVCE**

Dashboard Edit Profile Offers Companies My Applications

⚠️ Not eligible.  
Not eligible.

Akamai Technologies



**Akamai Technologies**  
Akamai Technologies, Inc. is a global content delivery network, cybersecurity, and cloud service company, providing web and Internet security services. Akamai's Intelligent Edge Platform is one of the world's largest distributed computing platforms.

- Already Applied
- Your branch is not eligible.
- CGPA Cutoff requirement not met.
- X percentage Cutoff requirement not met.
- XII / Diploma percentage Cutoff requirement not met.
- your Gender is not eligible

Date	Deadline
Dec. 7, 2020	Dec. 31, 2020, 12:43 p.m.

3 weeks, 1 day left

Offer Type	Category
DREAM	INTERNSHIP

Eligible Branches

**ScreenShot 6**

**RVCE**

Dashboard Edit Profile Offers Companies My Applications Settings Logout



**Akamai**  
PRODUCTION AND  
MANUFACTURING



**OPTUM**  
Good for the system.



**VISA**  
R & D MEDIA AND MOBILE  
APPLICATIONS



**Siemens**  
Ingenuity for life.



**Becton Dickinson**

## ScreenShot 7

The screenshot shows the RVCE Placement Portal. At the top, there is a navigation bar with links for Dashboard, Edit Profile, Offers, Companies, My Applications, Settings, and Logout. Below the navigation bar, the page title is "Inner Page". A search bar and filter options ("Eligible Only", "Dream", "Open Dream") are present. The main content area is titled "Offers" and displays two job listings:

- Samsung R&D**  
Intuit Inc. is an American business and financial software company that develops and sells financial... BBBootstrap Team on 1 Nov , 2018  
Dec. 27, 2020 | DREAM | INTERNSHIP |
- Goldman Sachs**  
BBBootstrap Team on 1 Nov , 2018  
Dec. 16, 2020 | OPEN | INTERNSHIP |

## ScreenShot 8