

Summer Project Report on

“STARKONNECT”

By

Group No:5

Pratik Bhosale (5019106)

Mikhael Uzagare (5019139)

Madhur Thakkar (5019156)

Sam Thomas (5019164)



Department of Information Technology

Fr. Conceicao Rodrigues Institute of Technology

Sector 9A, Vashi, Navi Mumbai – 400703

University of Mumbai

2021-2022

Abstract

In this competitive era, the education among the people is so increasing that the jobs for them are now decreasing. The start-ups even want the people who are best in their fields. It becomes difficult to find the people who are intelligent enough to be hired. The work for the start-ups also increases to find the people who can fulfill their requirements. Thinking about these problems, one can think about the process which can handle this process and make the work less complex.

Job Portal is the solution where recruiter as well as the job seeker meet aiming at fulfilling their individual requirement. Therefore, we have developed an interactive website to connect start-ups with freelancers.

The following modules would be implemented:

1. Login/Sign Up
2. Profile page.
3. Jobs:
 - Jobs Applied
 - Create a job
 - Jobs posted

The Graphical User Interface (GUI) would be simple and user-friendly.

We aim to do our bit by facilitating a user-friendly website that will help start-ups by saving time and resources for hiring freelancers.

TABLE OF CONTENTS

Sr. No.	Topic	Page No.
1	Introduction 1.1 Background 1.2 Motivation/ need/ purpose 1.3 Problem Definition 1.4 Scope 1.5 Proposed System Features 1.6 Objectives 1.7 Issues /Limitations	1
2	Literature Survey 2.1 Existing System	4
3	System Design 3.1 Architecture Diagram	7
4	System Requirement 4.1 Hardware 4.2 Software	8

Sr. No.	Topic	Page No.
5	Implementation Details 5.1 User Interface 5.2 Database Details	9
6	Experimental Results	19
7	Conclusion/Future Scope 7.1 Conclusion 7.2 Future Scope	20
8	Appendix: Code Sample	21
9	References	30
10	Acknowledgement	31

LIST OF FIGURES

Fig. No.	Figure Name	Page No.
Fig 3.1	Architectural Design	7
Fig 5.1	Sign-up Page	9
Fig 5.2	Login Page	9
Fig 5.3	OTP Verification	10
Fig 5.4	Select Account Type	10
Fig 5.5	Freelancer Home Page	11
Fig 5.6	Start-up Home Page	11
Fig 5.7	Freelancer Successfully Applied to an opening	12
Fig 5.8	Freelancer My Applications Page	12
Fig 5.9	Freelancer Profile Creation	13
Fig 5.10	Freelancer Profile Page	13
Fig 5.11	Start-up Profile Creation	14
Fig 5.12	Start-up Profile Page	14
Fig 5.13	Start-up Create Job Page	14
Fig 5.14	Start-up Posted Jobs Page	15
Fig 5.21	Database Structure	15
Fig 5.22	users table	16
Fig 5.23	usersprofiles table	16

Fig 5.24	startup_profiles table	17
Fig 5.25	jobs_posted table	17
Fig 5.26	jobs_applied table	18

LIST OF TABLES

Table. No.	Table Name	Page No.
Table 1	Existing Systems	6

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

1.1 BACKGROUND

The importance of placement system is increasing day by day. Thousands of applicants are dependent upon the traditional placement system. But the applicants are facing so many problems. This project is an attempt to minimize the problems of an applicant to find a correct job. If we can make it an online computerized and a mobile alert process it will be so fast and convenient for the applicants. Far from upsetting matters, however, new technologies have predominantly made life better and considerably easier for most people, and so it must be concluded that they should be welcomed. Software does not require a dedicated computer and can be run on almost any PC running Windows.

1.2 MOTIVATION / NEED / PURPOSE

The purpose of developing an Online Job Search Portal comes from the idea to make the job search efficient and handy. It helps the recruiters as a primary source of talent search. It also helps the job seekers to search for current vacancies at a single point. Therefore, we can say that Online Job Search Portal act as a bridge of communication between organizations and applicants. With the evolution of technology and internet being the main source of information for the applicants, these job portals and have become an excellent method to reach wide range of audience.

1.3 PROBLEM DEFINITION

To design a user-friendly website to simplify the hiring process for Startups and Freelancers.

1.4 SCOPE

- In this competitive era, the education among the people is so increasing that the jobs for them are now decreasing.
- The start-ups even want the people who are best in their fields.
- Our website will provide two profile types, freelancer and start-ups.
- It will simplify the hiring process for start-ups and provide increased opportunities for freelancers.

1.5 PROPOSED SYSTEM FEATURES

Most of the job portals lists all the major companies in all major domains, what this does is these being big companies attract more students/freelancers but they hire negligible number of candidates and due to the presence of such big companies, small and emerging start-ups often get neglected by users. Therefore, we have designed a website specifically for start-ups and freelancers.

Our website provides two profile options freelancers and startups. Startups can create their profile page and view their profile at 'My profile' and they can post job openings through the 'Create jobs' tab. Start-ups can also view the jobs they have posted and the freelancers who have applied for the jobs through from the 'Jobs Posted' tab. Freelancers can also create their profile page and can view their profile at 'My profile' and can also apply for the job

openings posted by the startups through their homepage where the jobs of requirements of various start-ups will be listed. The applied jobs can be viewed through 'My Applications' tab. The website has simple and easy to use user interface

1.6 OBJECTIVES

1. To connect Startups and Freelancers.
2. Simplify internship Recruitment process.
3. To help Startups grow by providing them exposure.
4. To help freelancers keep track of positions they have applied to.

1.7 ISSUES/LIMITATIONS

1. Designed specifically for startups and freelancers.
2. No feature for messaging.
3. No feature for posting articles, videos, success stories.
4. Password is not Encrypted.
5. Data is not Validated.

CHAPTER 2

LITERATURE SURVEY

2. LITERATURE SURVEY

2.1 EXISTING SYSTEM

[1] LinkedIn

LinkedIn is the world's largest professional network on the internet. You can use LinkedIn to find the right job or internship, connect and strengthen professional relationships, and learn the skills you need to succeed in your career. You can access LinkedIn from a desktop, LinkedIn mobile app, mobile web experience, or the LinkedIn Lite Android mobile app.

A complete LinkedIn profile can help you connect with opportunities by showcasing your unique professional story through experience, skills, and education. You can also use LinkedIn to organize offline events, join groups, write articles, post photos and videos, and more. But LinkedIn can lead to a large number of spam messages and professional accounts are costly.

[2] Internshala

Internshala is a dot com business with the heart of dot org. It is a technology company on a mission to equip students with relevant skills & practical exposure through internships and online trainings. Internshala boasts a simple yet interactive User Interface which makes the task even easier.

They also provide various features like resume-builder and online trainings for applicants on various topics. The main thing about this website is that it lists a lot of startups providing stipend and non-stipend internship offers of all fields.

[3] MonsterIndia

Monster India is India's leading online career and recruitment resource with its cutting-edge technology provides relevant profiles to employers and relevant jobs to jobseekers across industry verticals, experience levels and geographies.

Monster provides the widest and most sophisticated job seeking, career management, recruitment and talent management capabilities globally.

Existing System	LinkedIn	InternShala	MonsterIndia	StarKconnect
Description	LinkedIn is an online platform that connects the world's professionals. A complete LinkedIn profile will summarize your professional experience to your connections, current and future employers, and recruiters.	Internshala is a technology company on a mission to equip students with relevant skills & practical exposure through internships and online trainings	Monster India is India's leading online career and recruitment resource, provides relevant profiles to employers and relevant jobs to jobseekers across industry verticals, experience levels and geographies. Monster provides the widest and most sophisticated job seeking, career management, recruitment and talent management capabilities globally.	StarKconnect is a virtual platform specifically designed to connect startups and freelancers so that startups can hire freelancers for various projects.
Advantages	<ol style="list-style-type: none"> 1. Easy to connect with companies of all domain. 2. Contains feature for messaging, posting articles, videos, etc. 	<ol style="list-style-type: none"> 1. Simple and attractive UI. 2. Contains feature for messaging. 	<ol style="list-style-type: none"> 1. Simple and attractive UI. 2. Easy to connect with companies of all domain. 	<ol style="list-style-type: none"> 1. Easy to connect with startups. 2. Easy to create and apply for jobs 3. Easy to keep track of jobs applied and created
Disadvantages	<ol style="list-style-type: none"> 1. Lists major companies so start-ups get neglected. 2. Too much spam messages. 	<ol style="list-style-type: none"> 1. Does not provide profile page for companies. 2. Too much spam messages. 	<ol style="list-style-type: none"> 1. Does not provide profile page for companies or Freelancers. 2. No feature for messaging or for posting 	<ol style="list-style-type: none"> 1. Designed specifically for start-ups and freelancers. 2. No feature for messaging or for posting articles, videos, success stories

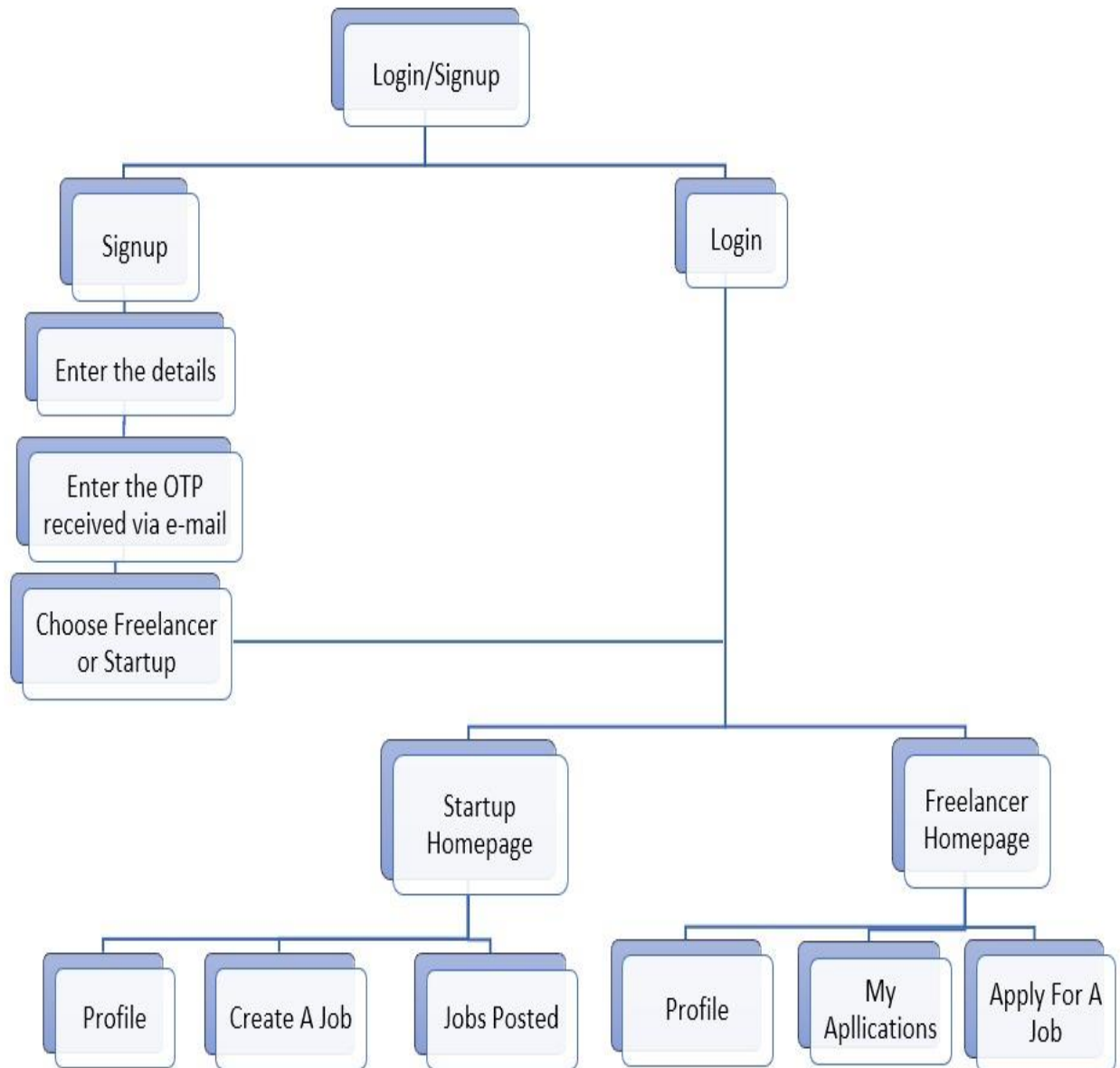
Table 1. Summary of the existing systems

CHAPTER 3

SYSTEM DESIGN

3 SYSTEM DESIGN

3.1 ARCHITECTURE DESIGN



CHAPTER 4

SYSTEM REQUIREMENTS

4. SYSTEM REQUIREMENTS

4.1 HARDWARE

- Memory: 4GB RAM or higher
- Graphics: 2GB
- Processor: i5 or higher
- Hard Disk Space: 500GB or higher

4.2 SOFTWARE (For User)

- Any compatible web browser (eg: Edge, Chrome, etc.)

4.3 SOFTWARE (For Development)

- Spyder for python-flask
- Atom for html, CSS, JavaScript
- Web browser

CHAPTER 5

IMPLEMENTATION DETAILS

5. IMPLEMENTATION DETAILS

5.1 USER INTERFACE

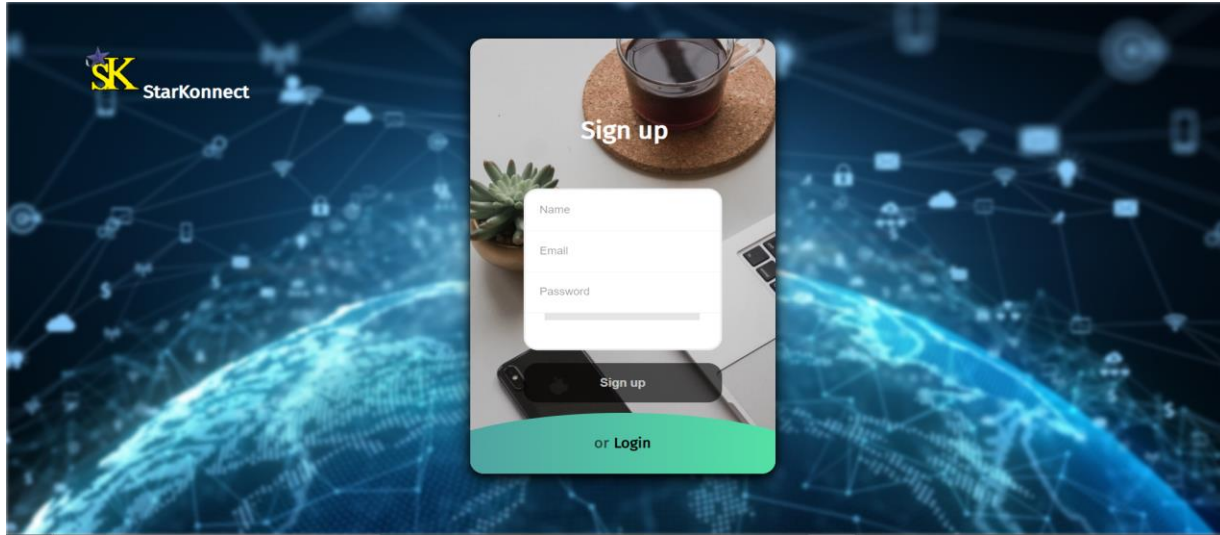


Fig 5.1 Sign up page

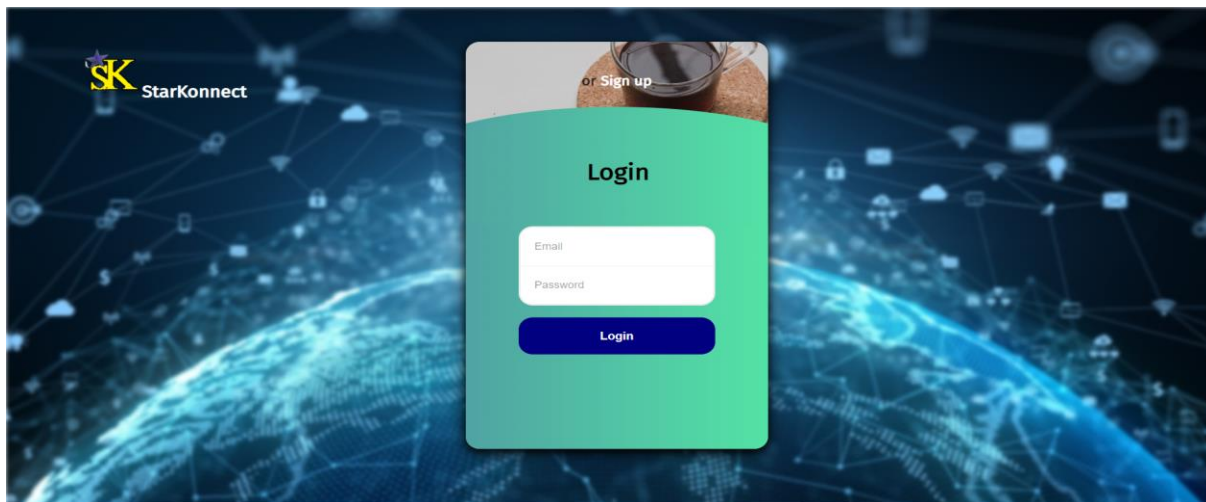


Fig 5.2 User Login page

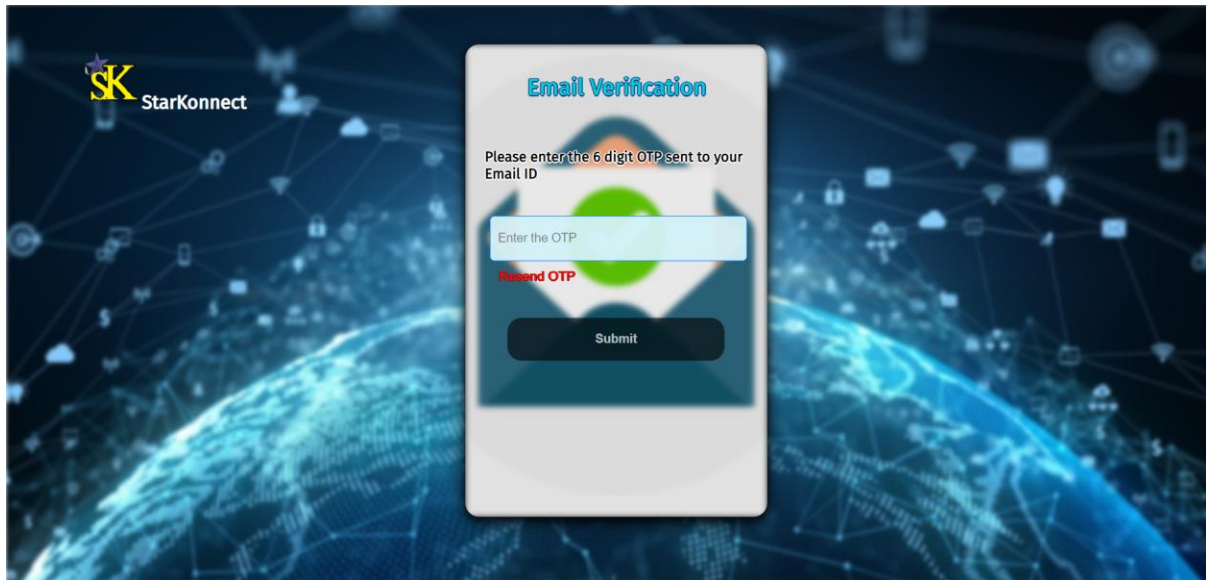


Fig 5.3 OTP Verification page

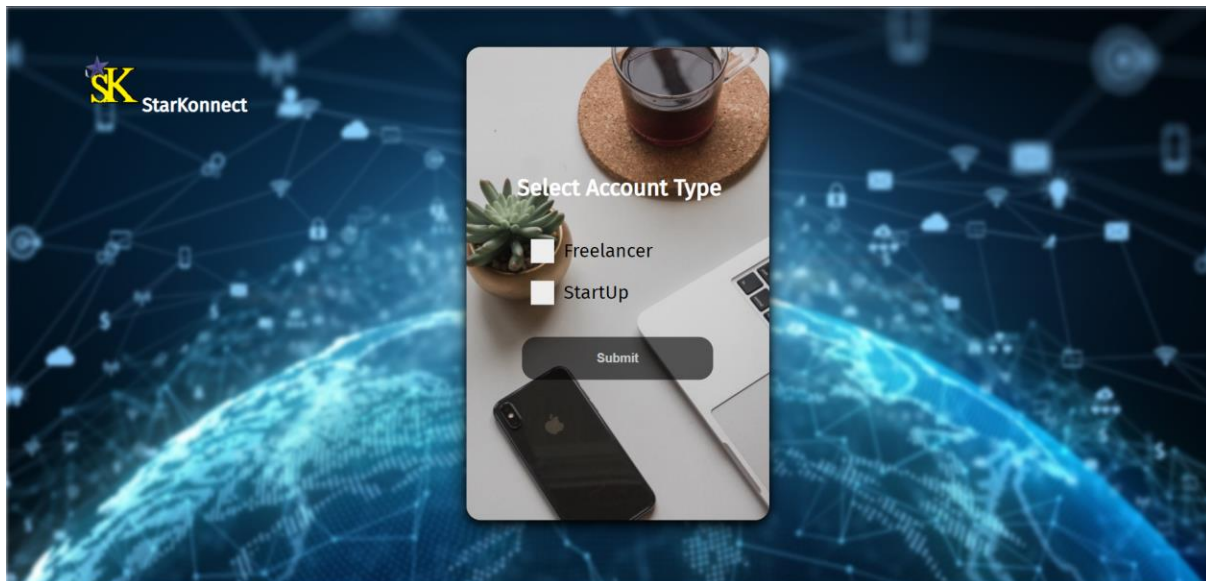


Fig 5.4 Select Account Type

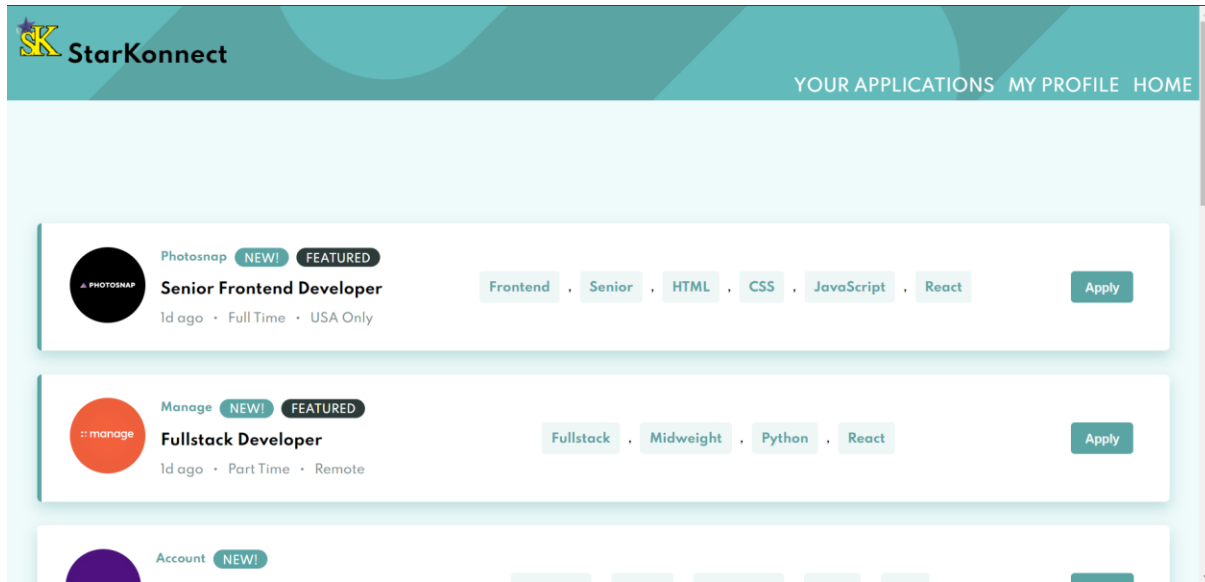


Fig 5.5 Freelancer Home Page

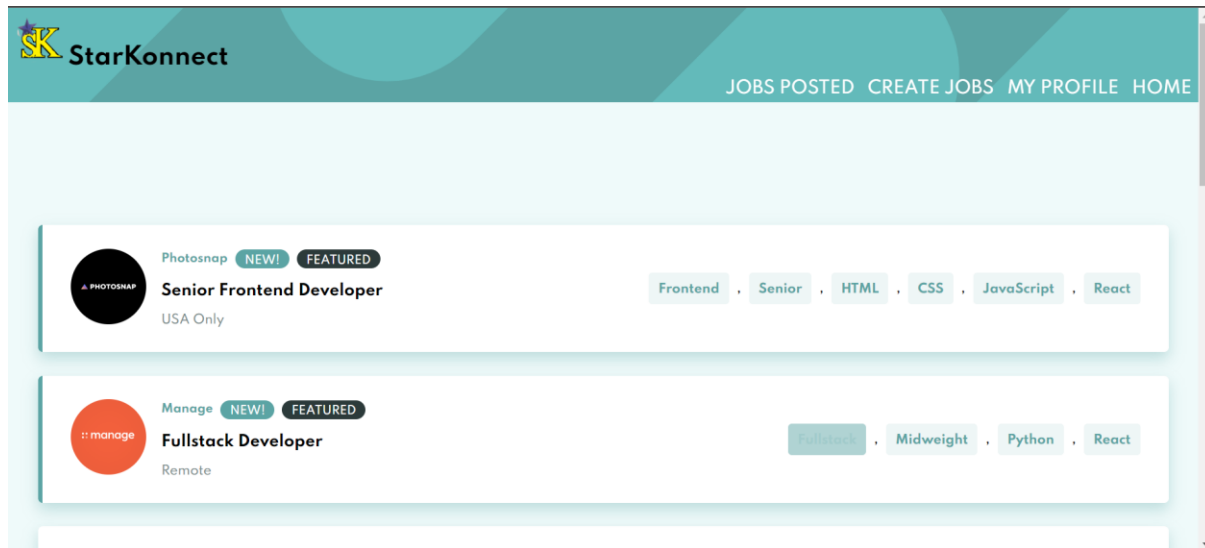


Fig 5.6 Start-up Home Page

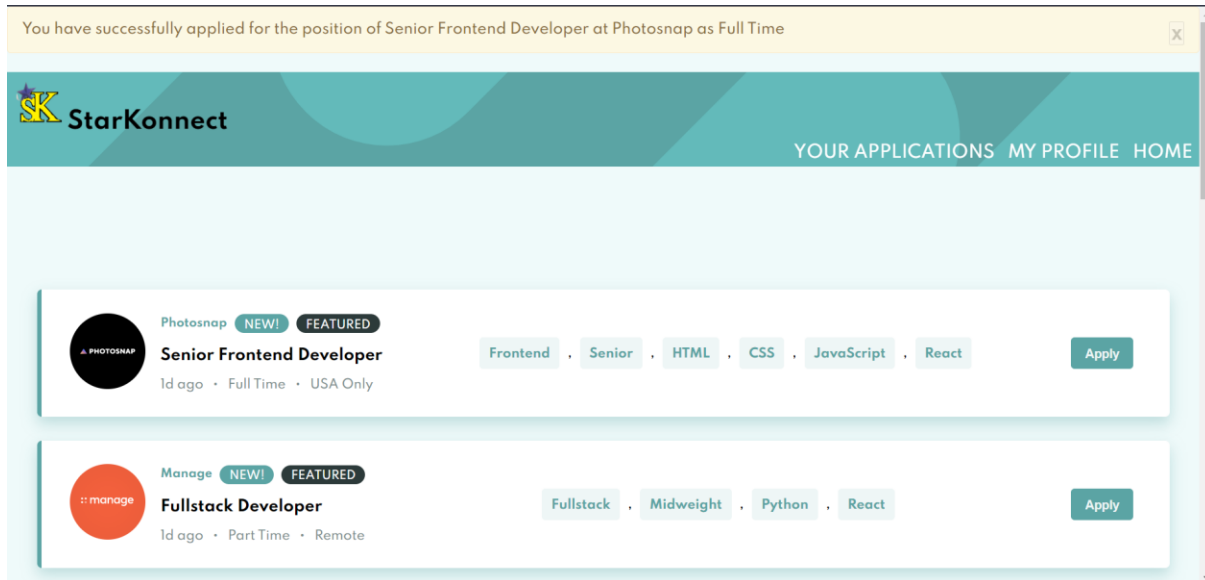


Fig 5.7 Freelancer Successfully Applied to an opening

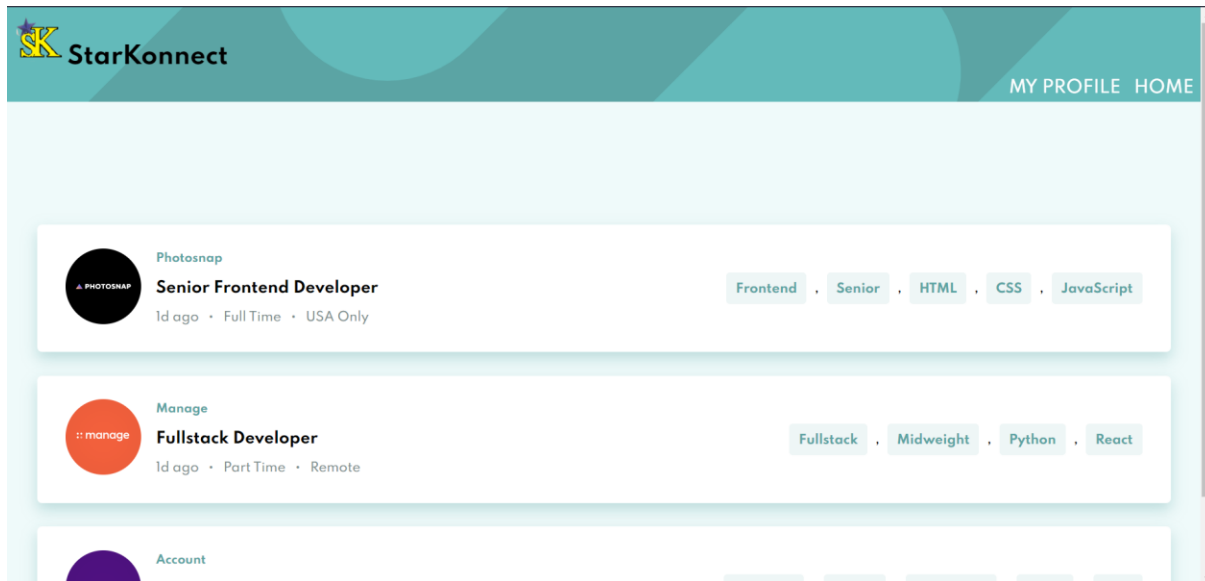


Fig 5.8 Freelancer My Applications Page - Listing all openings they have applied to.

StarKonnnect

FREELANCER PROFILE

DATE OF BIRTH

MOBILE NUMBER (10 digit number)

GENDER ☐ Female ☐ Male ☐ Other

CITY

PIN CODE (6 digit number)

STATE

COUNTRY

ABOUT

Fig 5.9 Freelancer Profile Creation

MADHUR THAKKAR

madhurthakkar247@gmail.com
9082008506


About Progressively evolve cross-platform ideas before impactful infomediaries. Energistically visualize tactical initiatives before cross-media catalysts for change.

Skills Web Design Interface Design Project Direction

Certifications

HTML5	PHP	MySQL
CSS3	Django	MongoDB
JavaScript	React	Angular

Fig 5.10 Freelancer Profile Page



STARTUP PROFILE

COMPANY DOMAIN

ABOUT

Provide insights about your Organisation

CONTACT NUMBER

+XXXXXXXXXX

(With STD Code)

WEBSITE

ADDRESS

PIN CODE

(6 digit number)

CITY

Fig 5.11 Start-up Profile Creation

EYECAM CO

[Eyecam.com](#)
[eyecam@gmail.com](#)
9167986468


About

At Eyecam Co, we draw on our deep expertise of a large pool of experienced security professionals to offer IT security solutions that address the key challenges faced by enterprises today. Our IT security services aim to improve the agility, flexibility and cost effectiveness of the next generation needs of information security and compliance programs. We ensure a holistic risk driven approach for organizations with our solutions in the areas of identity and access governance, data protection, risk & compliance.

Address

Eyecam Co, 3rd Floor, Avenger's

Fig 5.12 Start-up Profile Page



JOBS POSTED MY PROFILE HOME

CREATE VACANCY

POSITION

ROLE

LEVEL

WORK TYPE

Full-Time/Part-Time

LOCATION

LANGUAGES

(comma seperated)

TOOLS

(comma seperated)

Submit

Reset

Fig 5.13 Start-up Create Job Page

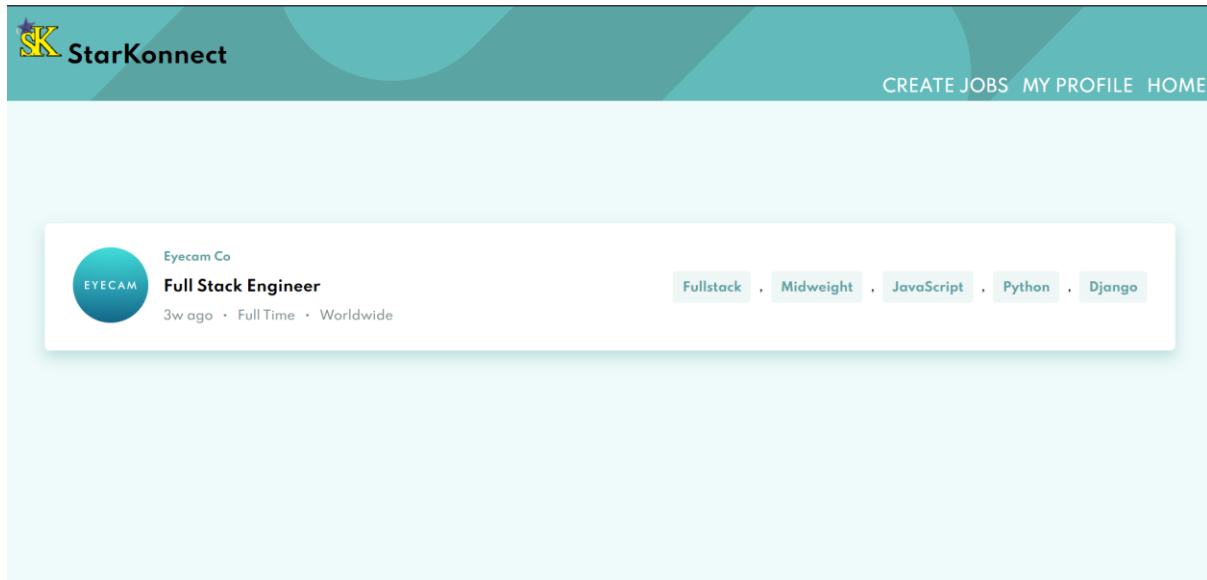


Fig 5.14 Start-up Jobs Posted Page – Displays all the jobs created

5.2 Database Details

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> jobs_applied	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> jobs_posted	★ Browse Structure Search Insert Empty Drop	10	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> startup_profiles	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> userprofiles	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> users	★ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
5 tables	Sum	22	InnoDB	utf8mb4_general_ci	96.0 KiB	0 B

5.21 Database Structure – The database contains 5 tables jobs_applied, jobs_posted, startup_profiles, userprofiles and users

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 4 (5 total, Query took 0.0007 seconds.)

SELECT * FROM `users`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

name	email	date	Profile	password
Madhur	madhurthakkar247@gmail.com	22/07/2021	freelancer	1234
Mikahel	mikhaelangel2001@gmail.com	2021-07-22	startup	123
Eyecam Co	eyecam@gmail.com	2021-07-23	startup	12345
Poonam Bari	poonam.bari@fcrit.ac.in	2021-07-24	freelancer	12345
Madhur	madhurthakkar2001@gmail.com	2021-07-27	freelancer	12345

5.22 users table – Contains data of all the users who are signed up. This data is used for logging in the user.

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

SELECT * FROM `userprofiles`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

name	dob	email	mobile	gender	city	pincode	state	country
Madhur	2001-07-24	madhurthakkar247@gmail.com	9082008506	Male	Kalyan	421301	Maharashtra	India

5.23 userprofiles table – Contains data of all the freelancers who have created their profiles. These details are visible in My Profile page.

Browse	Structure	SQL	Search	Insert	Export	Import	Privileges	Operations	Tracking	More
⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.										
✔ Showing rows 0 - 0 (1 total, Query took 0.0007 seconds.)										
SELECT * FROM `startup_profiles`										
<input type="checkbox"/> Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]										
<input type="checkbox"/> Show all Number of rows: 25 Filter rows: Search this table										
+ Options										
name	domain	email	number	website	pincode	city	state	country		
Eyecam Co	Eyecam Co	eyecam@gmail.com	9167986468	Eyecam.com	421301	Kalyan	Maharashtra	India		

5.24 startup_profiles table – Contains data of all the start-ups who have created their profiles. These details are visible in My Profile page

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Operations

Tracking

More

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 9 (10 total, Query took 0.0005 seconds.)

SELECT * FROM `jobs_posted`

Profiling

[Edit inline]

[Edit]

[Explain SQL]

[Create PHP code]

[Refresh]

Show all

Number of rows: 25

Filter rows: Search this table

+ Options

company	logo	new	featured	position	role	level	postedAt	contract	location	languages	tools
Photosnap	../static/images/photosnap.svg	true	true	Senior Frontend Developer	Frontend	Senior	1d ago	Full Time	USA Only	HTML,CSS,JavaScript	React
Manage	../static/images/manage.svg	true	true	Fullstack Developer	Fullstack	Midweight	1d ago	Part Time	Remote	Python	React
Account	../static/images/account.svg	true	false	Junior Frontend Developer	Frontend	Junior	2d ago	Part Time	USA Only	JavaScript	React,Sass
MyHome	../static/images/myhome.svg	false	false	Junior Frontend Developer	Frontend	Junior	5d ago	Part Time	USA Only	CSS,Javascript	React
Console	../static/images/console.svg	false	false	Software Engineer	FullStack	Midweight	1w ago	Full Time	Worldwide	Javascript	Ruby,Sass

5.25 jobs_posted table – Contains data of all the jobs that are created by the start-ups. These are visible to the freelancers for applying.

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.										
Showing rows 0 - 2 (3 total, Query took 0.0007 seconds.)										
SELECT * FROM `jobs_applied`										
<input type="checkbox"/> Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]										
<input type="checkbox"/> Show all Number of rows: 25 Filter rows: Search this table										
+ Options										
app_email	app_name	company	position	role	level	contract				
madhurthakkar247@gmail.com	Madhur	Photosnap	Senior Frontend Developer	Frontend	Senior	Full Time				
madhurthakkar247@gmail.com	Madhur	Manage	Fullstack Developer	Fullstack	Midweight	Part Time				
madhurthakkar247@gmail.com	Madhur	Account	Junior Frontend Developer	Frontend	Junior	Part Time				

5.26 jobs_applied table – Contains data of the jobs that are applied by the freelancers and the email ids of the freelancers who applied for them. These are visible to the freelancers in My Applications page.

CHAPTER 6

EXPERIMENTAL RESULTS

6. EXPERIMENTAL RESULTS

We had aimed to design a website that will connect Start-ups and Freelancers so as to simplify recruitment process and help emerging start-ups to grow. We successfully developed a website using HTML5, CSS3, JavaScript and Flask Python framework to achieve our objective and have created an elegant User Interface to provide a smooth experience. The Login/Sign Up process is easy and interactive as well as secure. We have provided various features like My Applications so that Freelancers can keep track of the jobs they have applied to and a Create Jobs and Job Posted page for Start-ups to post and keep track of jobs easily, apart from common features like My Profile and all this in a manner that will not overwhelm the users.

CHAPTER 7

CONCLUSION/FUTURE SCOPE

7 CONCLUSION/FUTURE SCOPE

7.1 CONCLUSION

Job Search Portals stands as a revolutionizing element in the sphere of recruitment. They act as a communication bridge between applicants and recruiters facilitating their requirements. StartKonnnect website helps organizations to have a greater exposure to the candidate pool and also job seekers facilitating wide search of jobs matching their interests. It provides flexibility to the jobseekers to view the openings and applied jobs. This website provides an enhanced user experience for both employer and jobseeker. It provides user friendly interface which facilitates in reaching wide range of audience. The application has achieved all the requirements that were initially set in the requirements gathering phase.

7.2 FUTURE SCOPE

1. Provide Encryption of password.
2. Perform Data Validation.
3. Provide Messaging feature.
4. Provide resume-builder
5. Create a search engine.

CHAPTER 8

APPENDIX: CODE SAMPLES

8 APPENDIX: CODE SAMPLE

```
from flask import *
from flask_mail import Mail, Message
import random
import secrets
from flask_mysql import MySQL
from flask_session import Session
from datetime import datetime

app = Flask(__name__)
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'starkconnect_users'

mysql = MySQL(app)
app.secret_key = secrets.token_bytes(16)
app.config['SESSION_PERMANENT'] = False
app.config['SESSION_TYPE'] = "filesystem"
Session(app)
app.config['MAIL_SERVER'] = 'smtp.gmail.com'
app.config['MAIL_PORT'] = 465
app.config['MAIL_USERNAME'] = 'startkconnect4@gmail.com'
app.config['MAIL_PASSWORD'] = 'startkconnect1234'
app.config['MAIL_USE_TLS'] = False
app.config['MAIL_USE_SSL'] = True
mail = Mail(app)

@app.route('/')
def index():
    return render_template("login.html")

@app.route('/verify', methods=['GET', 'POST'])
def verify():
    if request.method == "POST":
        if request.form['name'] != "" and request.form['email'] != "" and request.form['password'] != "":
            ver_email = request.form['email']
```

```

session['email'] = ver_email
    reg = None

    try:
        cur = mysql.connection.cursor()
        cur.execute("SELECT * FROM users WHERE email = '"+session.get(
'email')+"'")
        reg = cur.fetchone()
        print(reg,session.get('email'))
        mysql.connection.commit()
        cur.close()
    except:
        print(reg,session.get('email'))
    if reg is None:
        otp_gen = random.randint(100000,999999)
        session['otp'] = otp_gen
        print("verify")
        print(session.get('otp'))
        msg="Your One Time Password for Starkconnect is: "+str(session.
get('otp'))
        message = Message("OTP Verification",sender="email",recipients
=[session.get('email')])

        message.body = msg

        mail.send(message)

        success = "Message Sent"

        session['name'] = request.form.get('name')
        session['password'] = request.form['password']

        return render_template("otpverification.html",success=success)
    else:
        flash('This Email is already registered')
        return render_template("login.html")
else:
    flash('Please enter all the details properly!')
    return render_template("login.html")

@app.route('/authenticate', methods=['GET','POST'])
def authenticate():
    if request.method == "POST":
        otp_enter = request.form['otp']
        if len(otp_enter)<1:

```

```

        otp_enter = -1
    if int(otp_enter)==session.get('otp'):
        return render_template("select.html")
    else:
        if otp_enter==--1:
            flash(u"New OTP is sent to {0}".format(session.get('email')))
        else:
            flash(u"Incorrect OTP\nTry Again!")
        return render_template("otpverification.html")

@app.route('/resend_verify', methods=['GET','POST'])
def resend_verify():
    if request.method == "POST":
        #email = request.form['email']
        otp_gen = random.randint(100000,999999)
        session['otp'] = otp_gen
        print("resend")
        print(session.get('otp'))
        msg="Your One Time Password for Starkconnect is: "+str(session.get('otp
'))
        message = Message("OTP Verification",sender="startkconnect4@gmail.com",
recipients=[session.get('email')])

        message.body = msg

        mail.send(message)

        success = "Message Sent"

        return render_template("otpverification.html",success=success)

@app.route('/selection',methods=['GET','POST'])
def selection():
    if request.method=='POST':
        chkbbox = request.form.getlist('chkbbox')
        session['profile'] = chkbbox[0]
        cur = mysql.connection.cursor()
        cur.execute("INSERT INTO users VALUES('"+session.get('name')+"','"+ses
sion.get('email')+"','"+str(datetime.now())+"','"+session.get('profile')+"','"+
+session.get('password')+"')")
        mysql.connection.commit()
        cur.close()
        # cur = mysql.connection.cursor()
        # cur.execute("SELECT * FROM jobs_posted")

```

```

# jobs = cur.fetchall()
# mysql.connection.commit()
# cur.close()
# lis = list()
# i=1
# for ele in jobs:
#     dic = dict()
#     dic['id'] = i
#     dic['company'] = ele[0]
#     dic['logo'] = ele[1]
#     dic['new'] = ele[2]
#     dic['featured'] = ele[3]
#     dic['position'] = ele[4]
#     dic['role'] = ele[5]
#     dic['level'] = ele[6]
#     dic['postedAt'] = ele[7]
#     dic['contract'] = ele[8]
#     dic['location'] = ele[9]
#     dic['languages'] = ele[10].split(',')
#     dic['tools'] = ele[11].split(',')
#     i+=1
#     lis.append(dic)
if session.get('profile')=="freelancer":
    return render_template("homepage.html")
elif session.get('profile')=="startup":
    return render_template("homepage-startup.html")

@app.route('/userlogin',methods=['GET','POST'])
def userlogin():
    if request.method=='POST':
        if request.form['loginemail'] != "" and request.form['loginpass'] != "":
            email = request.form['loginemail']
            session['email'] = email
            session['password'] = request.form['loginpass']
            reg = None
            try:
                cur = mysql.connection.cursor()
                cur.execute("SELECT * FROM users WHERE email = '"+session.get('email')+"'")
                reg = cur.fetchone()
                session['name'] = reg[0]
                session['profile'] = reg[3]
                mysql.connection.commit()
                cur.close()

```

```

except:
    pass
if reg is None:
    flash('This Email is not registered')
    return render_template("login.html")
else:
    jobs = ()
    if session.get('password')==reg[4]:
        # cur = mysql.connection.cursor()
        # cur.execute("SELECT * FROM jobs_posted")
        # jobs = cur.fetchall()
        # mysql.connection.commit()
        # cur.close()
        # lis = list()
        # i=1
        # for ele in jobs:
        #     dic = dict()
        #     dic['id'] = i
        #     dic['company'] = ele[0]
        #     dic['logo'] = ele[1]
        #     dic['new'] = ele[2]
        #     dic['featured'] = ele[3]
        #     dic['position'] = ele[4]
        #     dic['role'] = ele[5]
        #     dic['level'] = ele[6]
        #     dic['postedAt'] = ele[7]
        #     dic['contract'] = ele[8]
        #     dic['location'] = ele[9]
        #     dic['languages'] = ele[10].split(',')
        #     dic['tools'] = ele[11].split(',')
        #     i+=1
        #     lis.append(dic)
        if session.get('profile')=="freelancer":
            return render_template("homepage.html")
        elif session.get('profile')=="startup":
            return render_template("homepage-startup.html")
        else:
            flash('Incorrect Password')
            return render_template('login.html')
    else:
        flash('Please enter all the details properly!')
        return render_template("login.html")

```

```

@app.route('/myprofile',methods=['GET','POST'])
def myprofile():

```

```

    if request.method=="POST":
        profile = ""
        found = False
        try:
            cur = mysql.connection.cursor()
            cur.execute("SELECT Profile FROM users WHERE email = '"+session.get('email')+"'")
            profile = cur.fetchone()[0]
            mysql.connection.commit()
            cur.close()
        except:
            profile = ""
        try:
            cur = mysql.connection.cursor()
            if profile=="freelancer":
                cur.execute("SELECT name FROM userprofiles WHERE email = '"+session.get('email')+"'")
            elif profile=="startup":
                cur.execute("SELECT name FROM startup_profiles WHERE email = '"+session.get('email')+"'")
            if cur.fetchone()[0]:
                found = True
            mysql.connection.commit()
            cur.close()
        except:
            pass
        if profile=="freelancer":
            if found:
                return render_template("Freelancer-Profile.html")
            else:
                return render_template("Freelancer.html")
        elif profile=="startup":
            if found:
                return render_template("Startup-Profile.html")
            else:
                return render_template("Startup.html")
        print(profile)

@app.route('/viewprofile/<prof>', methods=['GET', 'POST'])
def viewprofile(prof):
    if request.method=="POST":
        if prof=="freelancer":
            session['dob'] = request.form['dob']
            session['mob'] = request.form['Mobile_Number']
            genbox = request.form.getlist('Gender')

```



```

        session['gender'] = genbox[0]
        session['city'] = request.form['city']
        session['pincode'] = request.form['pincode']
        session['state'] = request.form['State']
        session['country'] = request.form['Country']
        try:
            cur = mysql.connection.cursor()
            #print("INSERT INTO userprofiles VALUES('"+session.get('name')
+ "','"+session.get('dob')+"', '"+session.get('email')+"', '"+session.get("mob")+
+ "','"+session.get('gender')+"', '"+session.get('city')+"', '"+session.get('pinco
de')+"', '"+session.get('state')+"', '"+session.get('country')+"')")
            cur.execute("INSERT INTO userprofiles VALUES('"+session.get('n
ame')+"', '"+session.get('dob')+"', '"+session.get('email')+"', '"+session.get("m
ob")+"', '"+session.get('gender')+"', '"+session.get('city')+"', '"+session.get('
pincode')+"', '"+session.get('state')+"', '"+session.get('country')+"')")
            mysql.connection.commit()
            cur.close()
        except Exception as e:
            print(e)

    elif prof=="startup":
        session['domain'] = request.form['Company_Domain']
        session['number'] = request.form['Contact_Number']
        session['website'] = request.form['Website']
        session['city'] = request.form['City']
        session['pincode'] = request.form['Pin_Code']
        session['state'] = request.form['State']
        session['country'] = request.form['Country']
        try:
            cur = mysql.connection.cursor()
            print("INSERT INTO startup_profiles VALUES('"+session.get('nam
e')+"', '"+session.get('domain')+"', '"+session.get('email')+"', '"+session.get("
number")+"', '"+session.get('website')+"', '"+session.get('pincode')+"', '"+sessi
on.get('city')+"', '"+session.get('state')+"', '"+session.get('country')+"')")
            cur.execute("INSERT INTO startup_profiles VALUES('"+session.ge
t('name')+"', '"+session.get('domain')+"', '"+session.get('email')+"', '"+session
.get("number")+"', '"+session.get('website')+"', '"+session.get('pincode')+"', '"+
session.get('city')+"', '"+session.get('state')+"', '"+session.get('country')+"
')")
            mysql.connection.commit()
            cur.close()
            print("Added")
        except Exception as e:
            print(e)
    if prof=="freelancer":

```

```

        return render_template("homepage.html")
    elif prof=="startup":
        return render_template("homepage-startup.html")

@app.route('/myapp',methods=['GET','POST'])
def myapp():
    if request.method=="POST":
        return render_template("Freelancer-Applications.html")

@app.route('/jobsposted',methods=['GET','POST'])
def jobsposted():
    if request.method=="POST":
        return render_template("Jobsposted.html")

@app.route('/Createjob',methods=['GET','POST'])
def Createjob():
    if request.method=="POST":
        return render_template("Create-Vacancy.html")

@app.route('/homebt/<profi>',methods=['GET','POST'])
def homebt(profi):
    if request.method=="POST":
        if profi=="freelancer":
            return render_template("homepage.html")
        elif profi=="startup":
            return render_template("homepage-startup.html")

@app.route('/applyjob/<company>',methods=['GET','POST'])
def applyjob(company):
    if request.method=="POST":
        cur = mysql.connection.cursor()
        cur.execute("SELECT * FROM jobs_posted WHERE company = '"+company+"'")
        jobcom = cur.fetchone()
        mysql.connection.commit()
        cur.execute("SELECT * FROM jobs_applied WHERE company = '"+company+"'")
    )
    jobcheck = cur.fetchone()
    mysql.connection.commit()
    cur.close()
    if jobcheck:
        if jobcheck[0]==session.get('email') and jobcheck[2]==company and
jobcheck[3]==jobcom[4]:
            flash("You have already applied for the position of "+jobcom[4]
]+" at "+company+" as "+jobcom[8])
        return render_template("homepage.html")

```

```

        else:
            cur = mysql.connection.cursor()
            cur.execute("INSERT INTO jobs_applied VALUES ('"+session.get('email')+ "','"+session.get('name')+ "','"+company+ "','"+jobcom[4]+ "','"+jobcom[5]+ "','"+jobcom[6]+ "','"+jobcom[8]+ "','")")
            mysql.connection.commit()
            cur.close()
            flash("You have successfully applied for the position of "+jobcom[4]+" at "+company+" as "+jobcom[8])
            return render_template("homepage.html")
        else:
            cur = mysql.connection.cursor()
            cur.execute("INSERT INTO jobs_applied VALUES ('"+session.get('email')+ "','"+session.get('name')+ "','"+company+ "','"+jobcom[4]+ "','"+jobcom[5]+ "','"+jobcom[6]+ "','"+jobcom[8]+ "','")")
            mysql.connection.commit()
            cur.close()
            flash("You have successfully applied for the position of "+jobcom[4]+" at "+company+" as "+jobcom[8])
            return render_template("homepage.html")

if __name__ == '__main__':
    app.run()

```

CHAPTER 9

REFERENCES

9. REFERENCES

These numbers should come in the report in the serial order in which they are being used. Any content from the internet needs to be references properly in the manner given below.

[1] <https://www.linkedin.com/help>

[2] https://internshala.com/about_us

[3] <https://www.monster.com/about/>

CHAPTER 10

ACKNOWLEDGEMENT

10 ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to our college Fr. Conceicao Rodrigues Institute of Technology for giving us an opportunity to undertake this project. We are grateful to our H.O.D. Dhanashree Hadsul Mam for giving us a chance to work on this project in our course of Engineering. We are thankful to, and fortunate enough to get constant encouragement, support and guidance from all our teachers of IT Department which helped us in successfully completing our project.

We would also like to extend our gratitude to our parents for their support throughout, and the constant efforts and motivation by our teammates and friends which helped us attain successful completion of this project.

.