**Internship Project Report On**

**PLACEMENT PORTAL FOR CBIT**

Submitted in partial fulfilment for the requirements of

**COSC INTERNSHIP DRIVE-2020**

## In

**COMPUTER SCIENCE AND ENGINEERING**

## By

**Kedarnath Chaturvedula- 160119733**

**Nikitha Bogala- 160117733**

**Dasari Kini Sanjula- 160118733066**

**Sai Sasank -160118733112**

**Barnaboss Puli-160118733**

**Sai Bhanu Revanth-160119733102**

**Pooja Reddy- 160119733**

**Annanya Reddy- 160119733063**



**Department of Computer Science and Engineering,**

**Chaitanya Bharathi Institute of Technology (Autonomous),**

**(Affiliated to Osmania University, Hyderabad) Hyderabad, TELANGANA (INDIA) – 500 075 May-2020**

**CERTIFICATE**

This is to certify that the project titled **“PLACEMENT PORTAL FOR CBIT”** is the bonafide work carried out by **Kedarnath Chaturvedula, Nikitha Bogala, Dasari Kini Sanjula- 160118733066, Sai Sasank -160118733 , Barnaboss Puli, Sai Bhanu Revanth, Pooja Reddy, Annanya Reddy** student of B.E.(CSE) of Chaitanya Bharathi Institute of Technology, Hyderabad, Telangana(India) during the academic year 2020-2021, submitted in for the project completion of COSC Internship Drive and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

## Mentors: Rakesh Sirikonda

## Faculty Mentor:

Smt Dharani

## Head of the Department

Dr. Y. Rama Devi

# DECLARATION

We hereby declare that the project entitled **“PLACEMENT PORTAL FOR CBIT”** submitted for COSC INTERNSHIP DRIVE-2020 is our original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

## Name(s) and Signature(s) of the Students:

**Kedarnath Chaturvedula**

**Nikitha Bogala**

**Dasari Kini Sanjula**

**Sai Sasank**

**Barnaboss Puli**

**Sai Bhanu Revanth**

**Pooja Reddy**

**Annanya Reddy**

## Place: Hyderabad Date: 1-7-2020

**ABSTRACT**

Days at college get tougher as students head towards the final year of their Higher Education degree programs. Campus placements provide the students with a foot in the door opportunity, enabling them to start off their career right after they have completed their course curriculum. College placements give a good head start for students in their pursuit of the career.

To ease this vital process for the students, this project provides a platform for user/student where they can view the list of companies appearing for campus placement and apply to the respective companies according to their eligibility criteria.

Apart from just applying for companies, they are also notified and updated about upcoming seminars, exams, events, etc. The user/ students can also edit their profile and make changes as per their needs.

On the other hand, the Admin module deals with calling the rest apis whatever have been created. The rest API’s include checking users, adding a new company, checking user details based on the companies, getting the date statistics based on several categories , adding announcements and getting the announcements data by date.

The rest APIS’s ensure connectivity between the admin and the user modules for the flexibility of the application and to use the database in an effective manner.

# ACKNOWLEDGEMENTS

We would like to express our heartfelt gratitude to **Smt. Dr. Y Rama Devi, Smt. Dharani** project guides, for their invaluable guidance and constant support, along with her capable instruction and persistent encouragement.

Our special thanks are due to all members of **COSC Club**  and our faculty for providing us with the help required to carry out the groundwork of this project and members of COSC club for conducting this Internship Drive and delivering us informative sessions and constantly motivating us for the completion of work and patiently clarifying all our doubts throughout the COSC Internship Drive.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | Title Page | i |
|  | Certificate of the Guide | ii |
|  | Declaration of the Student | iii |
|  | Abstract | iv |
|  | Acknowledgement | v |
| **1.** | **INTRODUCTION** | **1** |
|  | * 1. Problem Definition including the significance and objective   2. Methodologies   3. Outline of the results   4. Scope of the project | 1  1  2  2 |
| **2.** | **LITERATURE SURVEY** | **3** |
|  | * 1. Introduction to the problem domain terminology   2. Existing Solutions   3. Hardware Requirements   4. Software Requirements | 3  3  3 |
| **3.** | **DESIGN OF THE PROPOSED SYSTEM/METHOD/ALGORITHM** | **4** |
|  | * 1. Block Diagram   2. Module Description   3.3. Algorithm | 4  5  6 |
| 4 | **IMPLEMENTATION OF THE PROPOSED SYSTEM** | **7** |
|  | * 1. Pseudo Code   2. Dataset Description | 7  10 |
| **5.** | **RESULT AND DISCUSSIONS** | **11** |
| **6.** | **CONCLUSION AND FUTURE WORK** | **17** |
|  | * 1. Conclusions      1. Limitations   2. Future Work | 17  17  17 |
| **7.** | **REFERENCES** | **18** |

1. **INTRODUCTION**

## Problem Definition including the significance and objective:

## 

Using emails as the only way to broadcast/ make announcements about company postings and events can end up as a problem as not every student checks his/ her email routinely.

To avoid this, there is an emergent need of an app that notifies students about important announcements by the placement cell, directly on their phones.

This project aims to develop an application that makes the aforementioned processes easier while overcoming the problems prevailing in the existing system.

## Methodologies:

The way of performing this process is precisely in the following way. To check announcements or posting created by the admin, the user has to login. In the same way, the admin has to login to make announcements, add companies, check user details and check registration details.

## Outline of results

Through this project, the user can check on various announcements made by the admin and also register for postings. The user has to login first. The admin can add new announcements, new companies and check on user details and registration details. The application that the user uses and the website that the admin uses make use of the same API and database; hence any update made by the admin is reflected on the user application and vice versa.

## Scope of project

This project can be used by the students and the management of a college/ university. The UI is simple to use and it makes announcing easier. It can be used instead of a broadcast email.

# LITERATURE SURVERY

## Introduction to the problem domain

## 

## The placement portal provides an all round functionality to the users without much learning curve where the users can create their accounts or profiles check companies and register for them , view announcements, and also can apply for placements.

## The admin on the other hand provides al these functions by adding announcements, adding companies and checking user details and obtaining statistics.

## Existing solutions

There are Static Approaches to the system but there is no sort of dynamic approach to this. The current project calculates dynamically.

## Hardware Requirements:

* + - Laptop/PC with minimum of 4GB RAM

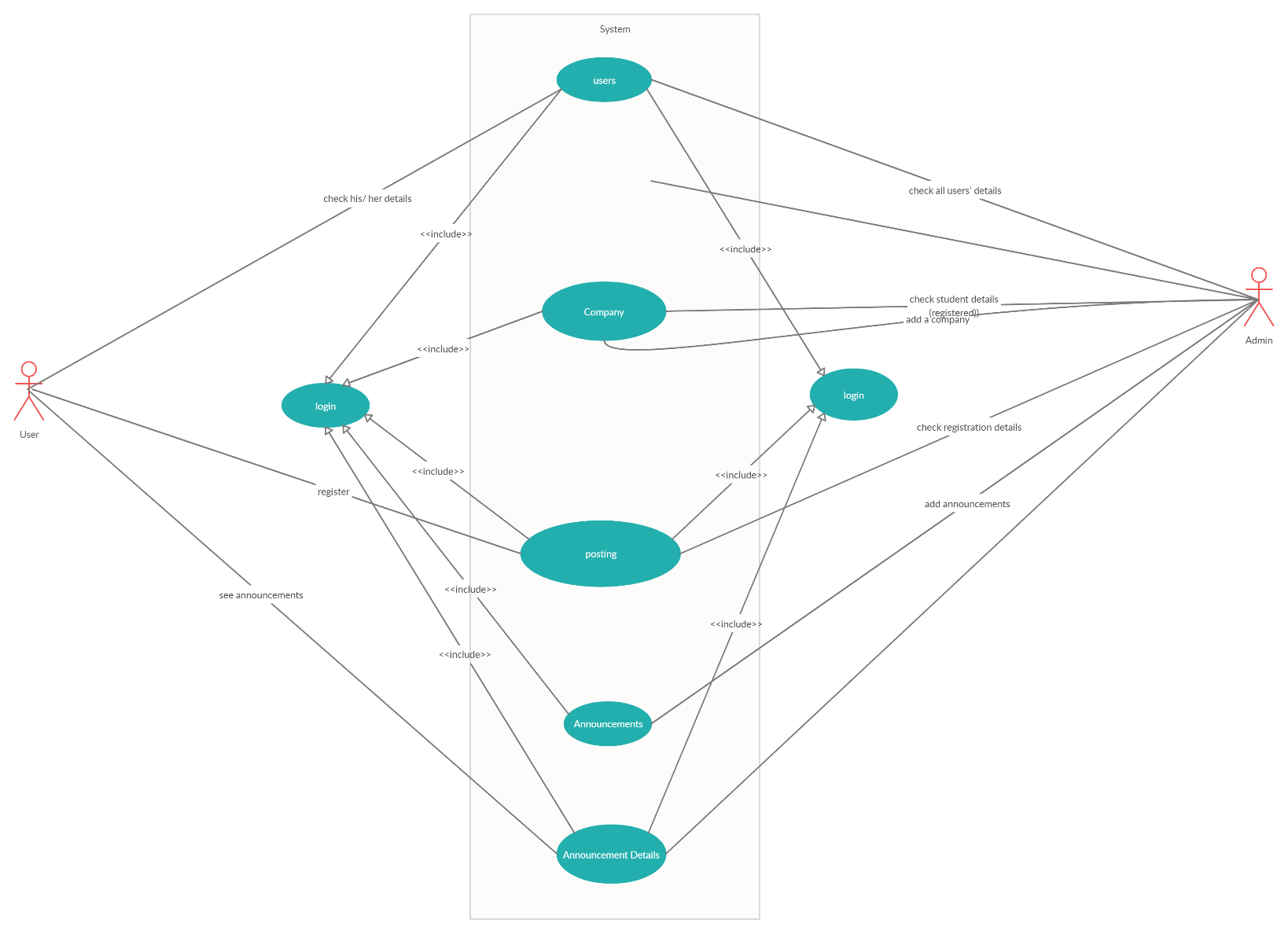
## Software Requirements

* + - Requests Library
    - VS code 2019
    - Python Latest Version
    - MySQl Workbench(Database)
    - Flask and Django(python web frameworks)
    - HTML/CSS,Bootsrap,Javascript

**3.**

# DESIGN OF THE PROPOSED SYSTEM

## Block Diagram



**Fig 3.1 – Block Diagram**

The design/flow of our project is very simple. Both the user and the admin have to login to access the functionalities of the app and the website respectively. The admin can check details about users, companies, students that have registered for a posting and announcements, whereas users can check their own details, register for a posting and check announcement details after logging in.

## Description of Modules:

**User Module:**

The User module is an android app which is developed in android studio where the user can register, create his accounts and access the portal’s functionality. It uses android studio and java to create a full fledged mobile application along with a mysql database. The rest apis required are called here

## Admin module:

The admin module is a website created using Django where all the required rest api’s are called and functionalities are provided to the admin where he can add companies, announcements and access user details.

## Front End:

It uses HTML, CSS, JavaScript and Bootstrap.

The Front End is different for the user modules and admin modules . In the mobile application the user can access all the functionalities through a simple UI designed for ease and responsiveness. On the admin module the admin can execute his functionalities through the website. Both the admin and the user module have to be logged in to perform any functions or actions and they can be logged in if they have their account registered on the website or app.

## Back End:

Backend is developed in Django and flask. The database used here is MySQl which is integrated with the front end and the back end . In flask the resp apis are written and deployed to the Heroku server. Those rest apis can be called in the backend using Django for admin module ans using android studio for the user module.

# IMPLENTATION OF THE PROPOSED SYSTEM

## Pseudo Code

**views.py- Code for calling apis:**

from django.shortcuts import render

from django.http import HttpResponse

import requests

# Create your views here.

def home(request):

return render(request,'home/home.html')

def about(request):

return render(request,'home/about.html')

def checkusers(request):

data=requests.get("https://placementcosc.herokuapp.com/userscheckadmin")

data=data.json()

context={'data':data}

return render(request,'home/check.html',context)

def addcompanyform(request):

return render(request,'home/addcompanyform.html')

def addcompany(request):

company\_name=request.POST.get('company\_name','')

job\_name=request.POST.get('job\_name','')

job\_description=request.POST.get('job\_description','')

job\_requirements=request.POST.get('job\_requirements','')

min\_gpa=request.POST.get('min\_gpa','')

data=requests.post("https://placementcosc.herokuapp.com/addcomadmin",data={"company\_name":company\_name,

"job\_name":job\_name,

"job\_description":job\_description,

"job\_requirements":job\_requirements,

"min\_gpa":min\_gpa

})

data=data.json()

return HttpResponse(data["message"])

def adminannounceform(request):

return render(request,'home/adminannounceform.html')

def adminannounce(request):

type=request.POST.get('type','')

title\_description=request.POST.get('title\_description','')

date\_event=request.POST.get('date\_event','')

time\_event=request.POST.get('time\_event','')

event\_description=request.POST.get('event\_description','')

image\_link=request.POST.get('image\_link','')

data=requests.post("https://placementcosc.herokuapp.com/announceadmin",data={"type":type,

"title\_description":title\_description,

"date\_event":date\_event,

"time\_event":time\_event,

"event\_description":event\_description,

"image\_link":image\_link

})

data=data.json()

return HttpResponse(data["message"])

def userregdetailsform(request):

return render(request,'home/userregdetailsform.html')

def userregdetails(request):

company\_name=request.POST.get('company\_name','')

data=requests.get("https://placementcosc.herokuapp.com/userregcomadmin",data={"company\_name":company\_name})

data=data.json()

context1={'data':data}

return render(request,'home/userregdetails.html',context1)

def datestatsforplace(request):

Date=request.POST.get('Date','')

data=requests.get("https://placementcosc.herokuapp.com/plastudate",data={"Date":Date})

data=data.json()

context1={'data':data}

return render(request,'home/datestatsplace.html',context1)

def datestatsforplaceform(request):

return render(request,'home/datestatsforplaceform.html')

def datestatsforcom(request):

Date=request.POST.get('Date','')

data=requests.get("https://placementcosc.herokuapp.com/comstudate",data={"Date":Date})

data=data.json()

context1={'data':data}

return render(request,'home/datestatscom.html',context1)

def datestatsforcomform(request):

return render(request,'home/datestatsforcomform.html')

def datestatsforann(request):

Date=request.POST.get('Date','')

data=requests.get("https://placementcosc.herokuapp.com/anndate",data={"Date":Date})

data=data.json()

context1={'data':data}

return render(request,'home/datestatsann.html',context1)

def datestatsforannform(request):

return render(request,'home/datestatsforannform.html')

**app.py- Code for creating apis:**

from flask import Flask,jsonify

from flask\_restful import Api

from flask\_jwt\_extended import JWTManager

from resources.place import Phase,PlaceLogin,Checkusers,Registerstud,Userregistrstiondet,AdminAnnounce,Addcompany,UserAnnouncecheck,DateStatsforplace,Datestatsforcom,Datestatsforann,userupdate

app=Flask(\_\_name\_\_)

app.config['PROPAGATE\_EXCEPTIONS']=True

app.config['JWT\_SECRET\_KEY']='coscskillup'

api=Api(app)

jwt=JWTManager(app)

@jwt.unauthorized\_loader

def missing\_token\_callback(error):

return jsonify({'error':'authourisation required','description':'request doesnot contain an access token'}),401

@jwt.invalid\_token\_loader

def invalid\_token\_callback(error):

return jsonify({'error':'invalid token','description':'signature verification failed'}),401

api.add\_resource(Phase,'/placesignup')

api.add\_resource(PlaceLogin,'/login')

api.add\_resource(Registerstud,'/regstud')

api.add\_resource(UserAnnouncecheck,'/anntostud')

api.add\_resource(Checkusers,'/userscheckadmin')

api.add\_resource(Userregistrstiondet,'/userregcomadmin')

api.add\_resource(AdminAnnounce,'/announceadmin')

api.add\_resource(Addcompany,'/addcomadmin')

api.add\_resource(DateStatsforplace,'/plastudate')

api.add\_resource(Datestatsforcom,'/comstudate')

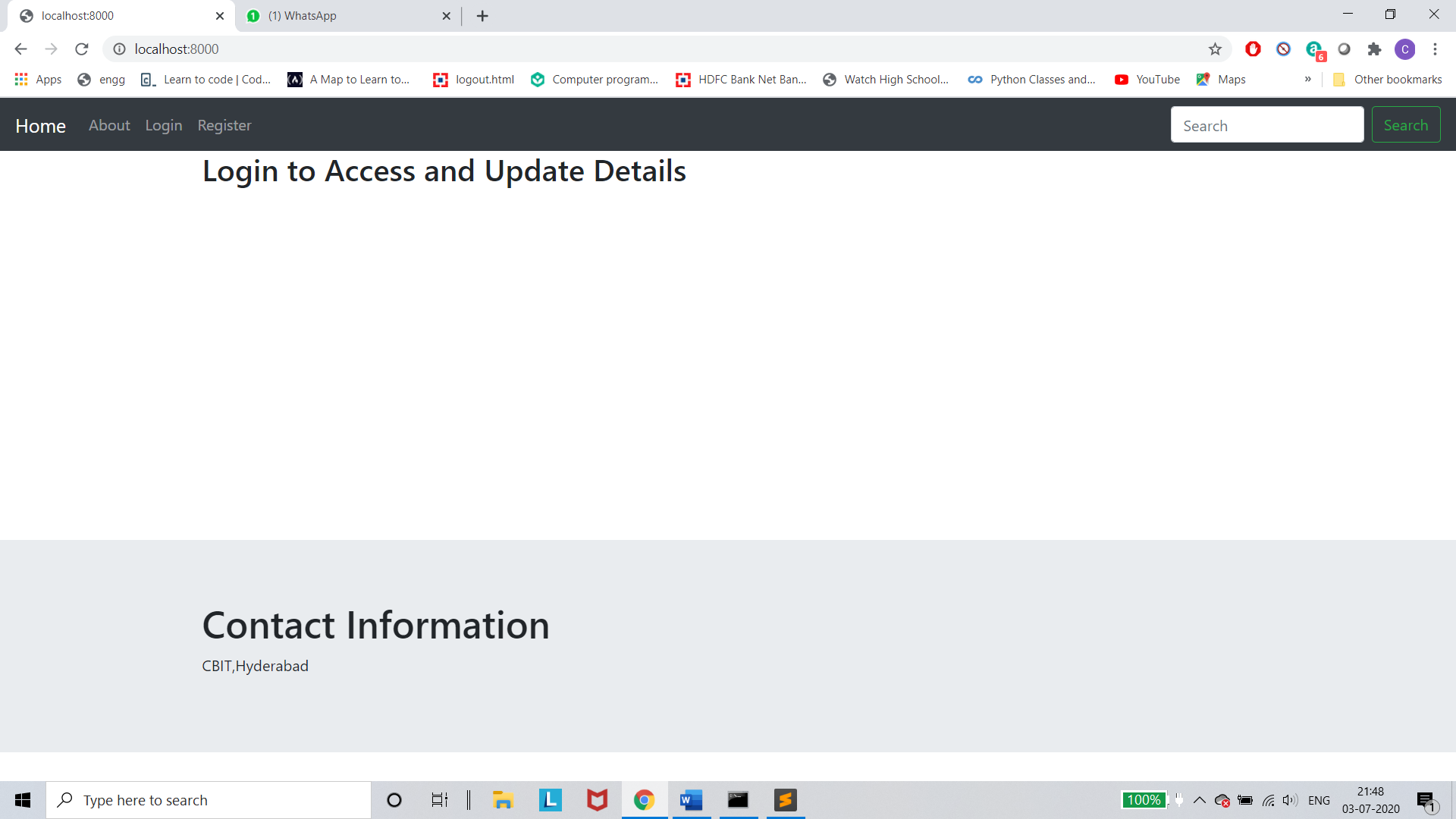
api.add\_resource(Datestatsforann,'/anndate')

api.add\_resource(userupdate,'/userupdatein')

if \_\_name\_\_ == "\_\_main\_\_":

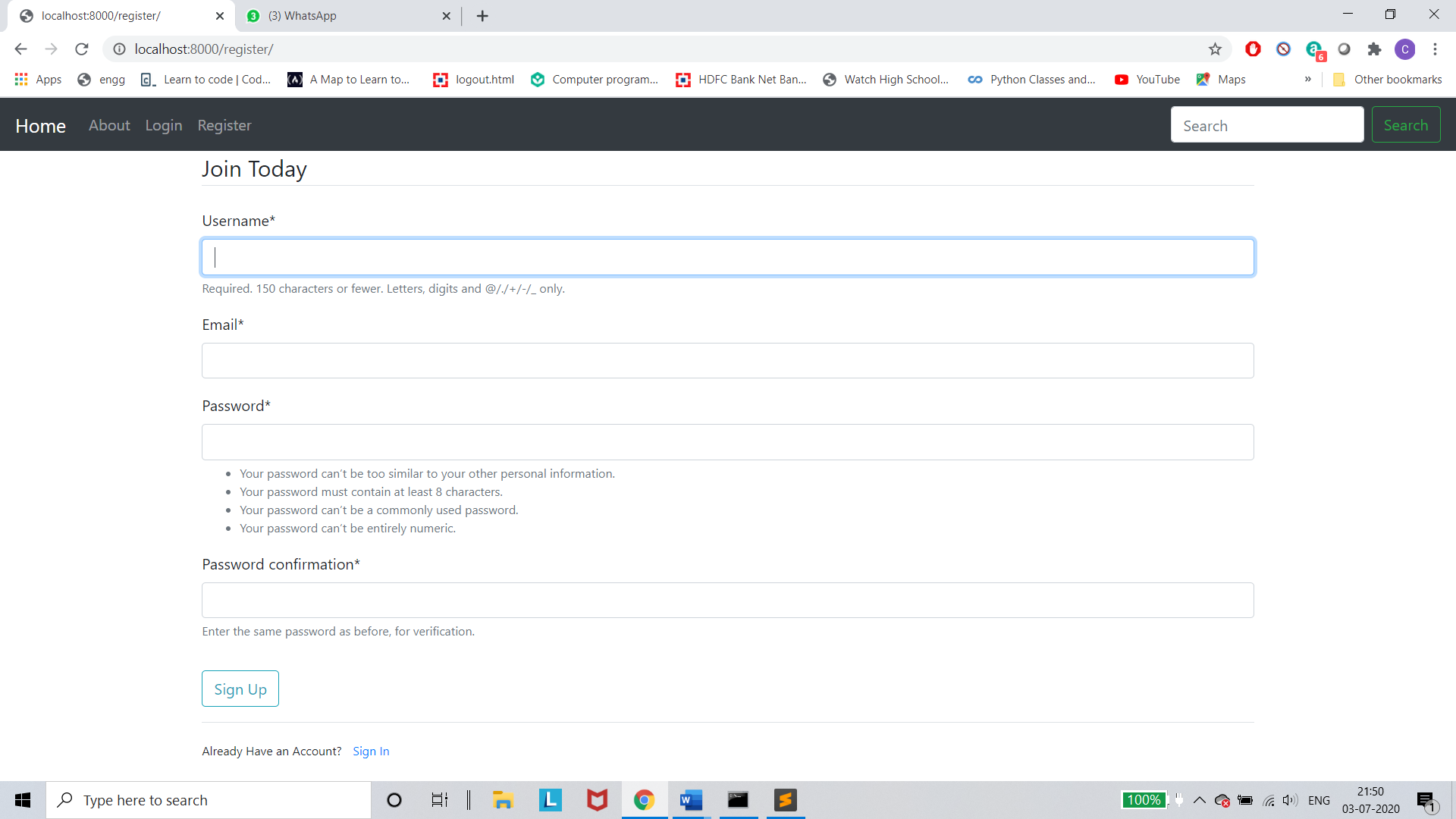
app.run()

# RESULT AND DISCUSSIONS



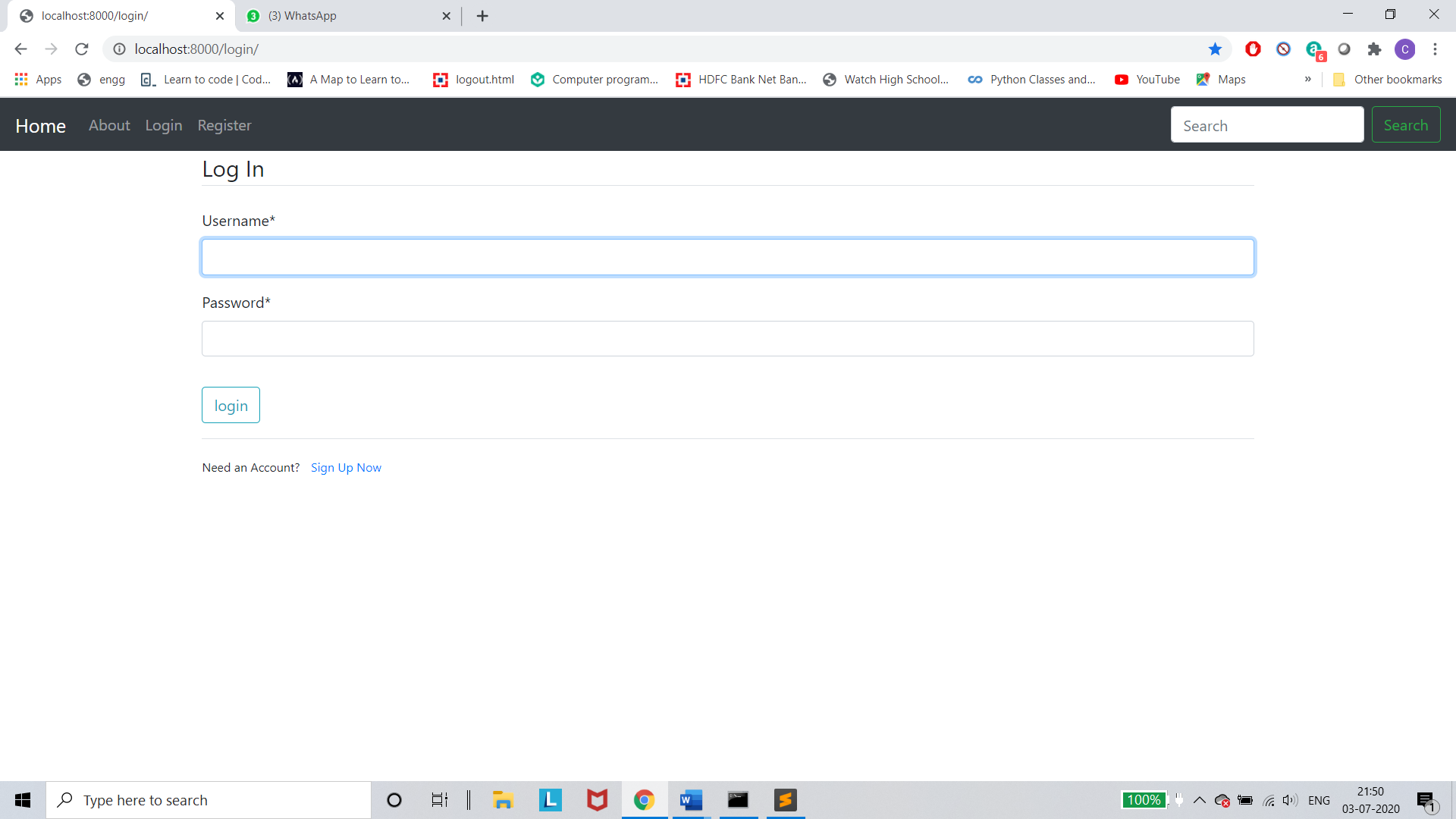
## Fig 5.1 – Home Page

This figure shows the home page of the admin website



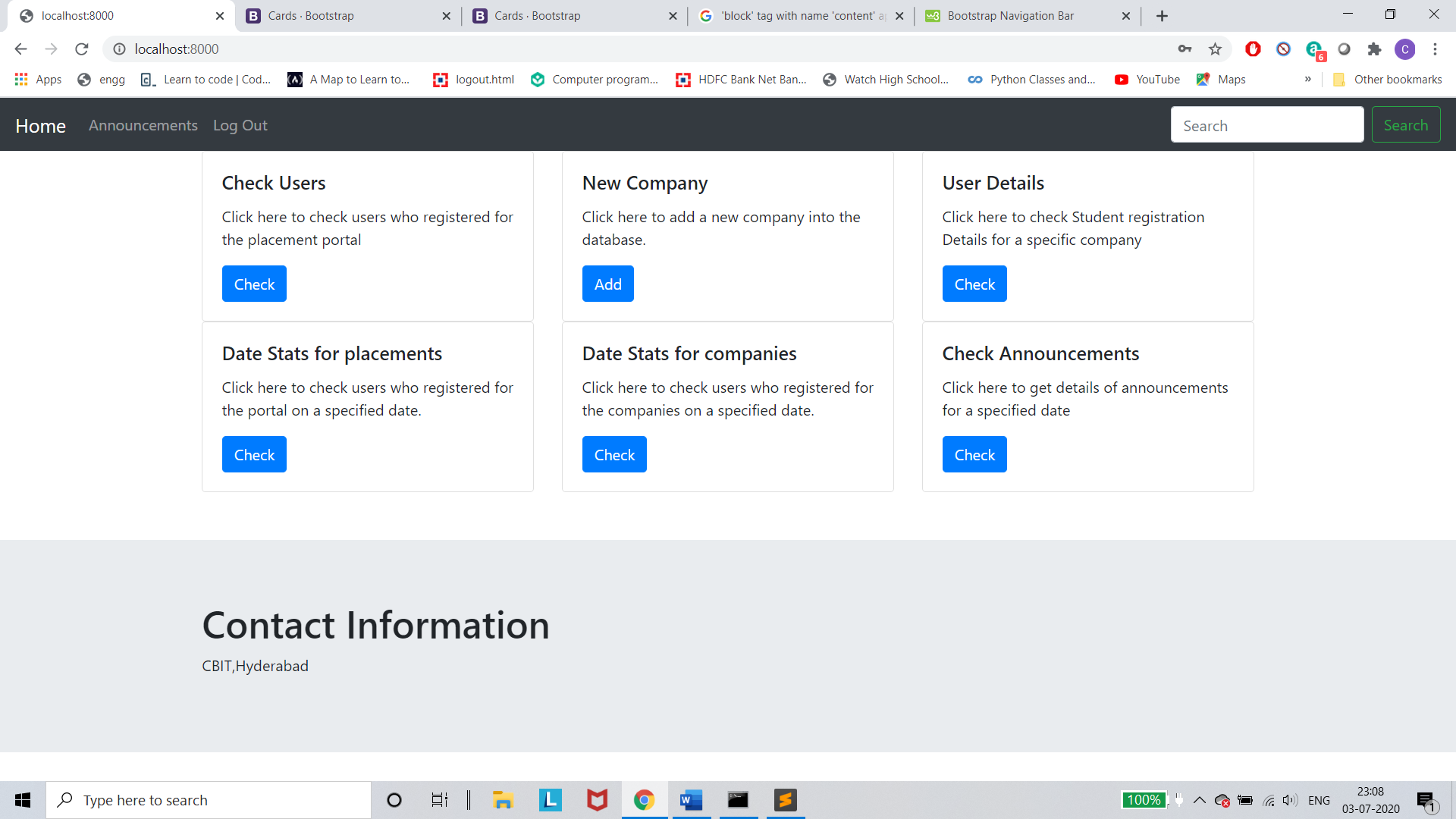
## Fig 5.2 – Register Page

This figure shows the registration page of the admin



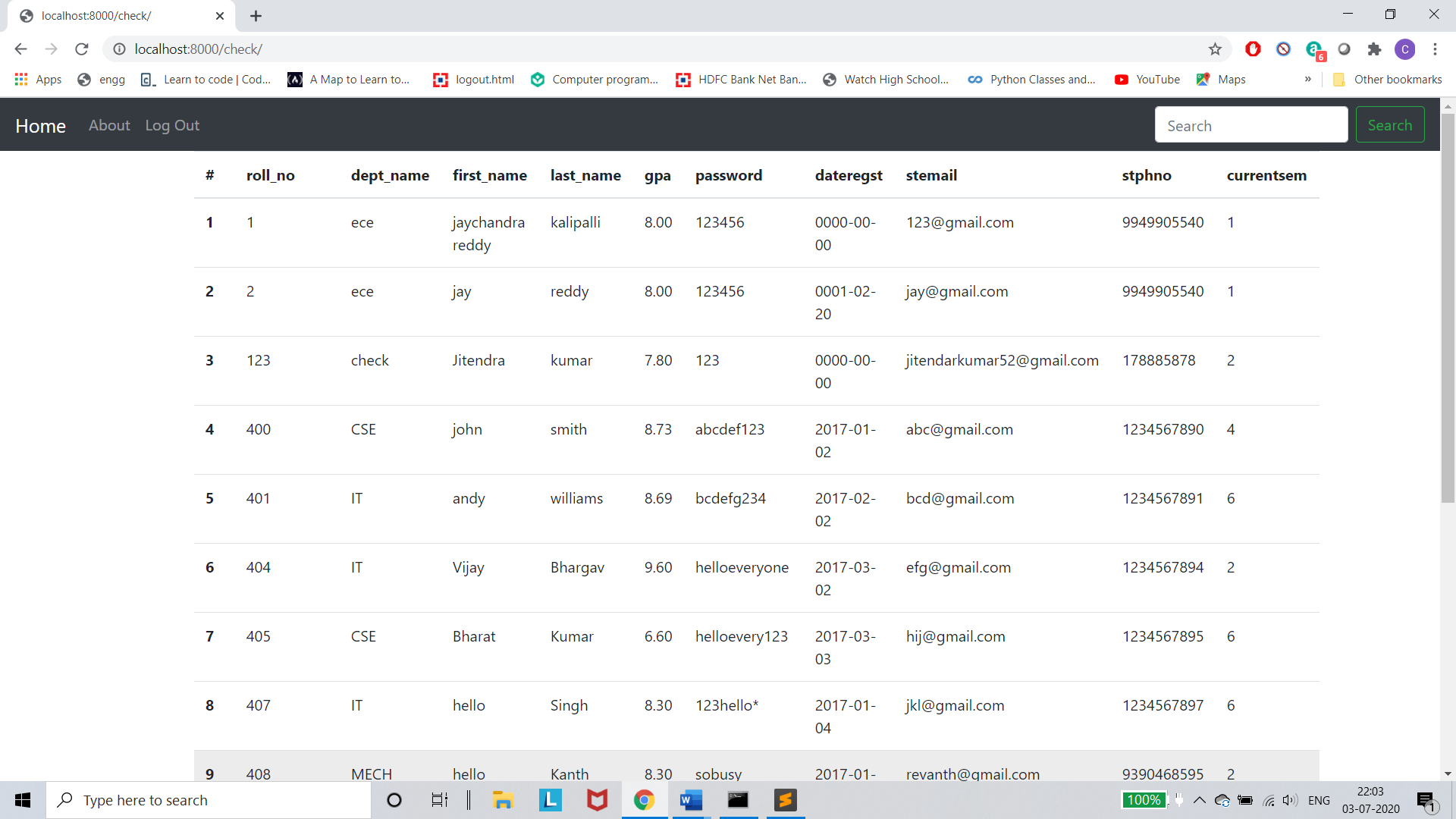
## Fig 5.3 – Login Page

This figure shows the login page of the admin site



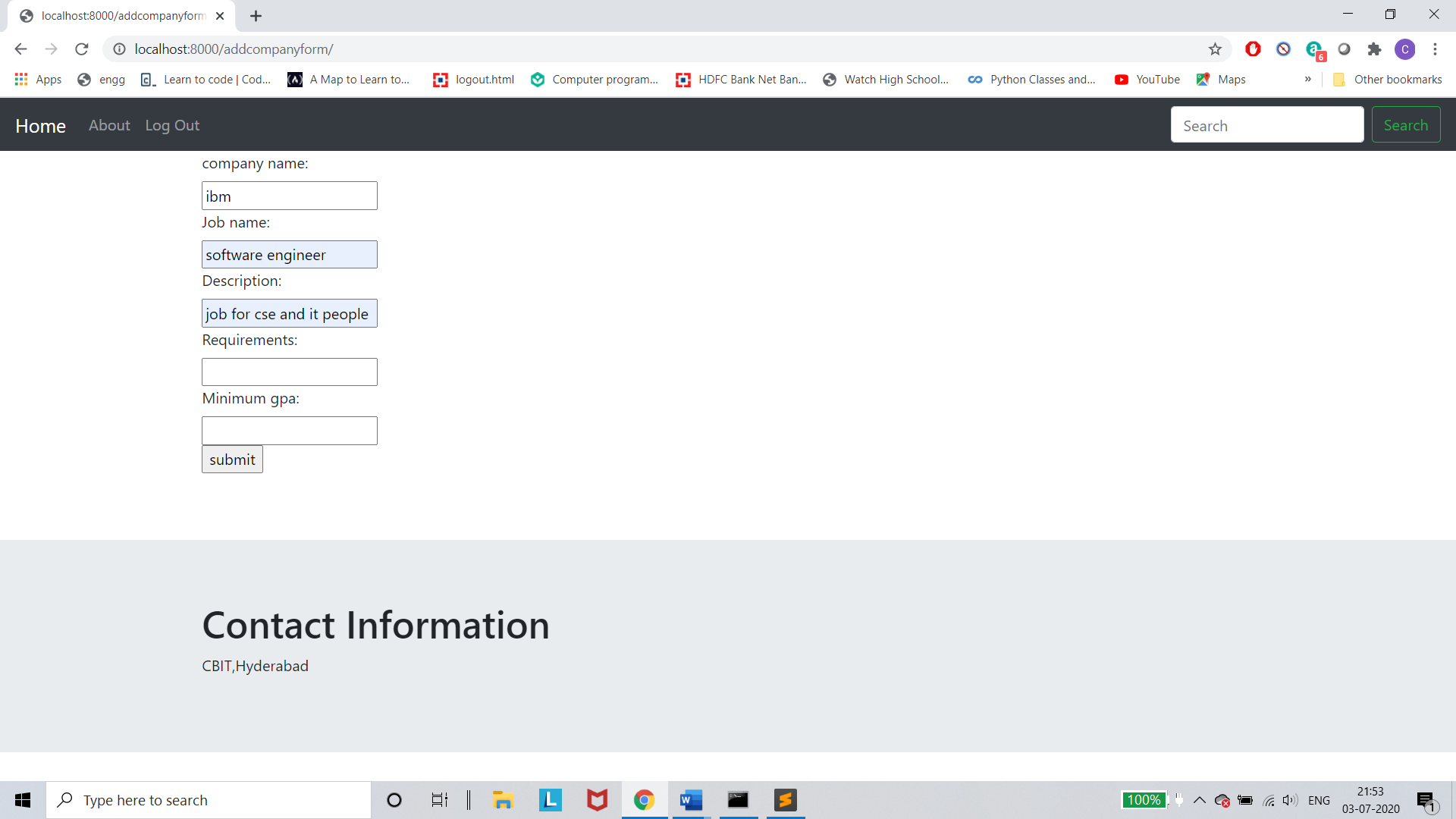
## Fig 5.4 – Home Page

This figure shows the home page of the admin site after logging in



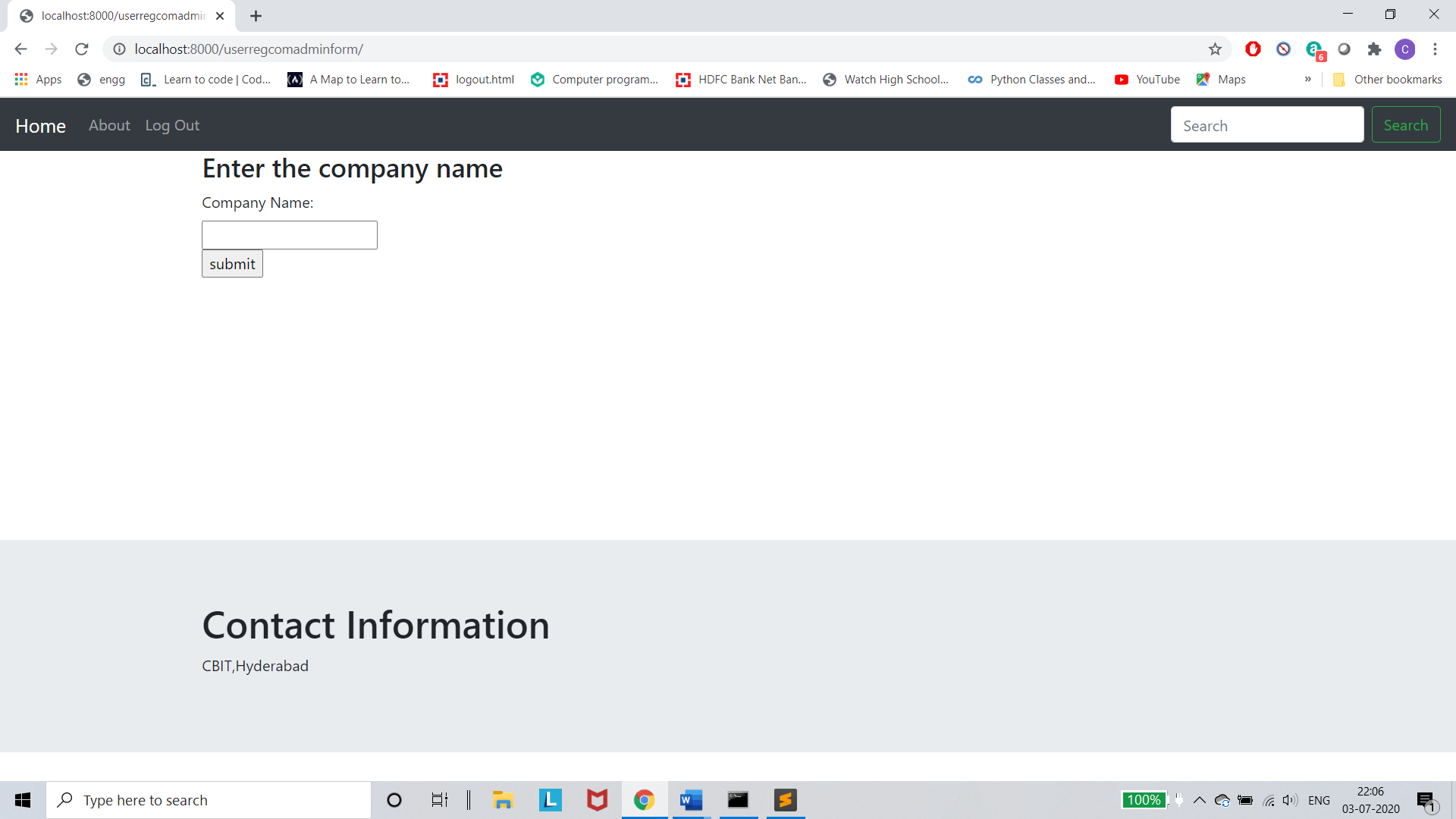
## Fig 5.5 – Check users

This figure shows users who registered for the portal



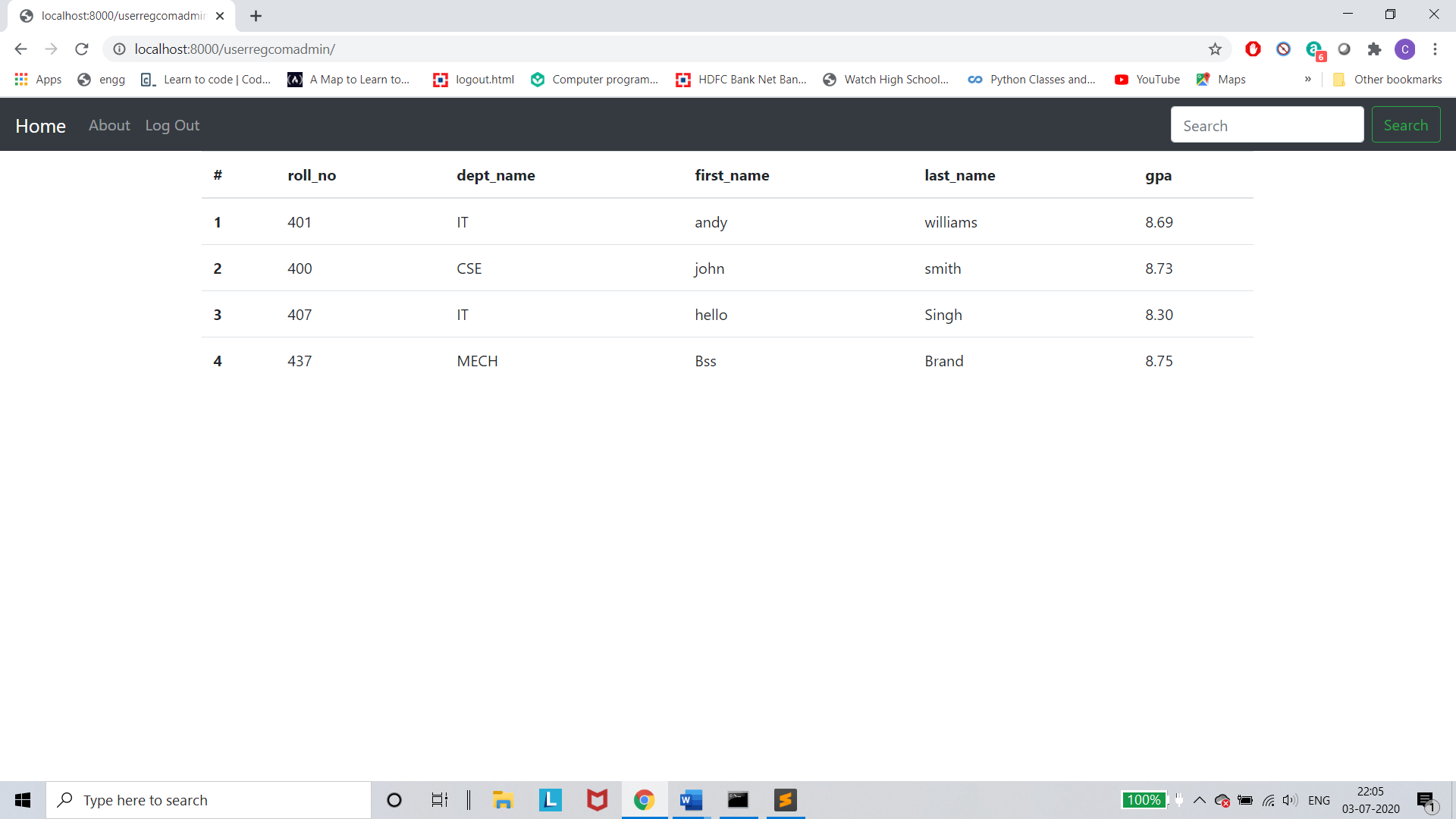
## Fig 5.6 –Add company

This figure shows form for adding a new company



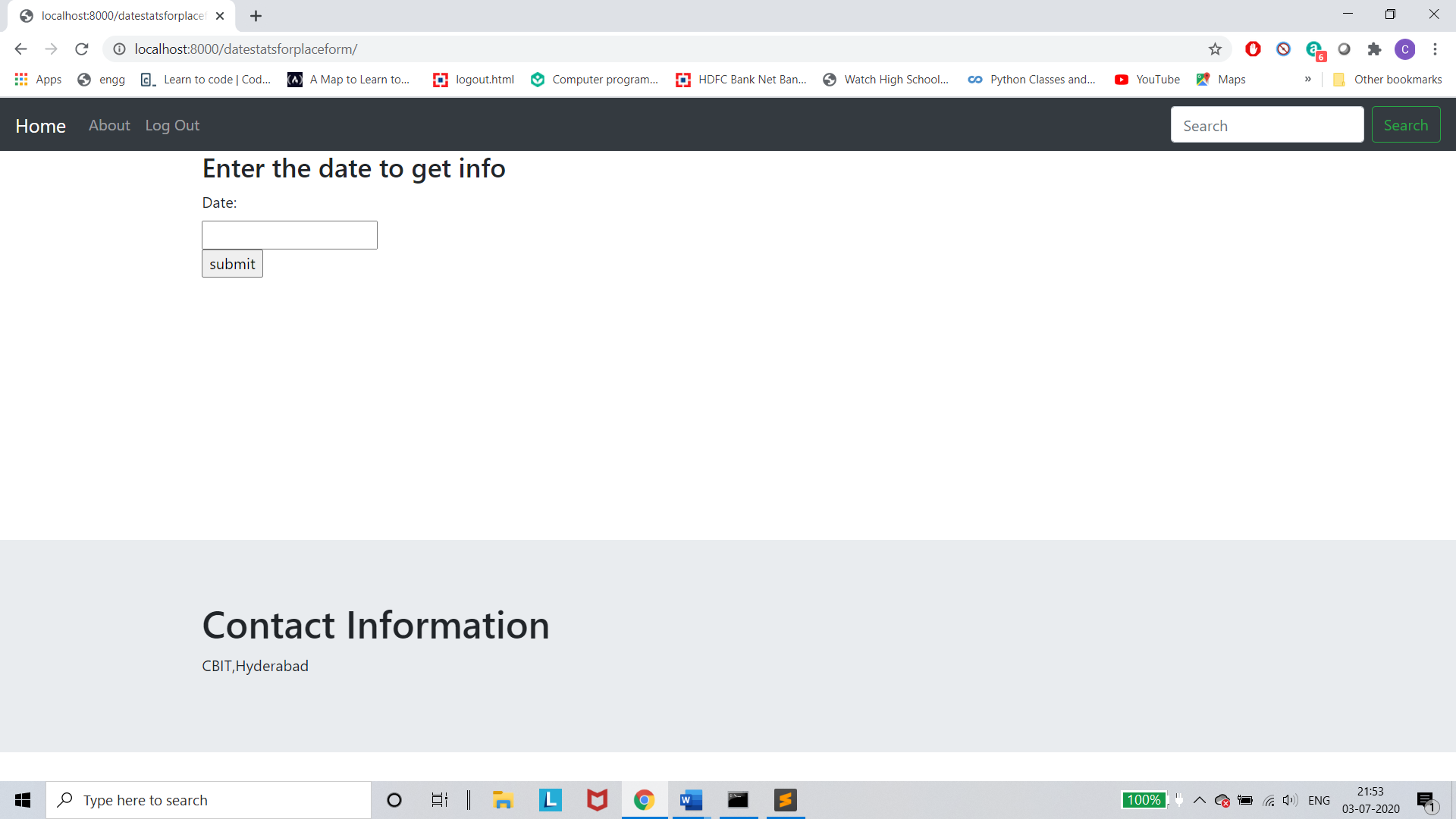
## Fig 5.6 –check company

This figure shows the form to check users for a company



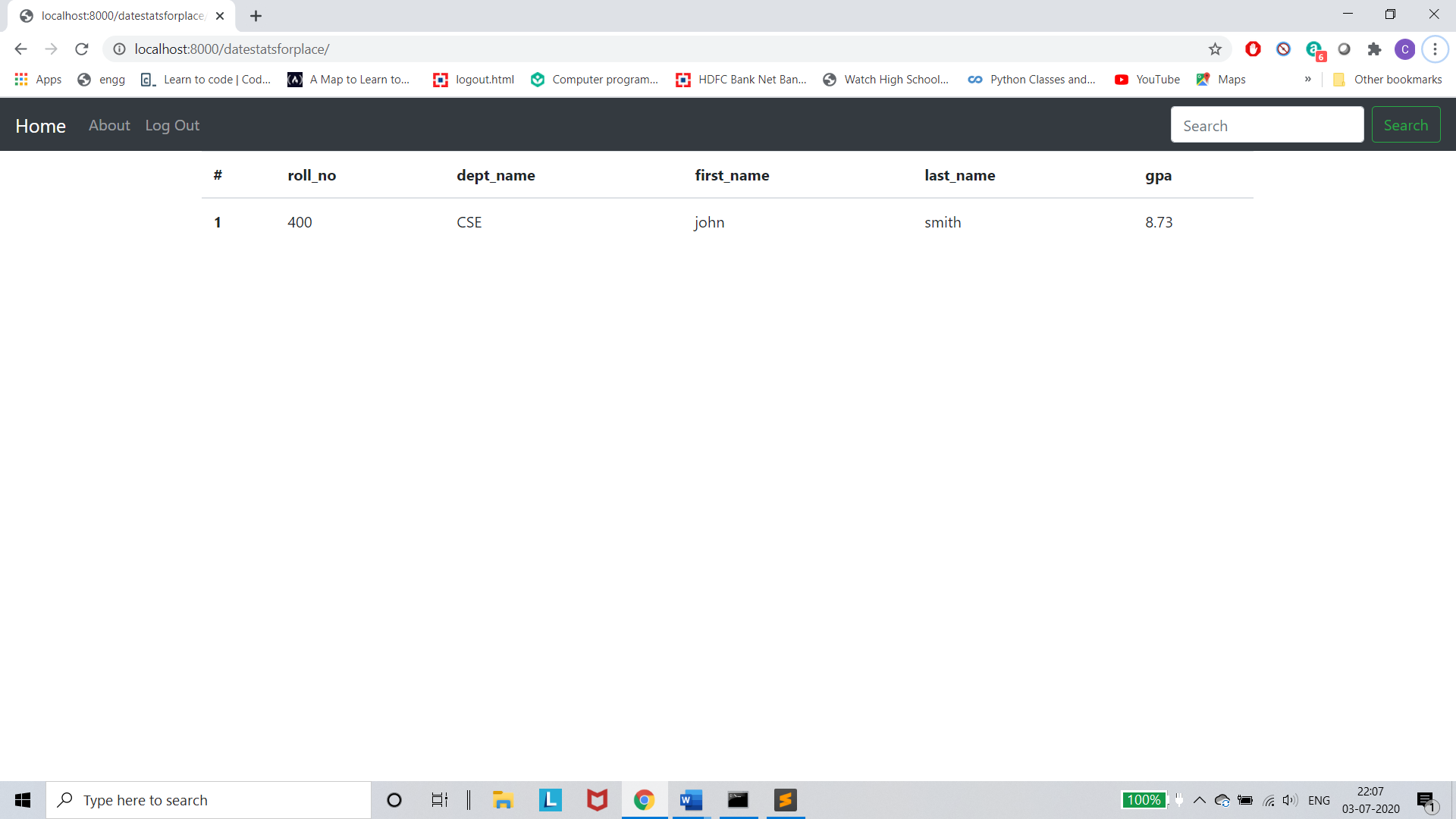
## Fig 5.6 –Details

This figure shows the details of users for a company



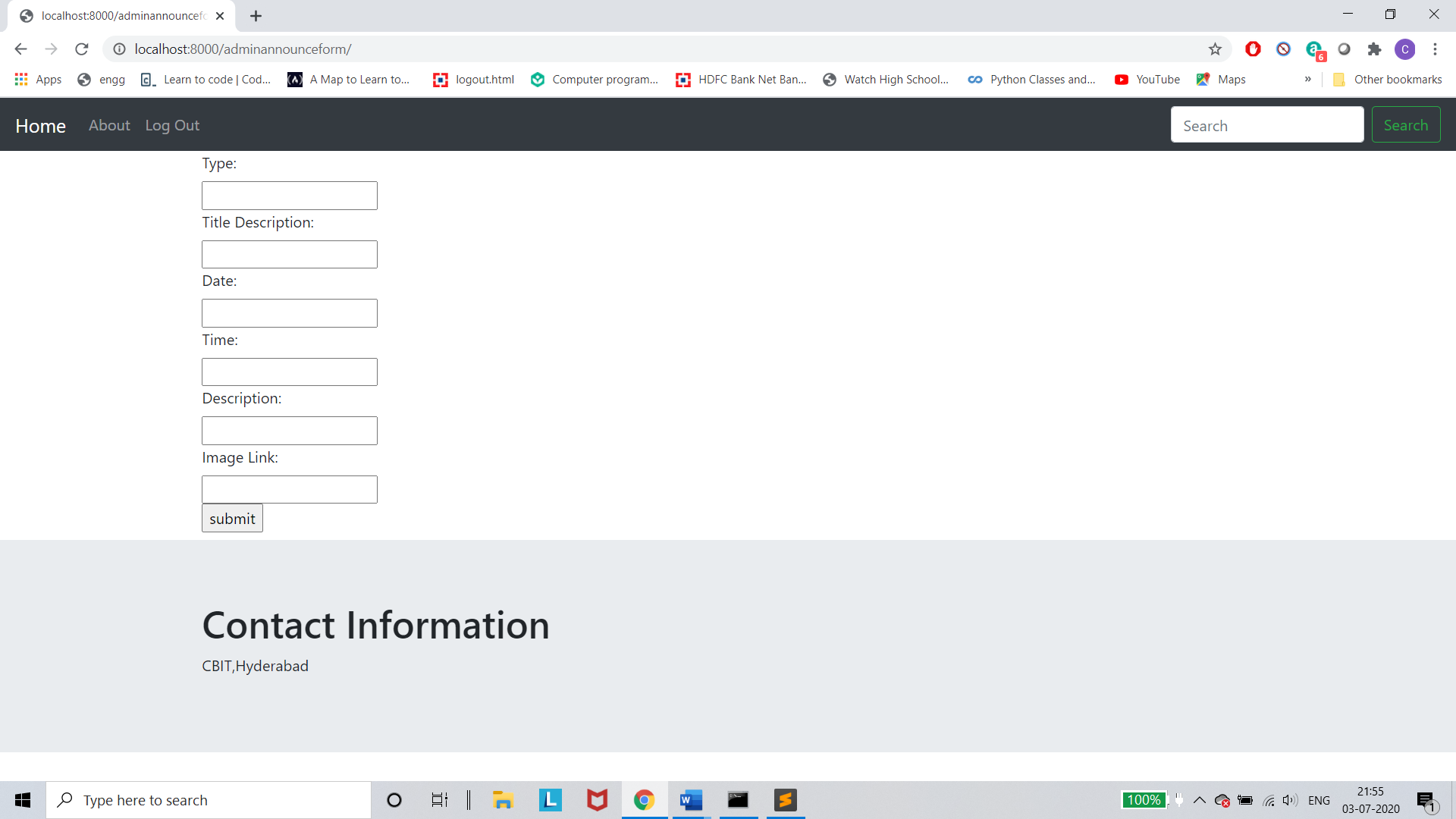
## Fig 5.6 –Datestats

This figure shows the form for getting date statistics



## Fig 5.6 –Datestats

This figure shows date statistics



## Fig 5.6 –Announcements

This figure shows the form to add announcements into the database

# Conclusion and Future Work

## Conclusion

Therefore, what we can conclude is that this sort of provision is necessary because of the flexibility of the placement portal and it’s utility which makes data gathering and data management a lot easier. Safety is given at most importance and securing of is done.

## Limitations

In our project, we are limited to some features like conducting online examinations and assessing the result , displaying results or sending notifications to the users whenever required .

## Future Work

In the Future scheduling a web conference /meeting or an online examination is done. Also notifications to the user will be sent whenever there is an new update on the company or an announcement.

# REFERENCES

1. [<https://pypi.org/project/requests/>]
2. [<https://www.w3schools.com/>]
3. [<https://flask.palletsprojects.com/en/1.1.x/>]
4. [<https://docs.djangoproject.com/en/3.0/>]
5. [[https://getbootstrap.com/]](https://getbootstrap.com/%5d)