Report
On
Python Django Internship

Student Name
Nishant Movaliya
(180160107042)

Faculty Mentor
R. N. Vaza

Submitted to

Department of Computer Engineering & Information Technology
Government Engineering College, Modasa

Year:2021

CERTIFICATE

This is to certify that **Nishant Movaliya (180160107042)**, of Computer Engineering has successfully completed the Summer Internship(3170001) during 26th May 2021 to 09th June 2021.

Signature of Faculty Mentor

R. N. Vaza

Signature of Head of Department

Minubhai Chaudhary

ACKNOWLEDGEMENT

I heartily want to express gratitude of thanks to my Guide, Mr. Rahul N. Vaza Sir, whose guidance, Supervision and support from the preliminary to the concluding level enabled me to complete complete Internship process. Under Sir's guidance, I was able to complete the internship Smoothly & with Ease. He Conducted Meeting with all the Members Individually to make the Internship Experinece Memorable. Also I am thankful to faculties such as – Anil Prajapati Sir, Jwalant Baria Sir, Respected H.O.D Sir of our college "Government Engineering College, Modasa" for clearing our every doubts related to this Summer Internship. Lastly, I offer our regards and blessings to all of those, engaged in creating such a Inetrnship Program from the GTU university team, due to which I could learnt a lot of new tools/technologies under the help such a program.

Signature of Student

Nishant Movaliya

(Enrollment No: 180160107042)

DECLARATION

I, hereby declare that the project/work submission is my own work and that, to the best of my knowledge and belief, it contains no code/material previously published or written by another person as a part of the completion of the Summer Internship.

Place: Government Engineering College, Modasa
Date: 16/06/2021

Signature of the Student

Nishant Prakabhai Movaliya
180160107042

Summer Internship Completion Certificate Provided by:

<u>Akash Technolabs</u>

Akash Technolabs
Web and Mobile App Development

CERTIFICATE
INTERNSHIP

This is to Certify that

Mr./Ms. Nishant Movaliya

has Succesfully completed 15 Days Internship in

Python Django at Akash Technolabs

During the Period of 26th May 2021 to 9th June 2021.

We wish you all the Best for future endeavours.

12-06-2021
Date

Akash Padhiyar
Mr. Akash Padhiyar
CEO, Akash Technolabs

Abstract

INTERNSHIP

At Akash Technolabs, Internship was based on developing DJANGO Webapp using python. Initially, Learning's about python working model such as OOP's in python, Module implementation in python, Function in python were given training. After covering the python module, we learnt about the Django framework for speed & rapid development of the project with ease & comfort. Django framework – the installation, creation of the project, setting up the repositories, URL routing, theme integration, CSS defining, MVT architecture pattern of Django, Analysis in the flow of data through MVT Architecture, setting up the required infrastructure, database integration, admin panel, Data retrieving using get & post method, multiple databases, crpf-token implementation, developing a model to set-up databases, CRUD-operation & several other implementations.

Index

CHAPTER-1

INTRODUCTION

INTRODUCTION

Starting with the Internship, started from Covering the fundamentals of python to learning & implementing the Django Framework in python. The deployment of the project in Django consisted of creating a new app for developing web app under a new project, applying the standard package settings in files of Django Project, Starting with implementing Hello World Program Basics & continuing to integrate HTML themes into the project along with integrating database by creating model & setting up admin panel.

DJANGO FRAMEWORK

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.

ADVANTAGES:

- Ridiculously fast
  Django was designed to help developers take applications from concept to completion as quickly as possible.

- Reaasuringly Secure
  Django was designed to help developers take applications from concept to completion as quickly as possible.

- Exceedingly scalable
  Some of the busiest sites on the Web leverage Django's ability to quickly and flexibly scale.

- Incredibly versatile
  Companies, organizations and governments have used Django to build all sorts of things — from content management systems to social networks to scientific computing platforms.

- Fully Loaded
  Django includes dozens of extras you can use to handle common Web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds, and many more tasks — right out of the box.

With Django, you can take Web applications from concept to launch in a matter of hours. Django takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

INTRO TO DJANGO

OBJECT-RELATIONAL MAPPER:

Define your data models entirely in Python. You get a rich, dynamic database-access API for free — but you can still write SQL if needed.

```
class Band(models.Model):
    """A model of a rock band."""
    name = models.CharField(max_length=200)
    can_rock = models.BooleanField(default=True)


class Member(models.Model):
    """A model of a rock band member."""
    name = models.CharField("Member's name", max_length=200)
    instrument = models.CharField(choices=(
            ('g', "Guitar"),
            ('b', "Bass"),
            ('d', "Drums"),
        ),
        max_length=1
    )
    band = models.ForeignKey("Band")
```

URLS & Views:

A clean, elegant URL scheme is an important detail in a high-quality Web application. Django encourages beautiful URL design and doesn't put any cruft in URLs, like .php or .asp.

To design URLs for an application, you create a Python module called a URLconf. Like a table of contents for your app, it contains a simple mapping between URL patterns and your views.

```
from django.urls import path
```

```
from . import views

urlpatterns = [
    path('bands/', views.band_listing, name='band-list'),
    path('bands/<int:band_id>/', views.band_detail, name='band-detail'),
    path('bands/search/', views.band_search, name='band-search'),
]

from django.shortcuts import render

def band_listing(request):
    """A view of all bands."""
    bands = models.Band.objects.all()
    return render(request, 'bands/band_listing.html', {'bands': bands})
```

TEMPLATES:

Django's template language is designed to strike a balance between power and ease. It's designed to feel comfortable and easy-to-learn to those used to working with HTML, like designers and front-end developers. But it is also flexible and highly extensible, allowing developers to augment the template language as needed.

```
<html>
  <head>
    <title>Band Listing</title>
  </head>
  <body>
    <h1>All Bands</h1>
    <ul>
    {% for band in bands %}
      <li>
        <h2><a href="{{ band.get_absolute_url }}">{{ band.name }}</a></h2>
        {% if band.can_rock %}<p>This band can rock!</p>{% endif %}
      </li>
    {% endfor %}
    </ul>
  </body>
```

```
</html>
```

FORMS:

Django provides a powerful form library that handles rendering forms as HTML, validating user-submitted data, and converting that data to native Python types. Django also provides a way to generate forms from your existing models and use those forms to create and update data.

```
from django import forms

class BandContactForm(forms.Form):
    subject = forms.CharField(max_length=100)
    message = forms.CharField()
    sender = forms.EmailField()
    cc_myself = forms.BooleanField(required=False)
```

AUTHENTICATION:

Django comes with a full-featured and secure authentication system. It handles user accounts, groups, permissions and cookie-based user sessions. This lets you easily build sites that allow users to create accounts and safely log in/out.

```
from django.contrib.auth.decorators import login_required
from django.shortcuts import render

@login_required
def my_protected_view(request):
    """A view that can only be accessed by logged-in users"""
    return render(request, 'protected.html', {'current_user': request.user})
```

ADMIN:

One of the most powerful parts of Django is its automatic admin interface. It reads metadata in your models to provide a powerful and production-ready interface that content producers can immediately use to start managing content on your site. It’s easy to set up and provides many hooks for customization.

```
from django.contrib import admin
from bands.models import Band, Member
```

```
class MemberAdmin(admin.ModelAdmin):
    """Customize the look of the auto-generated admin for the Member model"""
    list_display = ('name', 'instrument')
    list_filter = ('band',)

admin.site.register(Band)  # Use the default options
admin.site.register(Member, MemberAdmin)  # Use the customized options
```

INTERNATIONALIZATION:

Django offers full support for translating text into different languages, plus locale-specific formatting of dates, times, numbers, and time zones. It lets developers and template authors specify which parts of their apps should be translated or formatted for local languages and cultures, and it uses these hooks to localize Web applications for particular users according to their preferences.

```
from django.shortcuts import render
from django.utils.translation import gettext

def homepage(request):
    """
    Shows the homepage with a welcome message that is translated in the
    user's language.
    """
    message = gettext('Welcome to our site!')
    return render(request, 'homepage.html', {'message': message})
```

SECURITY:

Django provides multiple protections against:

- Clickjacking
- Cross-site scripting
- Cross Site Request Forgery (CSRF)
- SQL injection
- Remote code execution

CHAPTER-2

PROBLEM STATEMENT/DEFINITION

PROBLEM DEFINITION

In the Recent Times,it is very exhausting to code Regular HTML & CSS along with Javascript for Web-Development, it takes a long-time for designing the process & becomes exhaustive for the Web-Designers. The Problem statement is that if we can develop Web-Development using such a technology that provides fast & rapid development platform then we could solve the issue of such instances.

Here comes the role of framework- DJANGO based on python. With Django, you can take Web applications from concept to launch in a matter of hours. Django takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

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.

CHAPTER-3

TOOLS/TECHNOLOGIES

TOOLS INVOLVED:

- Visual Studio Code – Intergrated Development Environement for Executing & Running Written Program.
- Django – The Framework technology based on python.
- Pip Installer – The interface for installing required Libraries & framework.
- Sqlite3 Database – the database to work with in background.
- Github – for version control system

TECHNOLOGIES INVOLVED:

- Python 3.0+ Version
- HTML – For Writing code in HTML Language
- CSS – For Designing the HTML Code
- JAVASCRIPT – Scripting Language for Connection between pages.
- Sqlite3 database – For Quering & Inserting Data operations from Backend database.

DATABASE QUERY LANGAUAGE-SQL – For Quering & Inserting Data operations from Backend database.

CHAPTER-4

TIMELINE CHART/DAILY TASK

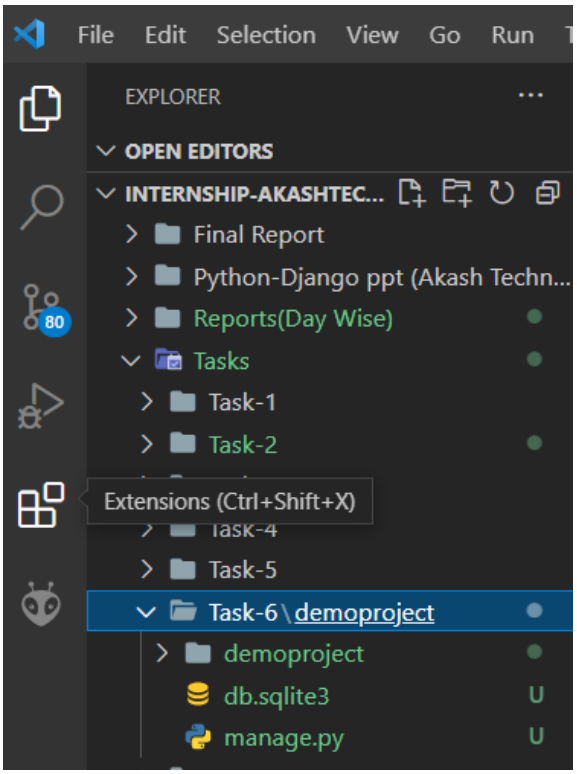
Task	WORK DONE	TOOLS USED
1	Basic HTML based : To Create Registration form using div/table. and also Github	Vs code, Html, CSS, github
2	Database Connection Creation & Query- Insert, Update, Delete.	Python, Jupyter Notebook, github
3	Python Programs- Average, factorial, Greatest_Num, Swap_Num, Less_100, Square_no, Greatest of 3, Smallest of 3, etc.	Python, Jupyter Notebook, github
4	Creating Function, Implementing Module & Operators in Python Anaconda.	Python, Jupyter Notebook, github
5	Implanting programs which is given by mem using object oriented concept.	Python, Jupyter Notebook, github
6	Install Django, Creating New Project in DJANGO, Setting up- Directory, Configuring files in settings.py & manage.py.Run and Display Browser Window	Django,Vs code, Command Promt, Python, github
7	Setting up- Directory, Configuring files in urls.py, settings.py , views.py Display Hello World Text in Browser in Django.	HTML,Django,Vs code, Command Promt, Python, github
8	Manage.py, apps.py, model.py, tests.py, urls,py – Creating Views in project, Registering them in urls – Routing URL & registering in settings of project configuration.	Django,Vs code, Command Promt, Python, github
9	Integrating HTML template in Django Project, Setting up CSS file & learning about implementing common page views & Creating form for accepting user data.Using POST method to get User Values and returning back on page.	My owned Template, Django, Vs code, python, Github
10	Implementing Models and Fetching Values using Function Based Views and Class Based Views. Integrating Database with Django & fetching data from user to display it.	Django, Sqlite3 DB,Vs code, python, Github

CHAPTER-5

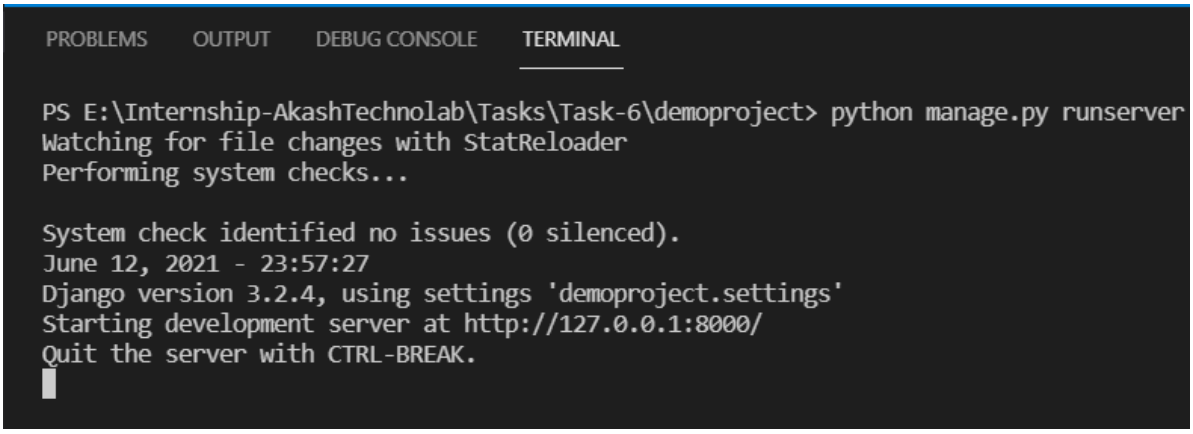
MODULE DEVELOPMENT/IMPLEMENTATION

Django Installation
Then create new project

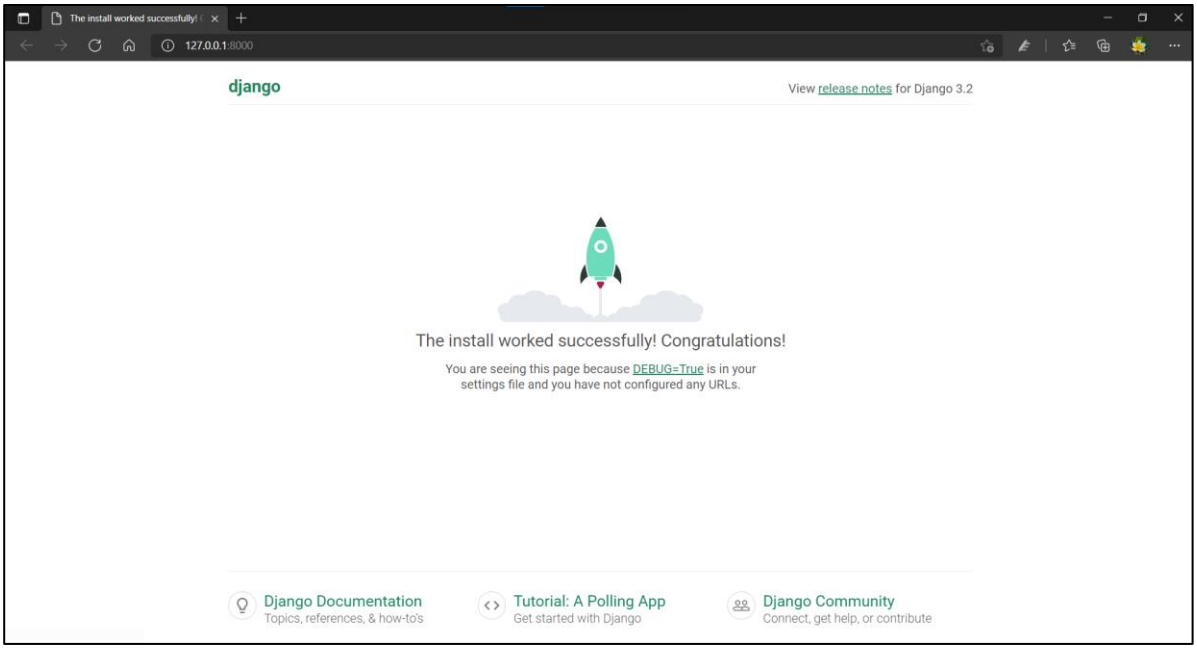
```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS E:\Internship-AkashTechnolab\Tasks\Task-6> django-admin startproject demoproject
PS E:\Internship-AkashTechnolab\Tasks\Task-6> cd demoproject
PS E:\Internship-AkashTechnolab\Tasks\Task-6\demoproject> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
PS E:\Internship-AkashTechnolab\Tasks\Task-6\demoproject>
```



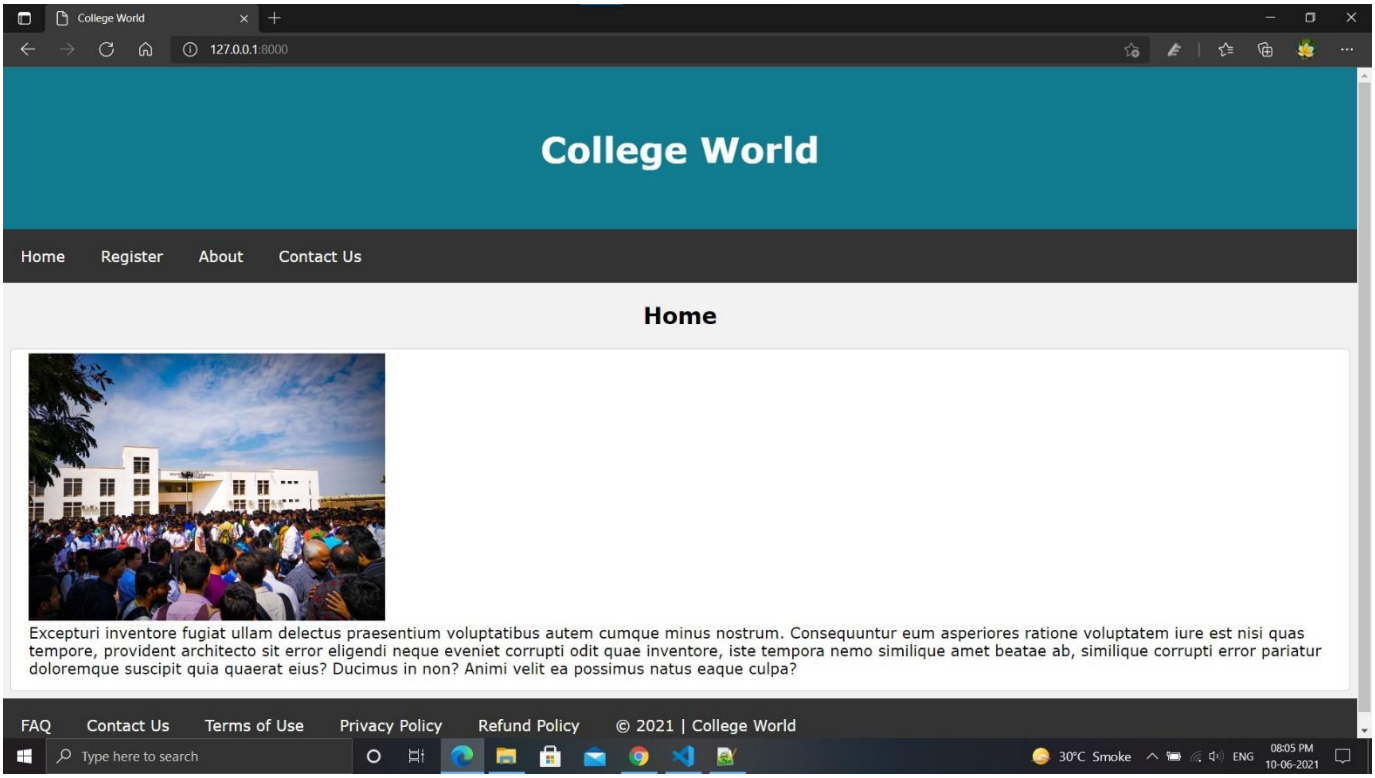
Necessary Command before running:
Python manage.py runserver



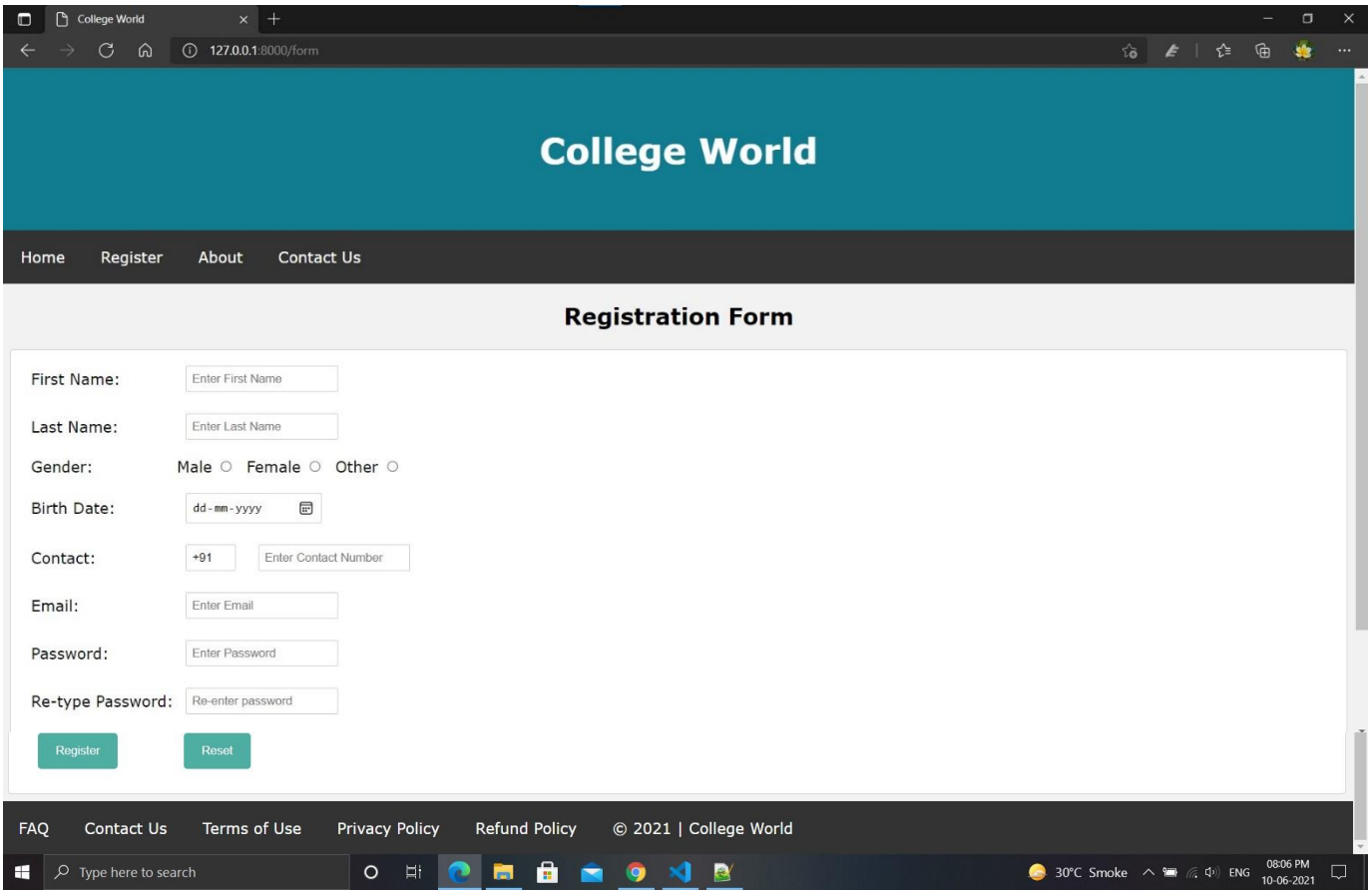
Browser Window:
Welcome Screen


The install worked successfully!
127.0.0.1:8000
django
View release notes for Django 3.2
The install worked successfully! Congratulations!
You are seeing this page because DEBUG=True is in your settings file and you have not configured any URLs.
Django Documentation
Topics, references, & how-to's
Tutorial: A Polling App
Get started with Django
Django Community
Connect, get help, or contribute

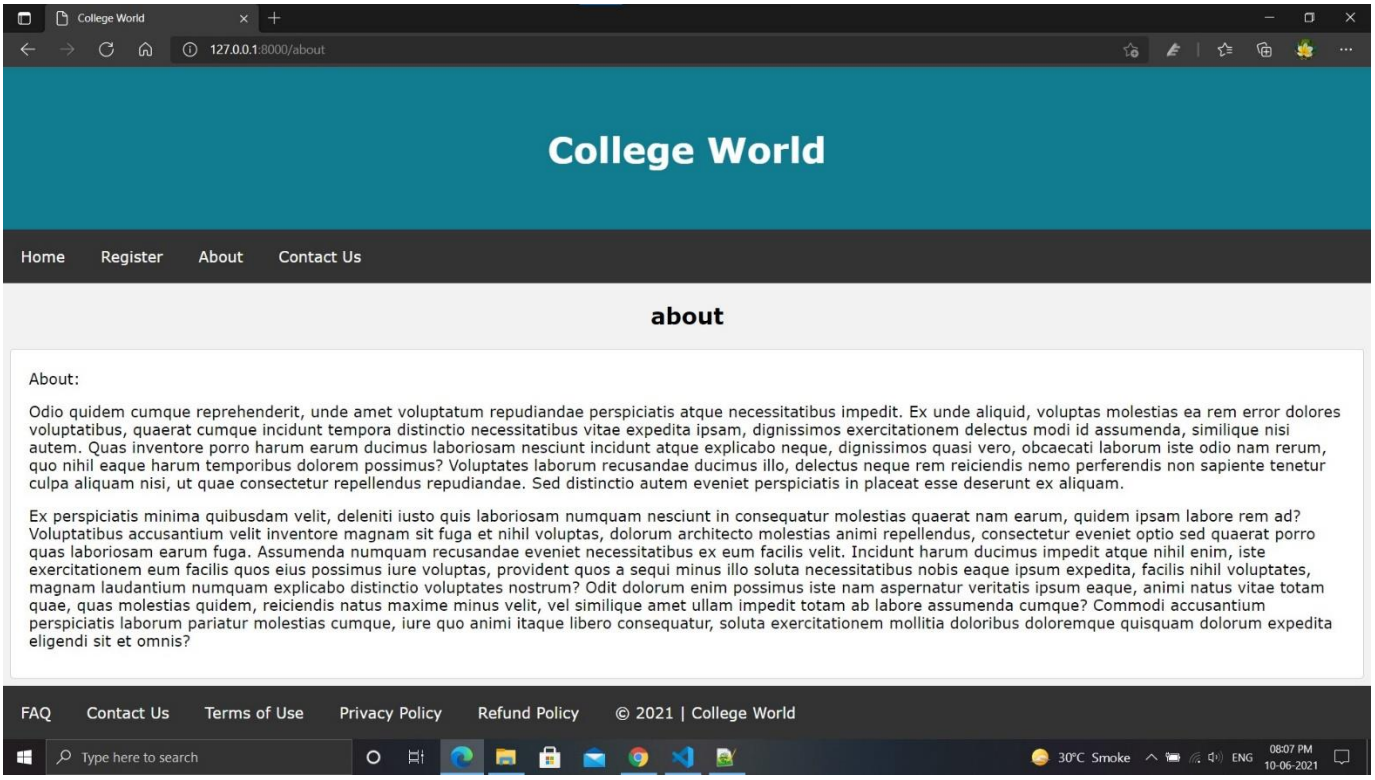
Home page


College World
127.0.0.1:8000
College World
Home
Register
About
Contact Us
Home
Excepturi inventore fugiat ullam delectus praesentium voluptatibus autem cumque minus nostrum. Consequuntur eum asperiores ratione voluptatem iure est nisi quas tempore, provident architecto sit error eligendi neque eveniet corrupti odit quae inventore, iste tempora nemo similique amet beatae ab, similique corrupti error pariatur doloremque suscipit quia quaerat eius? Ducimus in non? Animi velit ea possimus natus eaque culpa?
FAQ
Contact Us
Terms of Use
Privacy Policy
Refund Policy
© 2021 | College World
Type here to search
30°C Smoke
ENG
08:05 PM
10-06-2021

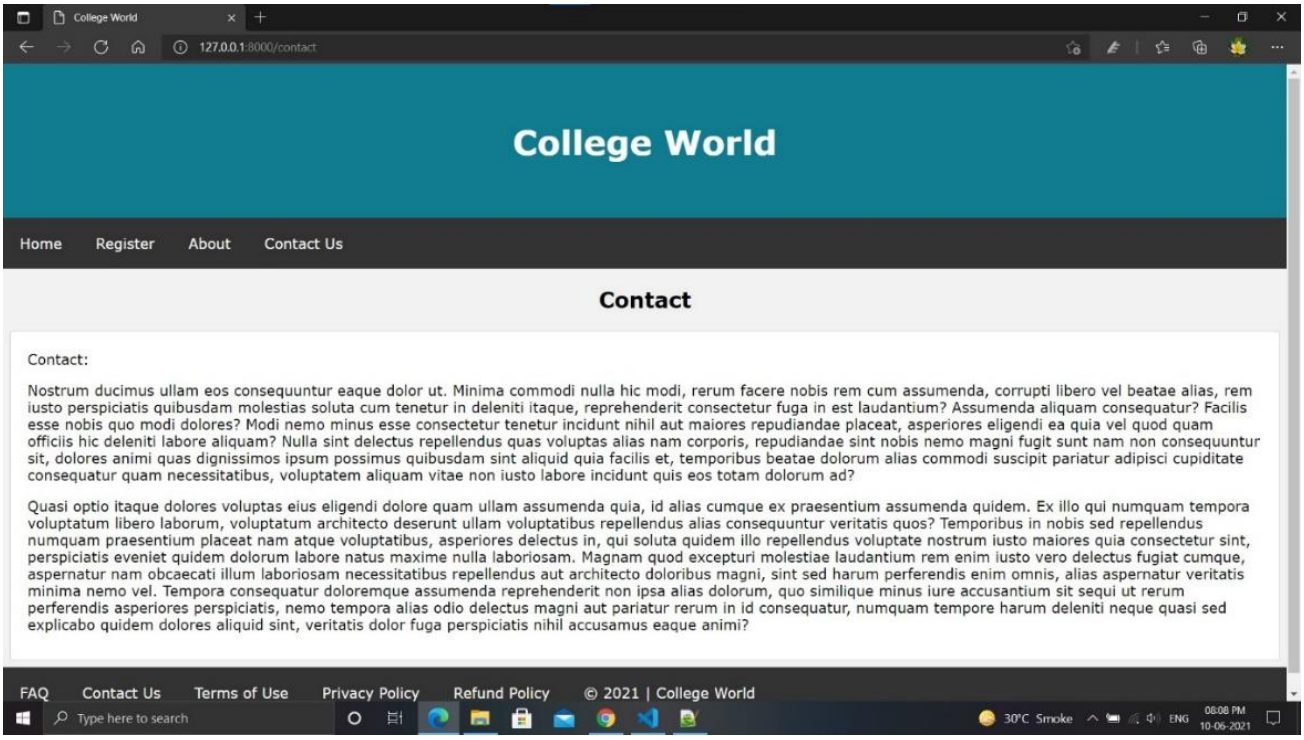
Registration Page



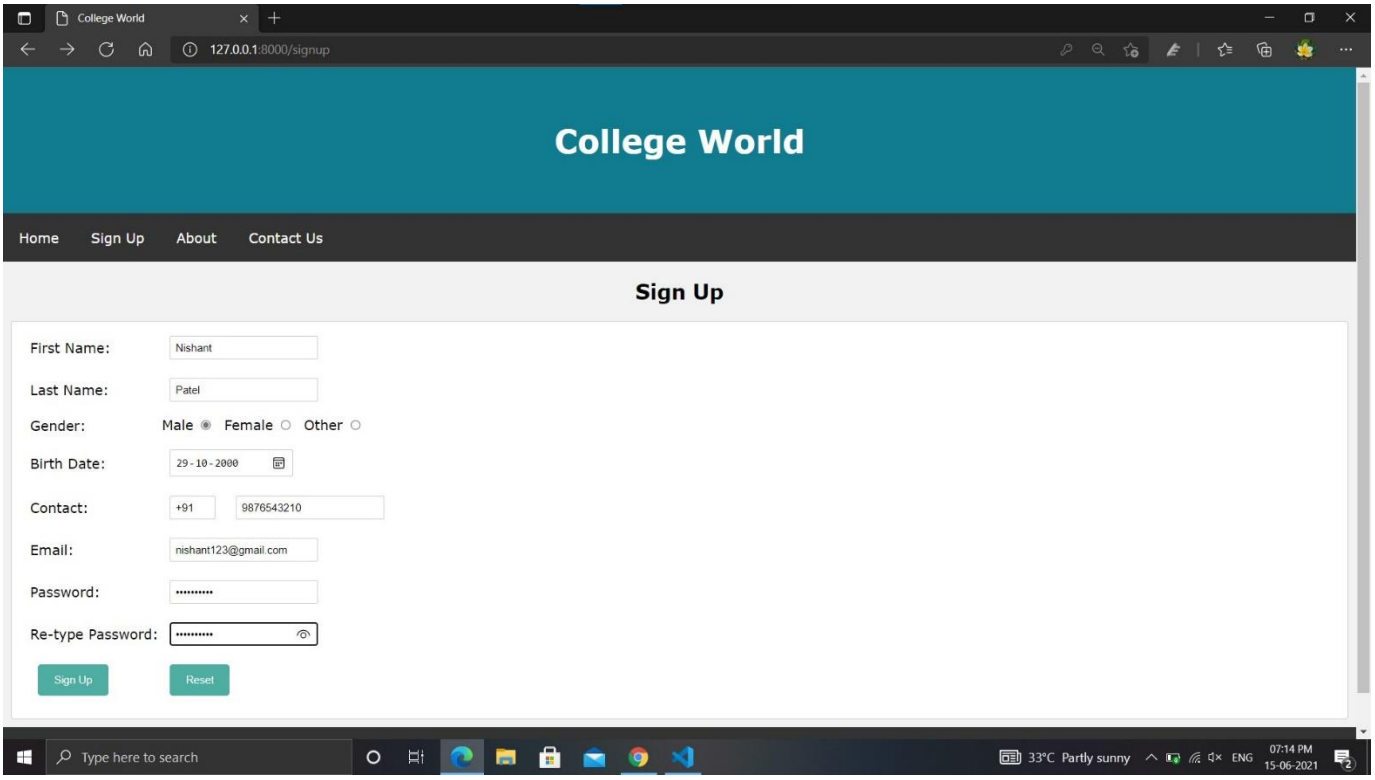
About Page



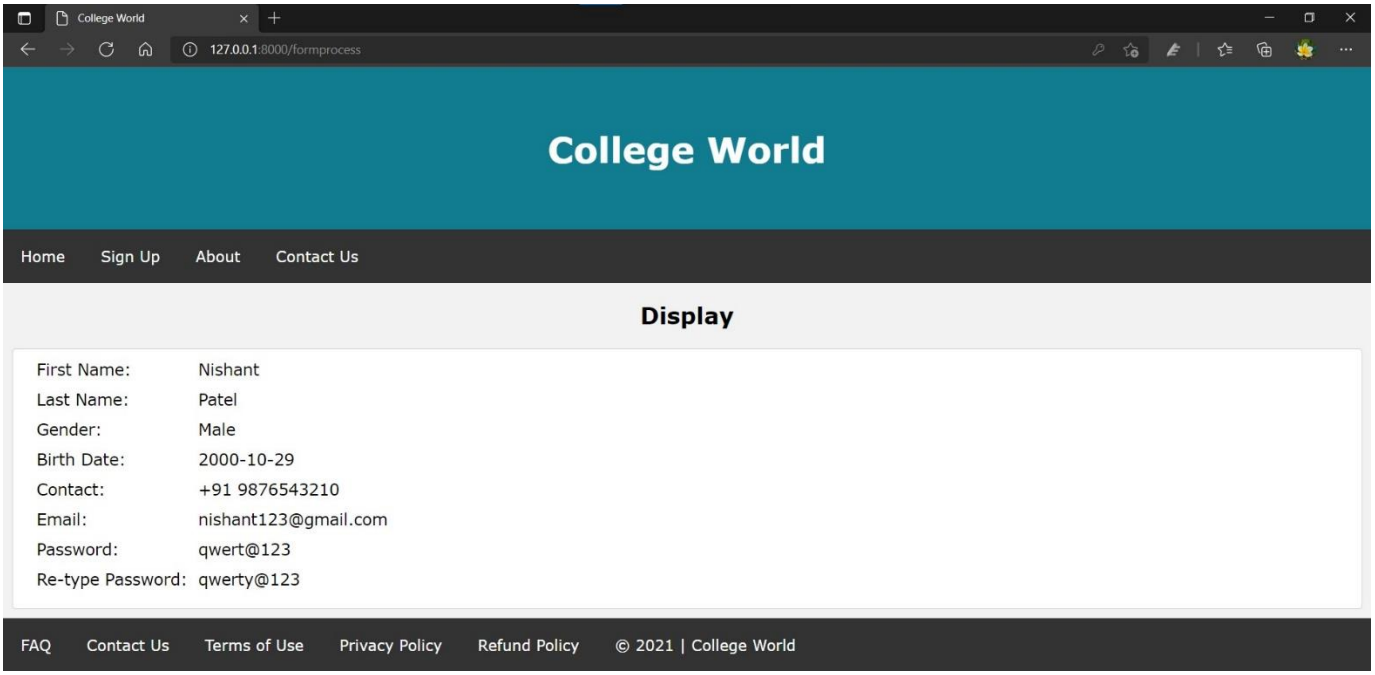
Contact Page



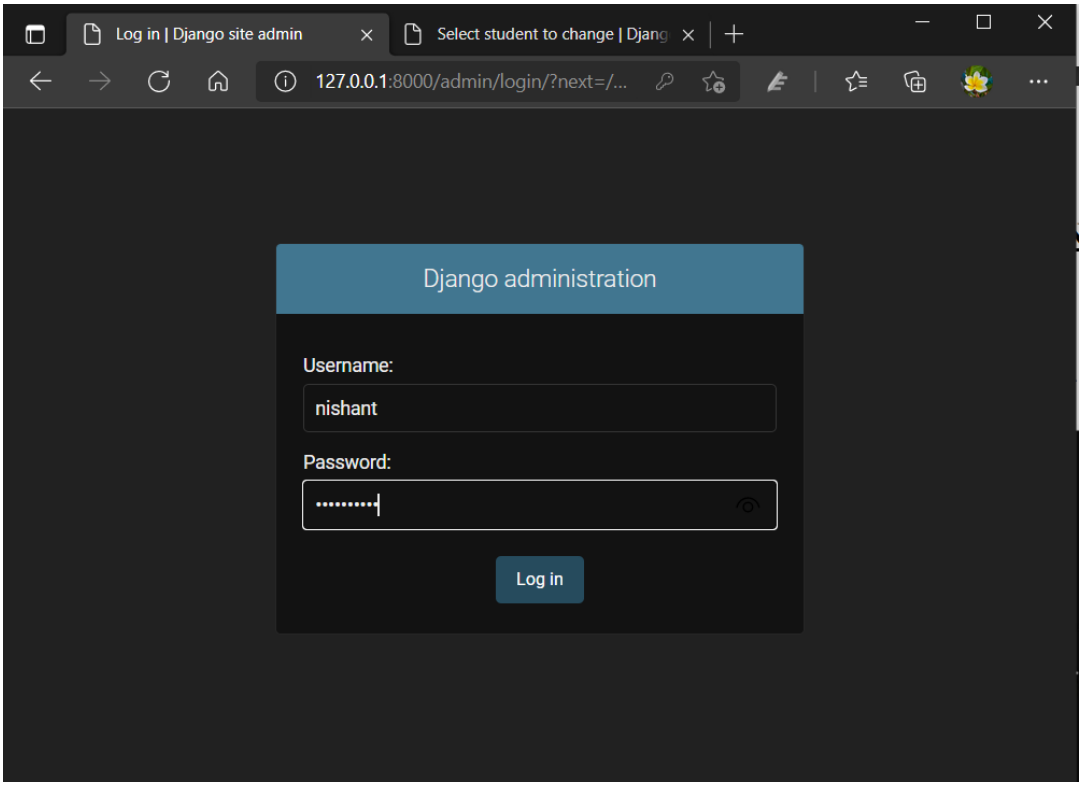
Sign Up Page

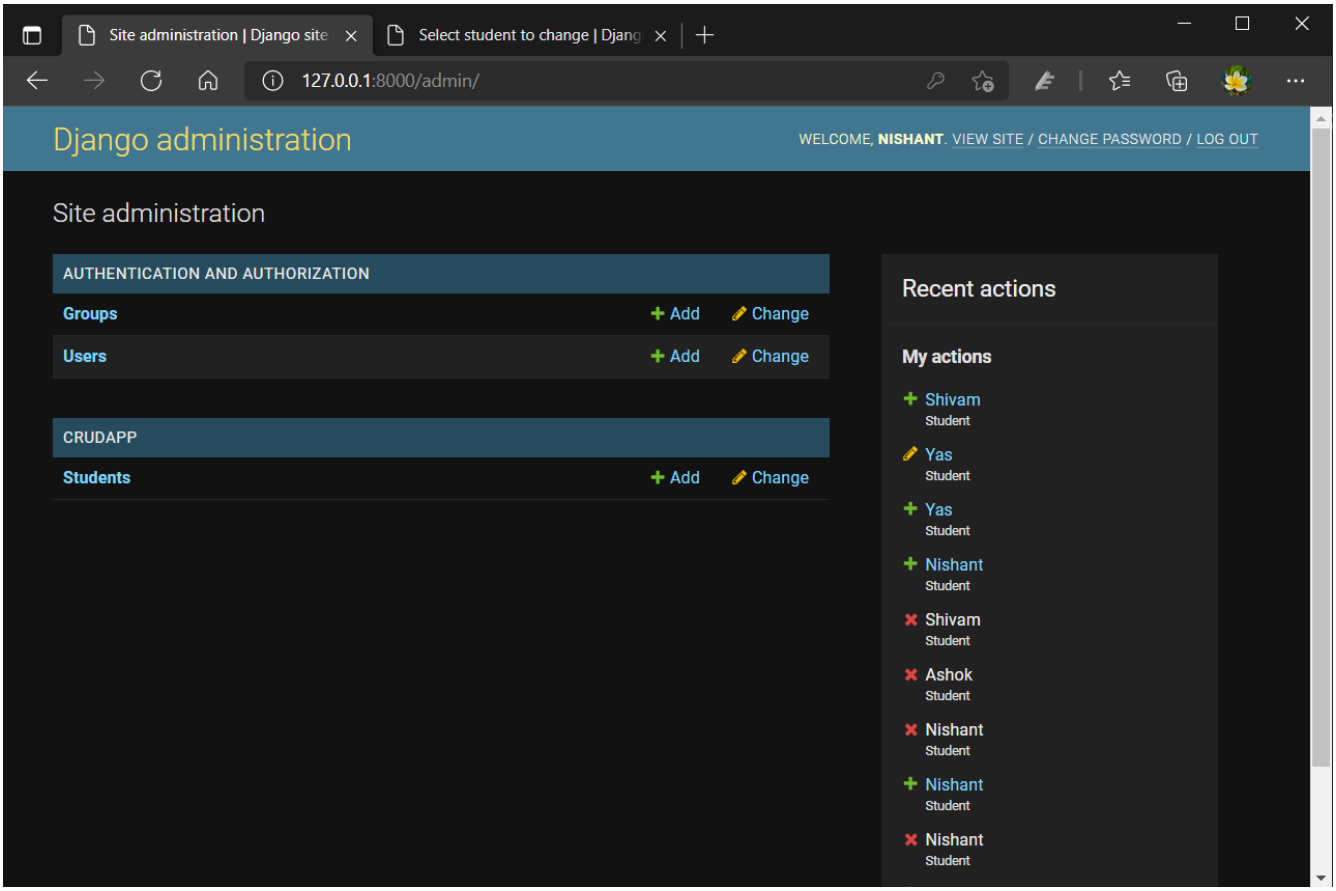


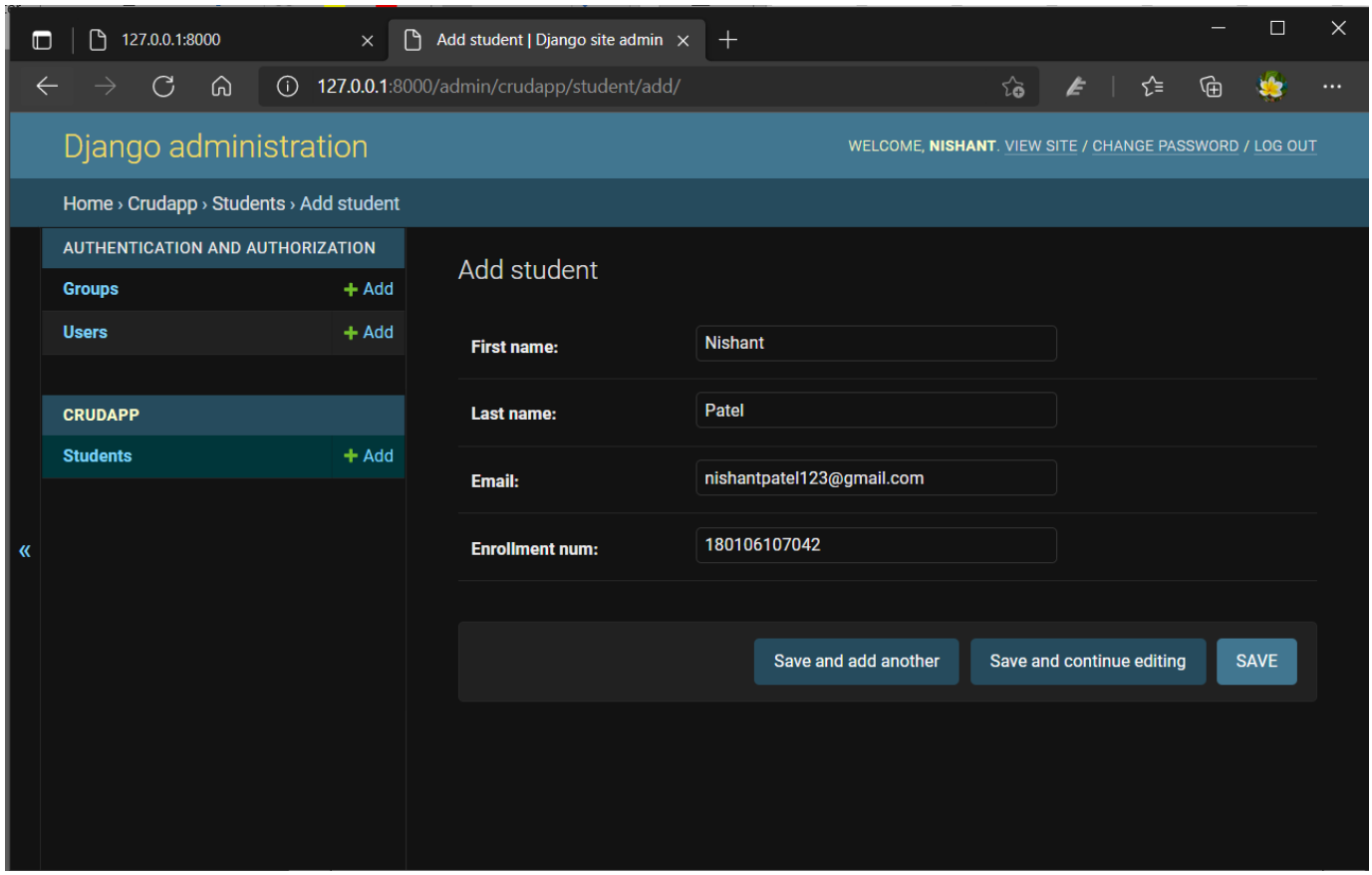
Display Data

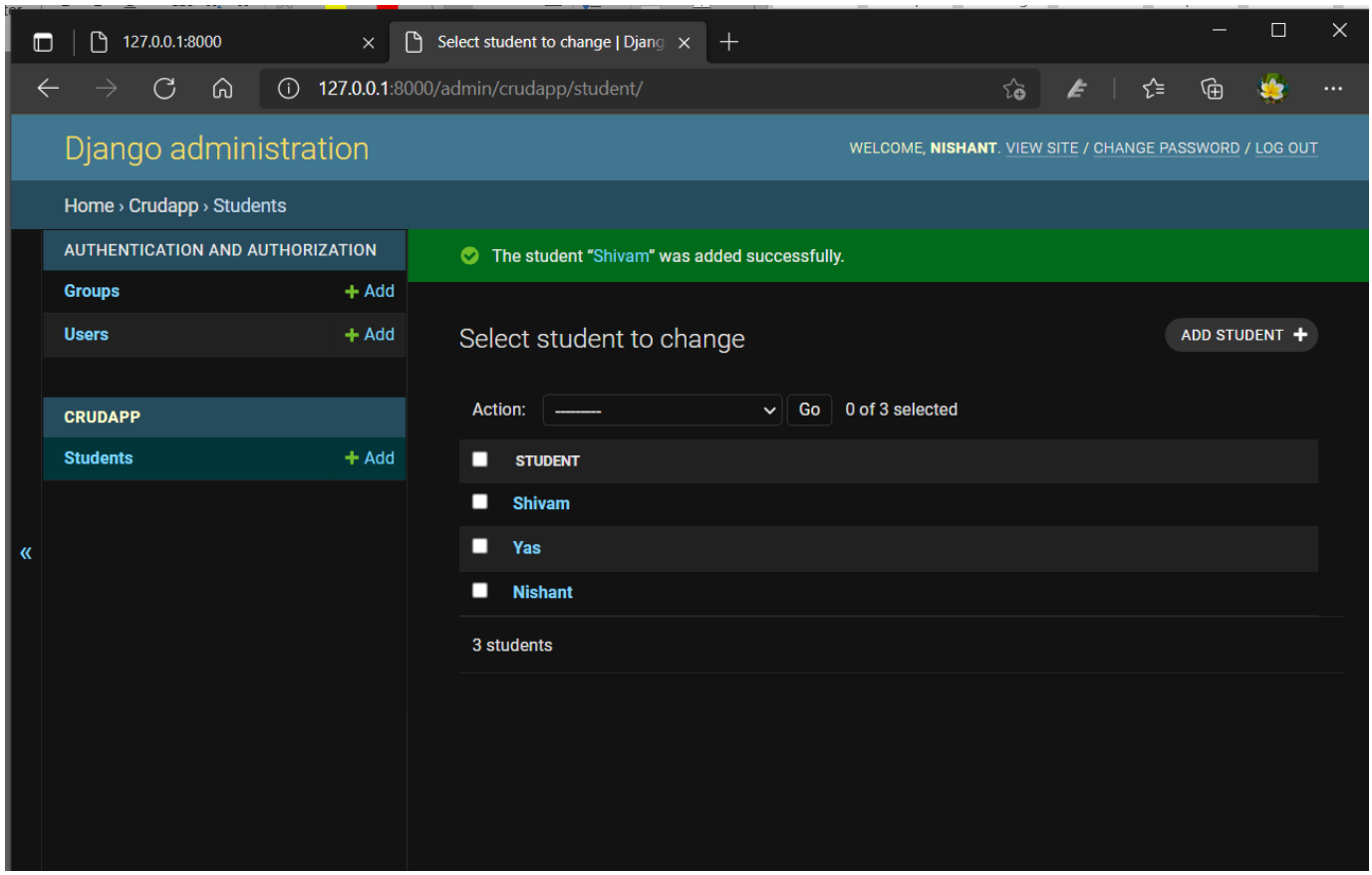


Admin Panel

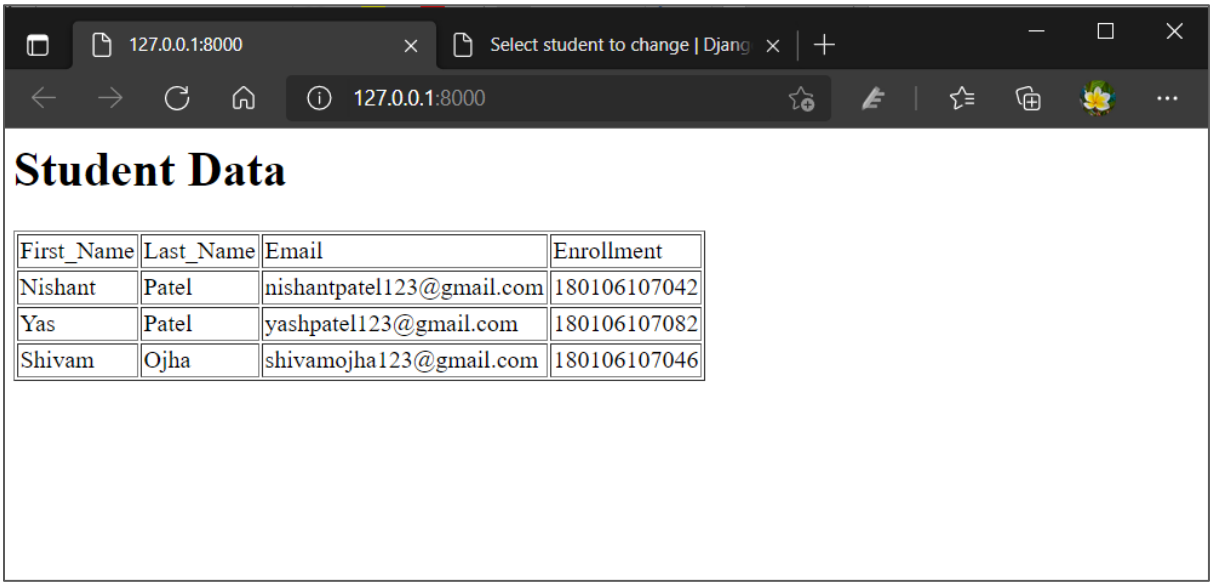

Log in | Django site admin
Select student to change | Djang
127.0.0.1:8000/admin/login/?next=/...
Django administration
Username:
nishant
Password:
Log in


Site administration | Django site
Select student to change | Djang
127.0.0.1:8000/admin/
Django administration
WELCOME, NISHANT. VIEW SITE / CHANGE PASSWORD / LOG OUT
Site administration
AUTHENTICATION AND AUTHORIZATION
Groups
Add
Change
Users
Add
Change
CRUDAPP
Students
Add
Change
Recent actions
My actions
Shivam
Student
Yas
Student
Yas
Student
Nishant
Student
Shivam
Student
Ashok
Student
Nishant
Student
Nishant
Student
Nishant
Student


127.0.0.1:8000
Add student | Django site admin
127.0.0.1:8000/admin/crudapp/student/add/
Django administration
WELCOME, NISHANT. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Crudapp › Students › Add student
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
CRUDAPP
Students + Add
Add student
First name:
Nishant
Last name:
Patel
Email:
nishantpatel123@gmail.com
Enrollment num:
180106107042
Save and add another
Save and continue editing
SAVE



Displaying Student Data



First_Name	Last_Name	Email	Enrollment
Nishant	Patel	nishantpatel123@gmail.com	180106107042
Yas	Patel	yashpatel123@gmail.com	180106107082
Shivam	Ojha	shivamojha123@gmail.com	180106107046

CHAPTER-6

CONCLUSION

KEY LEARNING POINTS

- Learnt about Django Framework – How to use it to develop Rapid & Fast Web Development. The way Django supports & makes easy for us to do CRUD operation in Admin Panel & how they can encourage rapid development.

- Database – Learnt about the CRUD operation in database – how the backend works once user submits the data & how it get's stored & retrieved on being asked for, Data Fetching & update operations. Accessing the data from Database – along with database integration i.e. how to connect database with front-end part.

- Learnt about Django Working Pattern of MVT – Model, Views & Template- How we create configuration files of project i.e. Starting with Creating views (UI) part in views.py & then registering it in urls.py for URL Mapping/Routing & configuring it in settings.py of the Project directory.

- The skill of integrating template into DJANGO- How the HTML & CSS are worked upon to integrate with Django: Exteding base template with all other pages & making changes rapidly to apply all over thereby reducing Boiler-Plate Code.

- Skill of Solving Errors by analyzing console & error description. Handling/Management of a project from ground base to creating Productive Website with the help of Django – Hosting it Online to generate View Content.

Setting up Admin Panel, creating Users & performing create, read, update & delete operation on the same, Database Configuration within admin panel – Setting up Models in model.view & creating database in admin panel. Adding users & making the changes.

CHAPTER-7

REFERENCE

- https://www.djangoproject.com/
- https://docs.djangoproject.com/en/3.2/
- https://akashtechnolabs.com/
- https://github.com/django/django
- https://www.w3schools.com/python/
- https://github.com/nishantmovaliya