EVENT REGISTRATION WEB PAGE USING AWS

A PROJECT REPORT
(Project Work I Phase I)

Submitted by

SIVAKUMAR P
18CSR190

SYED JAMAL HARRIS R
18CSR215

SUNDARESHWAR V A
18CSR206

in partial fulfilment of the

requirementsfor the award of the

degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMMUNICATION AND COMPUTER SCIENCES

Estd : 1984

KONGU ENGINEERING COLLEGE

(Autonomous)

PERUNDURAI ERODE – 638 060

MAY 2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERINGKONGU ENGINEERING COLLEGE

(Autonomous)

PERUNDURAI ERODE 638060

MAY 2021

BONAFIDE CERTIFICATE

This is to certify that the Project report entitled **EVENT REGISTRATION WEB PAGE USING AW**S is the bonafide record of project work done by SIVAKUMAR P (Register No.:18CSR190), SYED JAMAL HARRIS R (Register No.:18CSR215) and SUNDARESHWAR V A (Register No.:18CSR206) in partial fulfilment of the requirements for the award of the Degree of Bachelor of Engineering in Computer Science and Engineering of Anna University, Chennai during the year 2020- 2021.

SUPERVISOR **HEAD OF THE DEPARTMENT**

(Signature with seal)

Date:20.05.2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
KONGU ENGINEERING COLLEGE
(Autonomous)
PERUNDURAI ERODE – 638060
MAY 2021

DECLARATION

We affirm that the Project Report title **EVENT REGISTRATION WEB PAGE USING AWS** being submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering is the original work carried out by us. It has not formed the part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Date: 20.05.2021

SIVAKUMAR P
(Reg. No.:18CSR190)

SYED JAMAL HARRIS R
(Reg. No.:18CSR215)

SUNDARESHWAR V A
(Reg. No.:18CSR206)

I certify that the declaration made by the above candidates is true to the best of my knowledge.

Name and Signature of the Supervisor with seal

Date: 20.05.2021

ABSTRACT

The objective of the project " Event Registration Web page using AWS" is to create a web page that contains a registration form for a college event. This web page is to develop an event registration form for a college to maintain, update and process the data related to that college.

The problem in the existing system is the performance. In the existing system details are maintained and managed without Cloud computing. Without Cloud computing, maintenance of details is very difficult to handle.

This system is implemented to reduce the costs and to improve the scalability and performance. It will prepare reports to aid in smooth and speedy functioning of event activities. Through the web page students can easilyenroll in the college event. The main requirements for a webpage are registering the events with use of Django Framework and AWS EC2. It is an application which contains information about the events conducted by the college.

ACKNOWLEDGEMENT

We express our sincere thanks and gratitude to **Thiru.P.SACHITHANANDAN** our beloved Correspondent and all other philanthropic trust members of Kongu Vellalar Institute of Technology Trust who have always encouraged us in the academic and co- curricular activities.

We are extremely thankful with no words of formal nature to the dynamic Principal **Dr. V. BALUSAMY, M.Tech., Ph.D.,**for providing necessary facilities to complete our work.

We would like to express our sincere gratitude to our respected Head of the Department **Dr. N. SHANTHI M.E., Ph.D.,** for providing necessary facilities.

We extend our thanks to **Ms.M.SANGEETHA B.E.,M.E.,** Assistant Professor [SRG], Computer Science Engineering, Project Coordinator for her encouragement and valuable advice that made us carry out the project work successfully.

We extend our gratitude to our Supervisor **Dr.N.KRISHNAMOORTHY M.E., Ph.D.,** Associate Professor, Computer Science Engineering, for his valuable ideas and suggestions, which have been very helpful in the project. We are grateful to all the faculty members of the Computer Science and Engineering Department, for their support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

AWS	Amazon Web Services
CSS	Cascading Style Sheet
EC2	Elastic Cloud Computing
HTML	Hypertext Markup Language
PC	Personal Computer

CHAPTER 1

1.1 INTRODUCTION

Event Registration web page using AWS for a college can be a tool to create a communication interface among the student who wish to enroll his name to an event. They can provide a better platform for onlineservicing experience. The application is very useful where the students can directly get their work done from home by login through this system. The application reduces lot of work load for student as well as website admin. By this system the admin can get their registration status in the website itself. All students can access the web page based on the role they can have separate functionality. Only the admin of the web page who works in the college control over the website and they can able to monitor students registering in the website. There is no need of installation of website from the digital distribution systems.

The administrator can add images, upload events. The communication happens by providing feedback about the company. The student can view the web page. The webpage offers a complete automation, reduce the paper based records and make the user to feel convenient.

1.2 EXISTING SYSTEM

The existing system is to maintain all the activities manually in their own PC. There is lot of difficulties available in the existing system. It is so hard to maintain a large amount of data. So there rise the need for a proposed system to rectify the drawbacks of the existing system and so it is cloud computerized. The admin needs the serverless computing to maintain the data. There are no difficulties in maintaining the data. Such users will be easy to maintain the data, events happening and they can easily maintain with the help of cloud computing.

1.3 SYSTEM STUDY

In the application, there is a user who is the student involved in registration by using the web page. They having one login provided for student to select the required options and enroll in the event. Administrator has the highest privilege compared to others. Administrators have the authority to view all details of student using the web page and have the responsibilities like, save the events and remove the details who left the event. students have the lowest privilege they can only upload the Contents and view the registration web page.

1.4 OBJECTIVE

Main objective of the proposed system is to store the details about competition to establish the automation of maintaining overall details of the company through the web-based platform. The website includes contact details, participating event, year of study, name and roll no of his team members, department, section and their participating event.

The implementation of login modules allows only admin to view, update and modify their details. The web page is totally controlled and takeover by the user. It will be risk free which can be accessed from any smart devices. Web pages should be able to made available when needed and not available when not needed at any times.

1.5 SCOPE

The web-based platform for the college project will be a great help for the administrator to maintain the details of the customers. In traditional method, the college manually maintain the details without cloud computing. The project eliminates the paperwork, human faults, manual delay and speeds up the process of adding, updating and retrieving project details, events happening in the college. The proposed system helps users to get an idea about the company. The new user can visit the web page, the students can view the events conducted in the college and can give their response to the college. The application helps to provide an overall viewabout the event to the students using the web site. The application is developed such that above said enhancements can be integrated with current modules.

CHAPTER 2

GENERAL DESCRIPTION

2.1 PROJECT PERSPECTIVE

The main aim of the proposed system is to develop and the college to maintain the details of events, student name, participation details with their other corresponding details. The websitealso consists contact details, roll no, year and department of the corresponding participating student of the college.

Admin has the whole control over the web site and can periodically monitorthe people using this web site. Admin can have their control over their participation status and share it with the college. This will be very time effective, user friendly, interactive and provide student a good opinion about the event held in college.

2.1. USER CHARACTERISTICS

The basic functionalities of the student to enter the valid and correct information while creating or updating their details. The students need not to be technically expertise to use this application. The users with simple knowledge on web can easily use the application.

2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS

2.3.1 CONSTRAINTS

● Operating system	:	Windows with 8 GB RAM
● Coding language	:	HTML
● Framework	:	CSS, Django (Python)
● Cloud Computing Services:		AWS(EC2)

2.3.2 HTML

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

2.3.3 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.

CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics

2.3.4 DJANGO (PYTHON)

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel.

2.3.5 AWS EC2

Amazon Elastic Compute Cloud (EC2) is a part of AWS cloud-computing platform, Amazon Web Services (AWS), that allows users to rent virtual computers on which to run their own computer applications. EC2 encourages scalable deployment of applications by providing a web service through which a user can boot an Amazon Machine Image (AMI) to configure a virtual machine, which Amazon calls an "instance", containing any software desired. A user can create, launch, and terminate server-instances as needed, paying by the second for active servers – hence the term "elastic". EC2 provides users with control over the geographical location of instances that allows for latency optimization and high levels of redundancy.

CHAPTER 3

REQUIREMENTS

3.1 FUNCTIONAL REQUIREMENTS

A functional requirement describes a particular behavior of function of thesystem. Authorization levels, authentication and customer contents are the requirements of the system.

Authorization level

There are one levels of authentication in the application. They are admin the highest priority among all the users. Admin can upload the images, events and project application. Add or remove the customer and project details. The next priority is givento the student they can upload their information, upload the particular requirements and perform the participation status. The admin and student can interact with each other technically.

Authentication

The admin can login without any previous registration because the admin can directly store their details in the database. The student can login using the link generated by AWS provided by the college.

Customer Contents

The student contents are uploaded by the student and are displayed in the participation status. The corresponding images are also uploaded by the administrator.

3.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements describe how the application should behave depending upon the environment. Performance, reliability and security

Security

The subsystem should provide a high level of security and integrity of the data held by the system, only authorized personnel of the firm can gain access to the company's secured page on the system.

Reliability

Reliability is the ability of a system to perform its intended functions and stated conditions and behave consistently in a user acceptable manner and decrease the hardware failure.

Performance

The system should have high performance rate when users view the web page and should be able to provide feedback or response within a short time span.

3.3 USER INTERFACE

User interface elements include the checkboxes, text fields, dropdown lists by which student who visits this web page of this can easily interact in it. This interface helps to do all operation in application. Without the interfaces no one can interact with the web page. All actions are done using interfaces by student who can interact with application.

CHAPTER 4

PROPOSED WORK

4.1 ARCHITECTURAL DESIGN

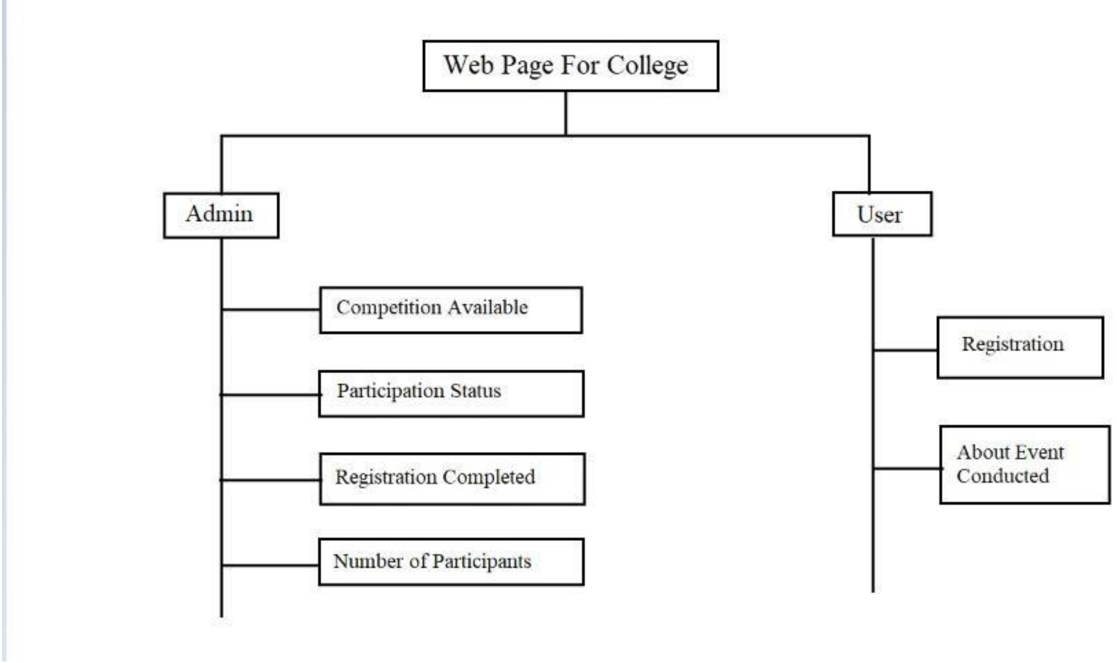


Figure 4.1 Architecture Flow

4.2 MODULE DESCRIPTION

There are two modules in the application, one is User Interface module and the other is database module

4.2.1 USER INTERFACE MODULE

User interface has one module such as admin.

ADMIN LOGIN:

The admin can perform the following functionalities. The admin has the privilege to manage the whole details and upload, remove events from the web page.

- The administrators can his/her separate login.
- The admin can upload the details and events happening in the college to the website.
- The admin can view the details of student and participation details separately.
- The admin can add, and delete the details of any student.
- The admin can update the details of any student.
- The admin can interact with the student based on their feedback.

4.2.2 DATABASE MODULE

Database design is a second module. In database module, there are several tables like admin for storing and uploading events, storing participation details, signup for storing the registered student details. The tables are designed based on the fields used in user interface. The details provided by the user(student) are validated and thenstored in the database. User cannot delete, update or retrieve data from the database.

4.3. OUTPUT DESIGN

The details are retrieved whenever the student(user) wants to view the result. Output design shows the outputs obtained from the application based on the user’s input.

CHAPTER 5

RESULTS AND DISCUSSION

Building web page using AWS for the college is helpful for maintaining information about the event such us the name of the participants along with their details. The project eliminates the paperwork, human faults, manual delay and speeds up the process. The system provides transparency between user and admin.

CHAPTER 6

TESTING

VALIDATION TESTING

The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements.

TestCase1: Check whether details stored correctly.

TestCase2: Check whether details updated consistently.
TestCase3: Check whether all details can be viewed by admin, customers. TestCase4: Check whether problems in existing system is overcome or not. TestCase5: Check whether system properly respond to the user.

TEST ANALYSIS

The proposed system works properly for the above cases. The system validated and it satisfies the coding constraints.

UNIT TESTING

Unit testing is a software development process in which the smallest testable parts of an application called units are individually and independently scrutinized for proper operation. In this project each and every webpage Home, about us and services, portfolio, contact us modules are tested individually and it works fine.

INTEGRATION TESTING

Integration Testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. It occurs after Unit testing and before Validation testing. It is done to test the modules when integrated to verify that they workas expected i.e. to test the modules which are working fine individually does not have issues when integrated. In the project Integration testing is mainly applied in Events module.

VERIFICATION TESTING

The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of the phase.

TestCase1: UI design need to satisfy the user’s expectation.

TestCase2: Check whether system manages company details. TestCase3: Check whether the information are correctly stored or not.

Test Analysis:

The project manages the company details and information is stored correctly.

CHAPTER 7

CONCLUSION AN FUTURE WORK

Implementing an application “Building Web page using AWS” is surely be a profitable deal as this application helps to carry out task with ease and thereby reduces time and money on man power and materials. The proposed system helps users to get an idea about the company. This is an Open Source Application so that firm can edit and transform the website according to their needs. In future the project can be enhanced by incorporating the users query clarification and view the user’s reviews.

CHAPTER 8

APPENDICES

CODING

(views.py)

```
from django.shortcuts import render
from django.http import HttpResponse
from django.template import loader
from myapp.models import formcontact

def form(request):

    if request.method=="POST":
        name=request.POST['name']
        rollno=request.POST['rollno']
        email=request.POST['email']
        phone=request.POST['phone']
        part=request.POST['part']
        dept=request.POST['dept']
        year=request.POST['Year']
        sect=request.POST['sect']
        v1=False
        v2=False
        v3=False
        v4=False
        v5=False

        val=request.POST.getlist('tag[]')
        if '1' in val:
            v1 = True
        if '2' in val:
            v2 = True
        if '3' in val:
            v3 = True
        if '4' in val:
            v4 = True
        if '5' in val:
            v5 = True


        print(name,rollno,email,phone,part,dept,year,sect,v1,v2,v3,v4,v5)
        ins=formcontact(name=name,rollno=rollno,email=email, phone=phone,part=part,dept=dept,year=year,sect=sect,paperpresentation=v1,projectpresentation=v2,quiz=v3,coding=v4,nontechnical=v5)
        ins.save()
        print("Data saved sucessfully")
    return render(request, 'main.html')
```

url.py(myproject file)

```
from django.contrib import admin
from django.urls import path, include
from myapp import views
urlpatterns = [
    path('admin/', admin.site.urls),
    path('home/',include('myapp.urls')),]
```

models.py

```
from django.db import models

# Create your models here.
class formcontact(models.Model):
    name = models.CharField(max_length=25)
    rollno = models.CharField(max_length=25)
    email = models.EmailField()
    phone = models.CharField(max_length=25)
    part = models.CharField(max_length=300)
    dept = models.CharField(max_length=25)
    year = models.CharField(max_length=25)
    sect = models.CharField(max_length=25)
    paperpresentation = models.BooleanField(null=True)
    projectpresentation = models.BooleanField(null=True)
    quiz = models.BooleanField(null=True)
    coding = models.BooleanField(null=True)
    nontechnical = models.BooleanField(null=True)
```

(main.html)

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <link rel="stylesheet" href="index.css">

  <title>KEC Event Registration</title>
</head>

<body>
    <div class="Container">
        <form action="/formtemp/" method="post">
          {% csrf_token %}
          {{ form }}
          <div class="form-group">
            <h2 style="font-family: 'Roboto Condensed', sans-serif;font-weight: bold;">
              Event registration</h2>
            <!-- <label for="name">Team Caption Name : </label> -->
            <input type="text" id="name" name="name" placeholder="Enter Your Team Leader Name" required><br><br>
            <!-- <label for="rollno">Rollno : </label> -->
            <input type="text" id="rollno" name="rollno" placeholder="Enter Your Roll No" required><br><br>
            <!-- <label for="email"> Email id : </label> -->
            <input type="email" id="email" name="email" placeholder="user@emailid.com"
              pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{3}$" required><br><br>
            <!-- <label for="phone"> Phone no : </label> -->
            <input type="number" id="phone" name="phone" placeholder="phone no" pattern="[0-9]{10}"
              required><br><br>
            <!-- <label for="part">Participants Name & Rollno : </label><br> -->
            <textarea name="part" id="part" rows="3" cols="70"
              placeholder="Participant name Participant Rollno" required></textarea><br><br>
            <!-- <label for="dept">Dept : </label> -->
            <select name="dept" class="custom-select" id="inputGroupSelect02" required>
              <option selected>Select your Department Name*</option>
              <option value="cse">CSE</option>
              <option value="it">IT</option>
              <option value="ece">ECE</option>
              <option value="mts">MTS</option>
              <option value="eee">EEE</option>
              <option value="e&i">E&I</option>
              <option value="mech">MECH</option>
              <option value="chem">CHEM</option>
```

```
                            <option value="ft">FT</option>
                            <option value="auto">AUTO</option>
                            <option value="civil">CIVIL</option>
                        </select><br><br>
                        <!-- <label for="Event Registration">Year</label> -->
                        <select id="Year" name="Year">
                            <option selected>Select Year of study</option>
                            <option value="I">I</option>
                            <option value="II">II</option>
                            <option value="III">III</option>
                            <option value="IV">IV</option>
                        </select><br><br>
                        <label for="sect" style="font-size: 20px;font-family: 'Roboto Condensed', sans-serif;">Section : </label>
                        <input type="radio" id="A" name="sect" value="A">
                        <label for="I" style="font-family: 'Roboto Condensed', sans-serif;">A</label>
                        <input type="radio" id="B" name="sect" value="B">
                        <label for="II" style="font-family: 'Roboto Condensed', sans-serif;">B</label>
                        <input type="radio" id="C" name="sect" value="C">
                        <label for="C" style="font-family: 'Roboto Condensed', sans-serif;">C</label>
                        <input type="radio" id="D" name="sect" value="D" required>
                        <label for="IV" style="font-family: 'Roboto Condensed', sans-serif;">D</label><br></br>
                        <label for="event" style="font-size: 20px;font-family: 'Roboto Condensed', sans-serif;">Event : </label><br>
                        <input type="checkbox" id="v1" name="v1" >
                        <label for="v1" style="font-size: 20px;font-family: 'Roboto Condensed', sans-serif;">Paper Presentation</label></br>
                        <input type="checkbox" id="v2" name="v2">
                        <label for="v2" style="font-size: 20px;font-family: 'Roboto Condensed', sans-serif;">Project Presentation</label></br>
                        <input type="checkbox" id="v3" name="v3">
                        <label for="v3" style="font-size: 20px;font-family: 'Roboto Condensed', sans-serif;">Quiz</label></br>
                        <input type="checkbox" id="v4" name="v4">
                        <label for="v4" style="font-size: 20px;font-family: 'Roboto Condensed', sans-serif;">Coding</label></br>
                        <input type="checkbox" id="v5" name="v5">
                        <label for="v5" style="font-size: 20px;font-family: 'Roboto Condensed', sans-serif;">Non Technical</label></br></br>
                        <input type="submit" value="Submit">
                    </div>
                </form>
            <div class="svg-part">
                <div class="letter-part">
                    <div style="display: flex;flex-direction: column;">
                        <h5>Managed by</h5>
                        <h5>Kongu Engineering College</h5>
```

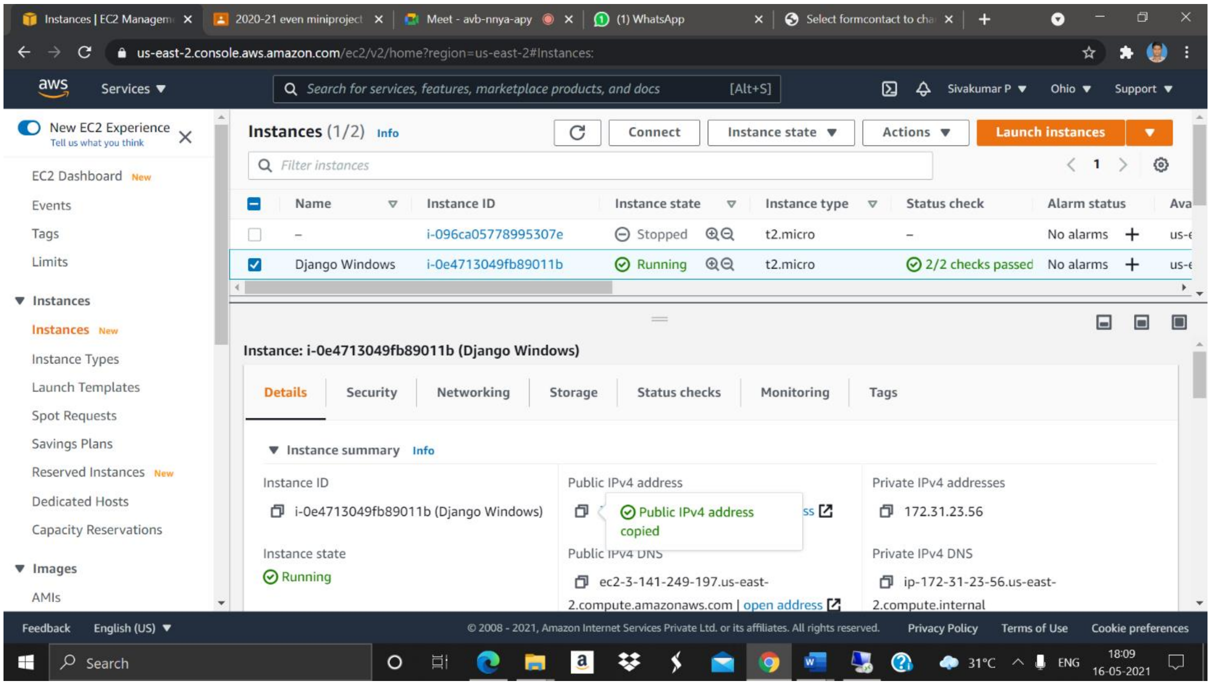
```
}



                </div>
            </div>
                    <img src="Group 24.svg" alt="svg" class="svgg">
        </div>
    </div>

</body>
</body>

</html>
```

(index.css)

```
@import url('https://fonts.googleapis.com/css2?family=Roboto+Condensed:wght@300;400;700&
display=swap');

*{
    margin: 0;
    padding: 0;
}

.Container{
    display:grid;
    grid-template-columns: 1fr 1fr;
    grid-row: 1fr;
    height: 100vh;
}

.form-group{
    display: block;
    grid-column: 1/2;
    padding:0px 320px ;
}
.form-group h2{
    text-align: center;
    margin-top: 52px;
    margin-bottom: 50px;
}

.svg-part{
    grid-column: 2/3;
    height: 100%;
```

```
   background-color: #EBF3FA;
}
.svgg{
   position: absolute;
   bottom: 0;
   margin: auto 50px;
}

input[type=text] {
   width: 300px;
   height: 40px;
   padding: 5px 15px ;
   font-size: 15px;
   border:1px solid black;
   color: black;
   border-radius: 5px;
   font-family: 'Roboto Condensed', sans-serif;
}
input[type=text]::placeholder{
   color: black;
}

input[type=email] {
   width: 300px;
   height: 40px;
   padding: 5px 15px ;
   font-size: 15px;
   border:1px solid black;
   color: black;
   border-radius: 5px;
   font-family: 'Roboto Condensed', sans-serif;
}
input[type=email]::placeholder{
   color: black;
}

input[type=number] {
   width: 300px;
   height: 40px;
   padding: 5px 15px ;
   font-size: 15px;
   border:1px solid black;
   color: black;
   border-radius: 5px;
   font-family: 'Roboto Condensed', sans-serif;
}
input[type=number]::placeholder{
   color: black;
}

textarea{
     width: 300px;
   height: 40px;
   padding: 5px 15px ;
   font-size: 15px;
   border:1px solid black;
   color: black;
   border-radius: 5px;
   font-family: 'Roboto Condensed', sans-serif;
}

textarea::placeholder{
   color: black;
}

select{
   width: 330px;
   height: 40px;
   padding: 5px 15px ;
   font-size: 15px;
   border:1px solid black;
   color: black;
   border-radius: 5px;
   font-family: 'Roboto Condensed', sans-serif;
}

input[type=radio] {
   font-size: 15px;
   font-family: 'Roboto Condensed', sans-serif;
}

input[type=submit] {
   width: 330px;
   height: 40px;
   padding: 5px 15px ;
   font-size: 15px;
   font-family: 'Roboto Condensed', sans-serif;
   background-color: #2B478B;
   border: none;
   color: white;
   border-radius: 5px;
}

.letter-part{
   position: absolute;
   top: 20px;
   right: 30px;
   background-color: #FAFAFA;
   font-size: 25px;
   padding: 20px;
   font-family: 'Roboto Condensed', sans-serif;
   font-weight: 300;
   border-radius: 5px;
   width: max-content;
   box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.061), 0 6px 20px 0 rgba(0, 0, 0, 0.034);
}
```

Screenshots



Instances (1/2) Info

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
☐	–	i-096ca05778995307e	Stopped	t2.micro	–	No alarms	us-e
☑	Django Windows	i-0e4713049fb89011b	Running	t2.micro	2/2 checks passed	No alarms	us-e

Instance: i-0e4713049fb89011b (Django Windows)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

▼ Instance summary Info

Instance ID: i-0e4713049fb89011b (Django Windows)

Public IPv4 address: Public IPv4 address copied

Private IPv4 addresses: 172.31.23.56

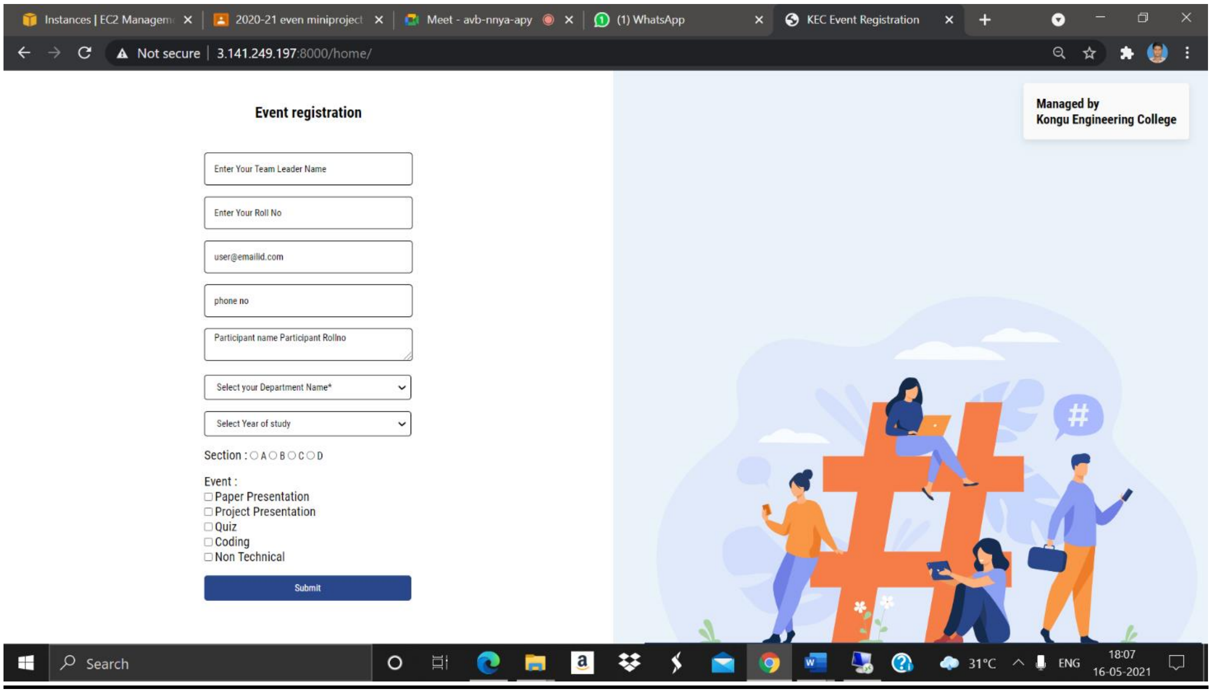
Instance state: Running

Public IPv4 DNS: ec2-3-141-249-197.us-east-2.compute.amazonaws.com | open address

Private IPv4 DNS: ip-172-31-23-56.us-east-2.compute.internal

© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

Figure 2: EC2 Running Console



Managed by
Kongu Engineering College

Event registration

Enter Your Team Leader Name

Enter Your Roll No

user@emailid.com

phone no

Participant name Participant Rollno

Select your Department Name*

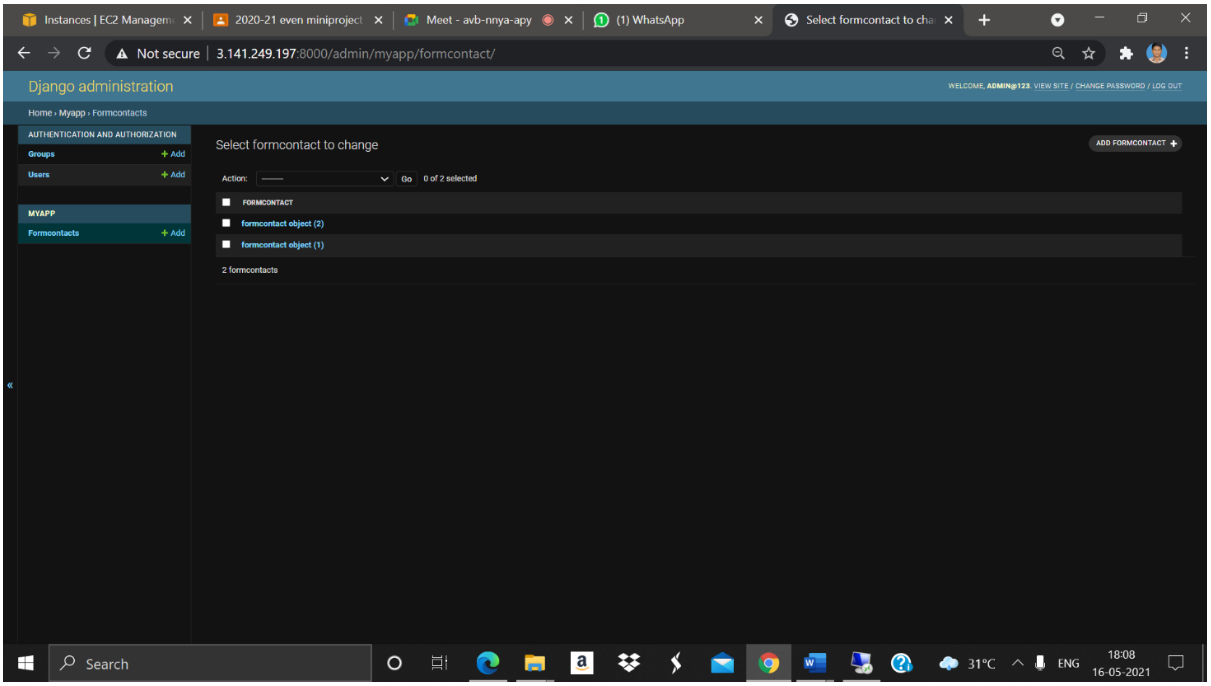
Select Year of study

Section : ○ A ○ B ○ C ○ D

Event :
□ Paper Presentation
□ Project Presentation
□ Quiz
□ Coding
□ Non Technical

Submit

Figure 3:User Page



Django administration

WELCOME, ADMIN@123. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Myapp › Formcontacts

AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add

MYAPP
Formcontacts + Add

Select formcontact to change

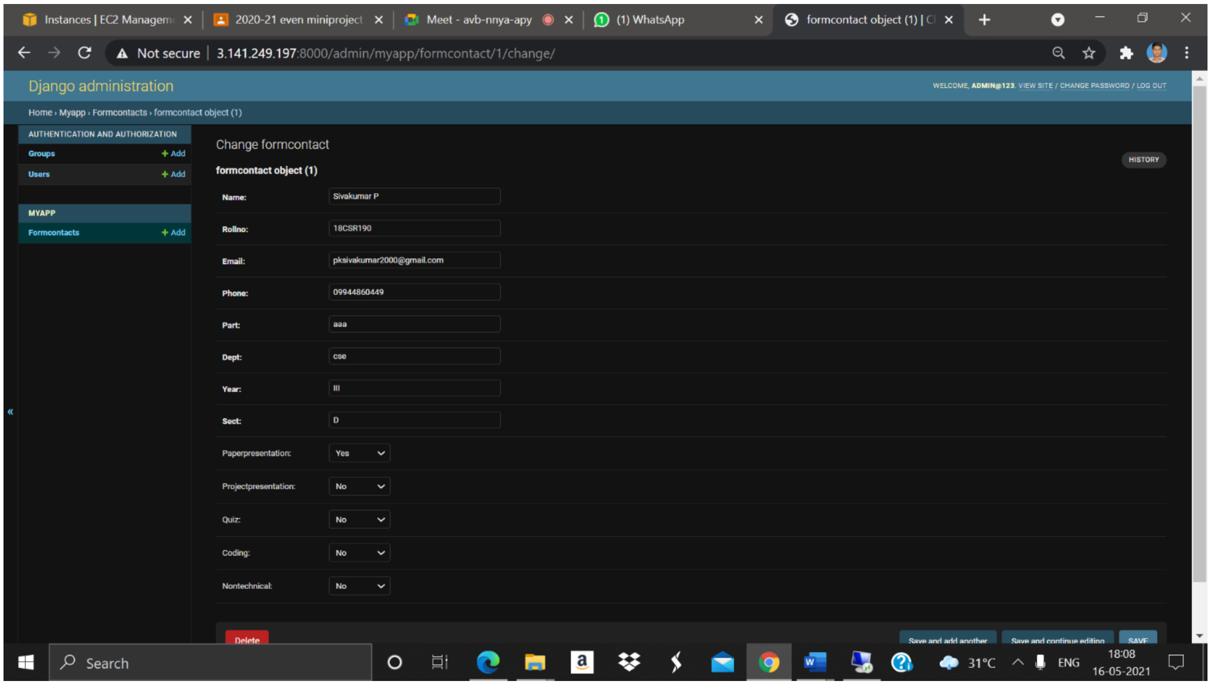
ADD FORMCONTACT +

Action: --------- Go 0 of 2 selected

□ FORMCONTACT
□ formcontact object (2)
□ formcontact object (1)

2 formcontacts

Figure 4: Admin Page



3.141.249.197:8000/admin/myapp/formcontact/1/change/

Change formcontact

formcontact object (1)

Name:	Sivakumar P
Rollno:	18CSR190
Email:	pksivakumar2000@gmail.com
Phone:	09944860449
Part:	aaa
Dept:	cse
Year:	III
Sect:	D
Paperpresentation:	Yes
Projectpresentation:	No
Quiz:	No
Coding:	No
Nontechnical:	No

Figure 5:Sored Data in Admin Page

CHAPTER 9

REFERENCES

[1] https://www.w3schools.com/html/

[2] https://www.w3schools.com/w3css/defaulT.asp

[3] https://www.tutorialspoint.com/django

[4] https://aws.amazon.com/ec2/

[5] https://stackoverflow.com/questions/19921705/how-do-i-access-my-django-app-running-on-amazon-ec2