WEB DEVELOPMENT LABORATORY

(20ES3151)

Week - 0

1.What is the full form of HTML?

Ans. HyperText MarkUp Language

2.What is the present version of HTML?

Ans. HTML5 with html file extension

3. Differentiate between WEB PAGE and WEB SITE

WEB PAGE	WEB SITE
A direct URL link or a website can be used to access it	A domain address is used to access it
Webpage consists of content regarding a single entity type	Website constitutes content regarding several entitieS
A combination of web pages is created using HTML and CSS	Information is in HTML language
An individual hypertext document linked under a website	A collection of multiple pages hosted on the server
Less time to develop	Takes more time to develop
Comparatively, less complex	More complex to develop

4. What are h1, h2, h3, h4, h5 and h6 tags ?

Ans. HTML defines six levels of headings. The heading elements are h1, h2, h3, h4, h5 and h6 with h1 being the highest level and h6 being the least. Used to create text headers for a document

5. What is the difference between LAN, WAN, MAN

Ans. LAN – A LAN connects computers within a small and specific area geographically

WAN –A WAN covers large area geographically such as continent, state or a country

MAN – A MAN is confined to a specific town, city or region

6. What is the difference between physical and logical address

Ans. 1- Logical address is generated by CPU during a program execution

2 – The physical address refers to a location in the memory unit

7. How to write a tag in HTML?

Ans. A HTML tag is special word or letter surrounded by angle brackets < and > paragraph element has tag followed by paragraph text followed by a closing tag

8. Brief History about HTML

Ans. HTML was created by Sir Tim Berners-Lee in late 1991 but was not released officially published in 1995 as HTML 2.0.

HTML 4.01 was published in late 1999 and was a major version of HTML.

HTML 1.0 was released in 1993 with the intension of sharing information that can be readable and accessible via web browsers

Week - 1

1. What are the different types of tags

Ans. There are two types of tags in HTML , they are:

a. Paired tags: In each paired tag set there will be an opening tag along with the closing tag, the closing tag contains a slash (/)

Ex: <h1>heading tag</h1>

paragraph tag

<html> html tag</html>

<head> head tag</head>

<body> body tag</body>

b. Unpaired tags: The unpaired tags does not contain the closing tag, these are also called as **singular tags**

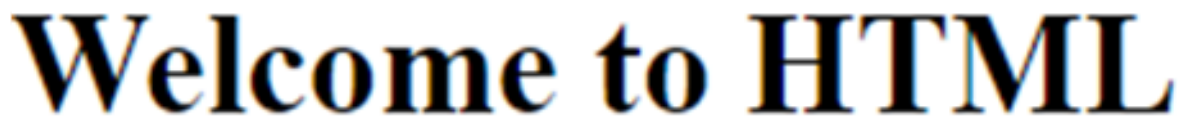
Ex:
 , <hr> , <meta> ,<input> , <img src>

2. AIM: To write a welcome program

PROGRAM:

```
<!DOCTYPE html>

<html>

<title>INTRODUCTION</title>

<head>

<h1>Welcome to HTML</h1>

</head>
```

OUTPUT:



3. AIM: Performing a task using different heading tags

PROGRAM:

```
<!DOCTYPE html>
<body>
<h1>Welcome to html</h1>
<h2>Welcome to html</h2>
<h3>Welcome to html</h3>
<h4>Welcome to html</h4>
<h5>Welcome to html</h5>
<h6>Welcome to html</h6>
</body>
</html>
```

OUTPUT:

Welcome to html

Welcome to html

Welcome to html

Welcome to html

Welcome to html

Welcome to html

WEEK – 2

1. AIM: Insertion of photo in the resume

PROGRAM:

```
<!DOCTYPE html>

<head>

<title>My Profile</title>

<h1><center>RESUME</center></h1>

</head>

<body>

<img src="passport.jpg" align="right" width="200" height="200"/>

<h3>ABOUT ME:</h3>

<p>My name is SPANDANA VATTIPALLI, I am 19 years old, Currently Studying second year in Information Technology at Velagapudi Ramakrishna Siddhartha Engineering College,

My native place is Vijayawada. My Hobbies are reading books, playing games.</p>

</body>

</html>
```

OUTPUT:

RESUME

ABOUT ME:

My name is SPANDANA VATTIPALLI, I am 19 years old, Currently Studying second year in Information Technology at Velagapudi Ramakrishna Siddhartha Engineering College, My native place is Vijayawada. My Hobbies are reading books, playing games.

WEEK - 3

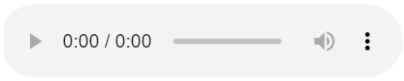
1. AIM: VIDEO & AUDIO UPLOADING

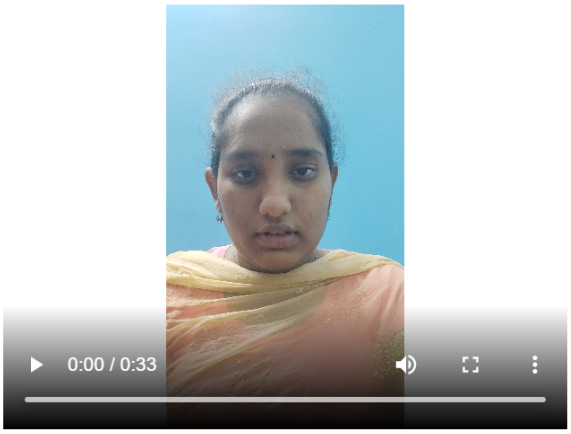
PROGRAM:

```
<!DOCTYPE html>
<head>
<title>My Profile</title>

<center><video width="420" height="315"controls>

<source src="video resume.mp4" type="video/mp4">

</video></center>

<audio controls>

<source src=”bird chirp.mp3” type=”audio/mpeg”>

Your browser does not support the audio tag.

</audio>
```

```
</body>

</html>
```

OUTPUT:





Week - 4

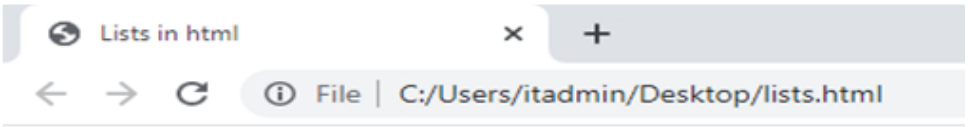
1.AIM: To create different types of LISTS

PROGRAM:

```
<!DOCTYPE html>
<head>
<title>LISTSl</title>
</head>
<body>
<h2>UNORDERED LIST</h2>
<h3>Fruits</h3>
<ul>
<li>Apple</li>
<li>Orange</li>
<li>Banana</li>
</ul>

<h2>ORDERED LIST</h2>
<h3>Colors</h3>
<ol>
<li>orange</li>
<li>green</li>
<li>blue</li>
```

```
</ol>


<h2>DESCRIPTION LIST</h2>

<h3>Menu</h3>

<dl>

<dt>Coffee</dt>

<dd>-espresso</dd>

<dd>-black coffee</dd>

<dd>-cold coffee</dd>

<dt>Milshake</dt>

<dd>-vanilla</dd>

<dd>-chocolate</dd>

<dd>-strawberry</dd>

</body>

</html>
```

OUTPUT:



UNORDERED LIST

Fruits

- Apple
- Orange
- Banana

LANGUAGES

- English
- Telugu
- Tamil

ORDERED LIST

Colors

1. orange
2. green
3. blue

DESCRIPTION LIST

Menu

Coffee
-espresso
-black coffee
-cold coffee
Milshake
-vanilla
-chocolate
-strawberry

WEEK - 5

1. AIM: To create the academic table

PROGRAM:

```
<!DOCTYPE html>
<head>
<style>
table, th, td {
  border:1px solid black;
}
</style>
<center><h2>TABLE</h2></center>
</head>
<body>
<table>
<tr>
<th>S.No</th>
<th>INSTITUTE/COLLEGE</th>
<th>BOARD</th>
<th>YEAR OF PASSING</th>
<th>CGPA</th>
</tr>
<tr>
```

<td>1</td>

<td>VRSEC</td>

<td>jntuk</td>

<td>2024</td>

<td>8</td>

</tr>

<tr>

<td>2</td>

<td>Sri Chaitanya Junior College</td>

<td>Board of Intermediate</td>

<td>2020</td>

<td>9</td>

</tr>

<tr>

<td>3</td>

<td>Sri Chaitanya High School</td>

<td>SSC</td>

<td>2018</td>

<td>10</td>

</tr>

</table>

</body>

</html>

OUTPUT:


table.html
File | C:/Users/itadmin/Desktop/table.html
TABLE
S.No INSTITUTE/COLLEGE BOARD YEAR OF PASSING CGPA
1 VRSEC jntuk 2024 8
2 Sri Chaitanya Junior College Board of Intermediate 2020 9
3 Sri Chaitanya High School SSC 2018 10

WEEK - 6

1.AIM: To create a RESUME using frames

PROGRAM:

```
<!DOCTYPE html>

<html>

<frameset cols="25%,50%,25%">

<frame src="frame_1.html">

<frame src="frame_2.html">

<frame src="frame_3.html">

<frame src="frame_4.html">

</frameset>

</html>
```

OUTPUT:

v.spandana223@gmail.com
9876543210
github linkedin
hackerank

Skill Set:

programming skills learned:

- HTML
- Python
- C
- C++
- CSS

courses completed:

- CODE RIDERS
- APSSDC

SPANDANA VATTIPALLI

About Me:

My name is SPANDANA VATTIPALLI, I am 19 years old, Currently Studying second year in Information Technology at Velagapudi Ramakrishna Siddhartha Engineering College, My native place is Vijayawada. My Hobbies are reading books, playing games.

Career Objective:

To use my skills in best possible ways to achieve company goals and to solve problems in an effective/creative mannerin a challenging position

S.No	INSTITUTE/COLLEGE	BOARD	YEAR OF PASSING	CGPA
1	VRSEC	jntuk	2024	8
2	Sri Chaitanya Junior College	Board of Intermediate	2020	9
3	Sri Chaitanya High School	SSC	2018	10

Strengths:

- Ability to cope up with the situation
- Planning and Accomplishing a task
- Confident and Determined

Hobies:

- Painting
- Writing
- Playing Badminton

Personal details:

DATE OF BIRTH : 2nd February 2003

FATHER'S NAME : V.BHASKARA RAO

GENDER : Female

LANGUAGES KNOWN : English,Telugu

Week-7

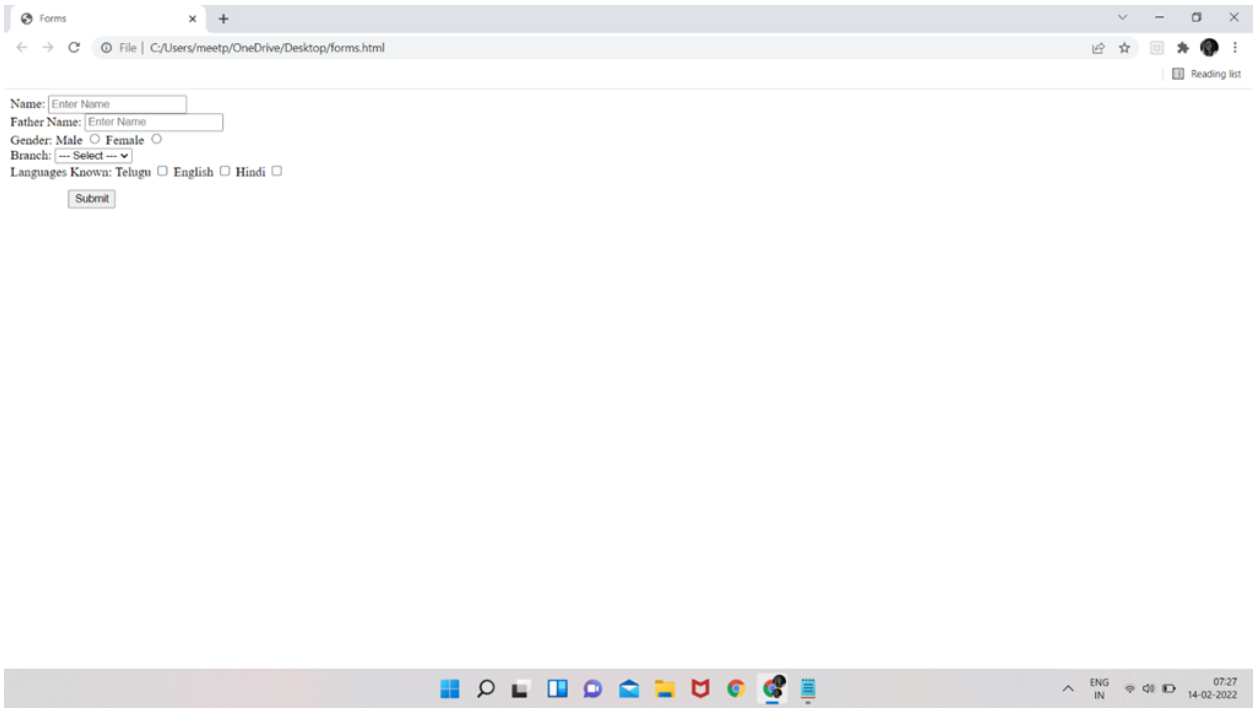
AIM : Create a form using html

PROGRAM :

```
<html>

<head>

<title>Forms</title>

</head>

<body>

<form name="it" action="next.php" method="post">

Name: <input type="textbox" placeholder="Enter Name" name="Name" required>

<br>

Father Name: <input type="textbox" placeholder="Enter Name" name="Father Name" required>

<br>

Gender: Male <input type="radio" name="gender" value="Male"> Female <input type="radio" name="gender" value="Female">

<br>

Branch:

<select name="branch">

<option>--- Select ---</option>
```

```
<option> CSE </option>

<option> IT </option>

<option> ECE </option>

</select>

<br>

Languages Known: Telugu

<input type="checkbox" value="telugu" name="telugu">

English

<input type="checkbox" value="english" name="english">

Hindi

<input type="checkbox" value="hindi" name="hindi">

 <pre>         <input type="submit" value="Submit"></pre>

 </form>

</body>

</html>
```

OUTPUT:


Forms
File | C:/Users/meetp/OneDrive/Desktop/forms.html
Reading list
Name: Enter Name
Father Name: Enter Name
Gender: Male Female
Branch: --- Select ---
Languages Known: Telugu English Hindi
Submit
ENG
IN
07:27
14-02-2022

WEEK - 8

AIM : Create a WEBSITE using HTML

PROGRAM :

HOME PAGE:

```
<!DOCTYPE html>

<head>

<style>

table,th,td{

   border:1px solid black;

}

</style>

<style>

h1 {

  text-shadow: 2px 2px 5px red;

}

</style>

<title>home page</title>

<h1 style="font-size:50px;"><center>INDIAN PREMIERE LEAGUE</center></h1>

</head>
```

```
<style>

a:link {

  text-decoration: none;

}

a:visited {

  text-decoration: none;

}

a:hover {

  text-decoration: underline;

}

a:active {

  text-decoration: underline;

}

</style>

<body style="background-image: linear-gradient(to right, #ffafbd, #ffc3a0);">

<nav style="color:red;">

<a href="home.html"><span
style="font-size:25px;color:red;">HOME    </a>

<a href="login.html"><span
style="font-size:25px;color:red;">LOGIN    </a>
```

```
<a href="registration.html"><span
style="font-size:25px;color:red;">REGISTRATION    </a>

<a href="editpage.html"><span
style="font-size:25px;color:red;">EDIT    </a>

<a href="more.html"><span style="font-size:25px;color:red;">MORE</a><br><br><br>

</nav>

<center><img src="ipl.jpg" align="center" height="500" width="500"></center>

<div>

<br><br><br>

<table>

<tr>

<th>S.No</th>

<th>TEAMS</th>

<th>TOTAL MATCHES PLAYED</th>

<th>WON</th>

<th>LOST</th>

<th>DRAW</th>

<th>POINTS</th>

</tr>

<tr>
```

<td>1</td>

<td>MI</td>

<td>15</td>

<td>12</td>

<td>2</td>

<td>1</td>

<td>+53</td>

</tr>

<tr>

<td>2</td>

<td>CSK</td>

<td>15</td>

<td>11</td>

<td>2</td>

<td>2</td>

<td>+48</td>

</tr>

<tr>

<td>3</td>

<td>RCB</td>

<td>14</td>

<td>10</td>

<td>3</td>

<td>1</td>

<td>+40</td>

</tr>

<tr>

<td>4</td>

<td>KXIP</td>

<td>14</td>

<td>9</td>

<td>3</td>

<td>2</td>

<td>+37</td>

</tr>

<tr>

<td>5</td>

<td>SRH</td>

<td>14</td>

<td>8</td>

<td>4</td>

<td>2</td>

<td>+32</td>

</tr>

<tr>

<td>6</td>

<td>DC</td>

<td>15</td>

<td>8</td>

<td>4</td>

<td>3</td>

<td>+29</td>

</tr>

<tr>

<td>7</td>

<td>KKR</td>

<td>14</td>

```
<td>8</td>

<td>5</td>

<td>1</td>

<td>+23</td>

</tr>

<tr>

<td>8</td>

<td>RR</td>

<td>15</td>

<td>8</td>

<td>5</td>

<td>2</td>

<td>+20</td>

</tr>

</table>

</div>

</body>

</html>
```

LOGIN PAGE:

```
<!DOCTYPE html>

<head>

<title>login page</title>

<h1 style="font-size:50px;"><center>INDIAN PREMIERE
LEAGUE</center></h1>

</head>

<style>

a:link {

  text-decoration: none;

}


a:visited {

  text-decoration: none;

}


a:hover {

  text-decoration: underline;

}


a:active {

  text-decoration: underline;

}
```

```
</style>

<body style="background-image: linear-gradient(to right, #ffafbd, #ffc3a0);">

<nav">

<a href="home.html"><span
style="font-size:25px;color:red;">HOME    </a>

<a href="login.html"><span
style="font-size:25px;color:red;">LOGIN    </a>

<a href="registration.html"><span
style="font-size:25px;color:red;">REGISTRATION    </a>

<a href="editpage.html"><span
style="font-size:25px;color:red;">EDIT    </a>

<a href="more.html"><span
style="font-size:25px;color:red;">MORE</a><br><br><br>

</nav>

<div>

<br><br><br>

<h2><center>WELCOME TO LOGIN PAGE</center></h2>

<center>

<table>

<form>

<tr>

<td>Username</td>

<td><input type="text" name="t1"></td>

</tr>
```

```
<tr>
<td>Password</td>
<td><input type="Password" name="t2"></td>
</tr>
<tr>
<td><center><input type="submit"></td>
</tr>
</form>
</table>
</center>
</div>
</body>
</html>
```

REGISTRATION PAGE:

```
<!DOCTYPE html>
<head>
<title>registration page</title>
<style>
table{
margin-top:50px;
```

```
}
</style>
<h1 style="font-size:50px;"><center>INDIAN PREMIERE
LEAGUE</center></h1>
</head>
<style>
a:link {
  text-decoration: none;
}

a:visited {
  text-decoration: none;
}

a:hover {
  text-decoration: underline;
}

a:active {
  text-decoration: underline;
}
</style>
<body style="background-image: linear-gradient(to right, #ffafbd, #ffc3a0);">
```

```
<nav>

<a href="home.html"><span
style="font-size:25px;color:red;">HOME    </a>

<a href="login.html"><span
style="font-size:25px;color:red;">LOGIN    </a>

<a href="registration.html"><span
style="font-size:25px;color:red;">REGISTRATION    </a>

<a href="editpage.html"><span
style="font-size:25px;color:red;">EDIT    </a>

<a href="more.html"><span
style="font-size:25px;color:red;">MORE</a><br><br><br>

</nav>

<h2><center>WELCOME TO REGISTRATION PAGE</center></h2>

<center>

<table>

<form>

<tr>

<td>Username: :</td>

<td><input type="text" name="t1"></td>

</tr>

<tr>

<td>Password:</td>

<td><input type="Password" name="t2"></td>

</tr>
```

```
<tr>
<td>Surname : </td>
<td><input type="surname" name="t3"></td>
</tr>
<tr>
<td>DateofBirth:  </td>
<td><input type="" name="t4"></td>
</tr>
<tr>
<td>Email: </td>
<td><input type="email" name="t5"></td>
</tr>
<tr>
<td><center><input type="submit"></td>
</tr>
</form>
</table>
</center>
</body>
</html>
```

EDIT PAGE:

```
<!DOCTYPE html>
<head>
<title>edit page</title>
<h1 style="font-size: 50px";><center>INDIAN PREMIERE
LEAGUE</center></h1>
</head>
<style>
a:link {
  text-decoration: none;
}

a:visited {
  text-decoration: none;
}

a:hover {
  text-decoration: underline;
```

```
}


a:active {
  text-decoration: underline;
}
</style>
<body style="background-image: linear-gradient(to right, #ffafbd, #ffc3a0);">
<nav>
<a href="home.html"><span
style="font-size:25px;color:red;">HOME    </a>
<a href="login.html"><span
style="font-size:25px;color:red;">LOGIN    </a>
<a href="registration.html"><span
style="font-size:25px;color:red;">REGISTRATION    </a>
<a href="editpage.html"><span
style="font-size:25px;color:red;">EDIT    </a>
<a href="more.html"><span
style="font-size:25px;color:red;">MORE</a><br><br><br>
</nav>
<h2><center>WELCOME TO EDIT PAGE</center></h2>
<form>
<center><input type="text" name="t1">
<input type="submit">
</form>
```

```
</body>
</html>
```

MORE PAGE:

```
<!DOCTYPE html>
<head>
<title>more page</title>
<center><h2>WELCOME TO MORE INFORMATION</h2></center>
<br><br>
<style>
table,th,td{
   border:1px solid black;
}
</style>
</head>
<style>
a:link {
  text-decoration: none;
}

a:visited {
```

```
  text-decoration: none;
}

a:hover {
  text-decoration: underline;
}

a:active {
  text-decoration: underline;
}
</style>
<style>
* {
  box-sizing: border-box;
}

body {
  margin: 0;
  font-family: Arial;
}

.header {
```

```
  text-align: center;
  padding: 32px;
}

.row {
  display: -ms-flexbox; /* IE10 */
  display: flex;
  -ms-flex-wrap: wrap; /* IE10 */
  flex-wrap: wrap;
  padding: 0 4px;
}

/* Create four equal columns that sits next to each other */
.column {
  -ms-flex: 25%; /* IE10 */
  flex: 25%;
  max-width: 25%;
  padding: 0 4px;
}

.column img {
  margin-top: 8px;
```

```
  vertical-align: middle;
  width: 100%;
}

/* Responsive layout - makes a two column-layout instead of four columns */
@media screen and (max-width: 800px) {
  .column {
    -ms-flex: 50%;
    flex: 50%;
    max-width: 50%;
  }
}

/* Responsive layout - makes the two columns stack on top of each other instead of next to each other */
@media screen and (max-width: 600px) {
  .column {
    -ms-flex: 100%;
    flex: 100%;
    max-width: 100%;
  }
}
</style>
```

```
<body style="background-image: linear-gradient(to right, #ffafbd, #ffc3a0);">

<nav>

<a href="home.html"><span
style="font-size:25px;color:red;">HOME    </a>

<a href="login.html"><span
style="font-size:25px;color:red;">LOGIN    </a>

<a href="registration.html"><span
style="font-size:25px;color:red;">REGISTRATION    </a>

<a href="editpage.html"><span
style="font-size:25px;color:red;">EDIT    </a>

<a href="more.html"><span
style="font-size:25px;color:red;">MORE</a><br><br><br>

</nav>

<div class="row">

  <div class="column">

    <img src="first.jpg">

    <img src="second.jpg">

    <img src="third.jpg">

    <img src="fourth.jpg">

  </div>

  <div class="column">

    <img src="fifth.jpg">

    <img src="sixth.jpg">

    <img src="seventh.jpg">
```

```
    <img src="eigth.jpg">
  </div>
  <div class="column">
    <img src="ninth.jpg">
    <img src="tenth.jpg">
    <img src="eleventh.jpg">
    <img src="twelth.jpg">
  </div>
  <div class="column">
    <img src="thirteenth.jpg">
    <img src="fourteenth.jpg">
    <img src="fifteenth.jpg">
    <img src="sixteenth.jpg">
  </div>
</div>
<br><br><br>
<table>
<tr>
<th>YEARS</th>
<th>WINNERS</th>
</tr>
<tr>
```

<td>2015</td>

<td>MUMBAI INDIANS</td>

</tr>

<tr>

<td>2016</td>

<td>SUNRISERS HYDERABAD</td>

</tr>

<tr>

<td>2017</td>

<td>MUMBAI INDIANS</td>

</tr>

<tr>

<td>2018</td>

<td>CHENNAI SUPER KINGS</td>

</tr>

<tr>

<td>2019</td>

<td>MUMBAI INDIANS</td>

</tr>

<tr>

<td>2020</td>

<td>MUMBAI INDIANS</td>

```
</tr>
<tr>
<td>2021</td>
<td>CHENNAI SUPER KINGS</td>
</tr>
</table>
</body>
</html>
```

OUTPUT:

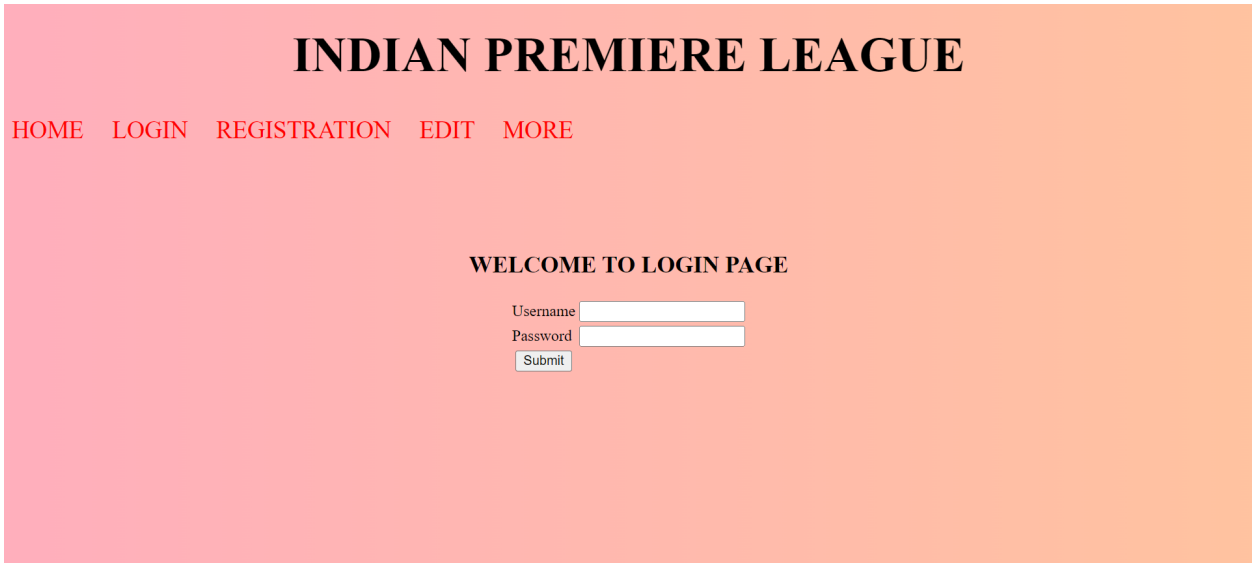
HOME -

INDIAN PREMIERE LEAGUE

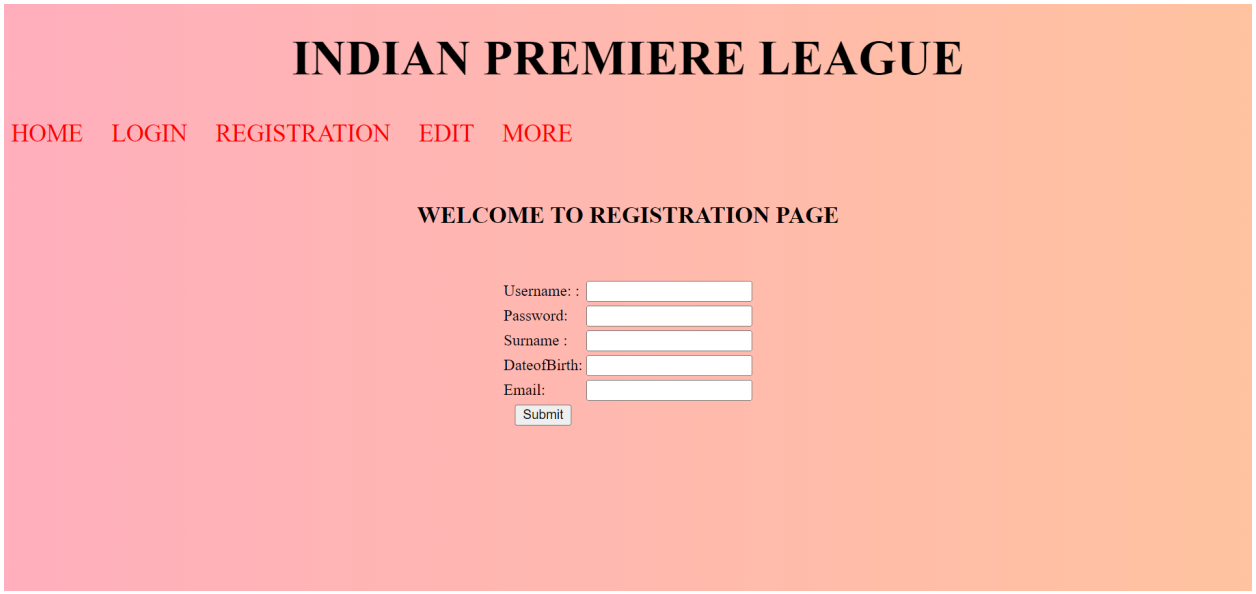
HOME LOGIN REGISTRATION EDIT MORE

S.No	TEAMS	TOTAL MATCHES PLAYED	WON	LOST	DRAW	POINTS
1	MI	15	12	2	1	+53
2	CSK	15	11	2	2	+48
3	RCB	14	10	3	1	+40
4	KXIP	14	9	3	2	+37
5	SRH	14	8	4	2	+32
6	DC	15	8	4	3	+29
7	KKR	14	8	5	1	+23
8	RR	15	8	5	2	+20

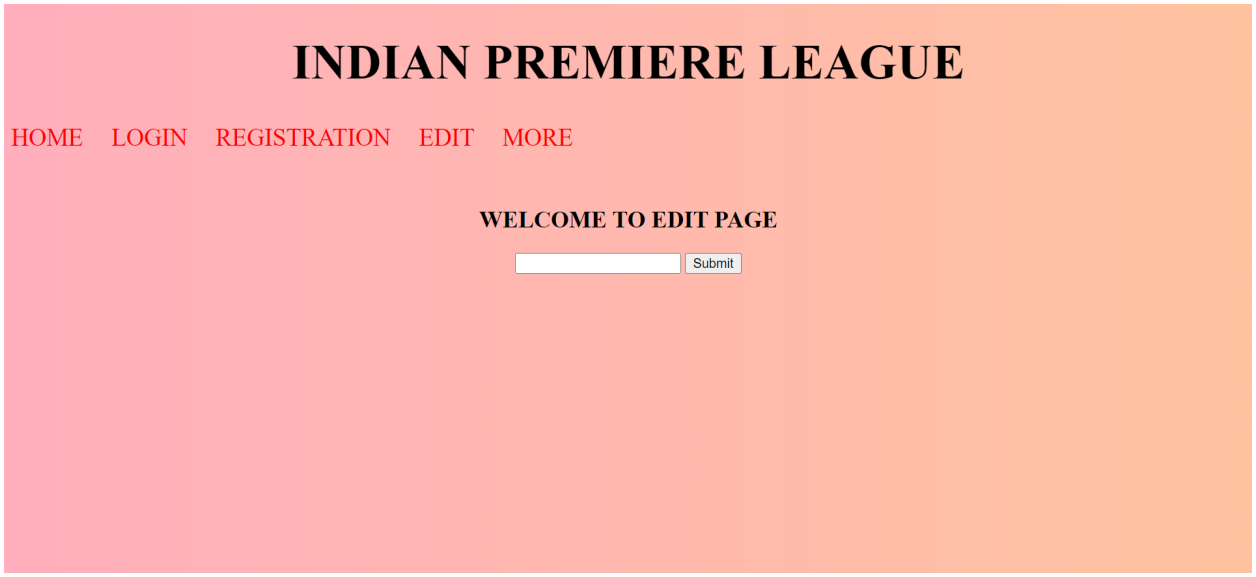
LOGIN PAGE -



REGISTRATION PAGE -



EDIT PAGE -


INDIAN PREMIERE LEAGUE
HOME LOGIN REGISTRATION EDIT MORE
WELCOME TO EDIT PAGE
Submit

MORE INFORMATION PAGE -

WELCOME TO MORE INFORMATION

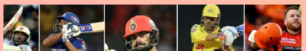
HOME LOGIN REGISTRATION EDIT MORE
















YEARS	WINNERS
2015	MUMBAI INDIANS
2016	SUNRISERS HYDERABAD
2017	MUMBAI INDIANS
2018	CHENNAI SUPER KINGS
2019	MUMBAI INDIANS
2020	MUMBAI INDIANS
2021	CHENNAI SUPER KINGS

Week - 9

1. What is Django, what is the framework that is used to implement Django.

DJANGO

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

Ridiculously fast.

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

Reassuringly secure.

Django takes security seriously and helps developers avoid many common security mistakes.

Exceedingly scalable.Some of the busiest sites on the web leverage Django’s ability to quickly and flexibly scale.

FRAMEWORK

Rich ecosystem

Read Django like a system, developers say. What they mean is that there are many third-party applications that come with Django. These applications can be integrated depending on project requirements. To imagine this better, think of Legos. There are many different Lego blocks. In app development, an authorization “block” or email sending “block” is present in almost every project. Django consists of many applications — such as for authorization and sending emails — that can easily be plugged into a system.

Maturity. Django has been around for 11 years and has gone through stages of significant improvement. A lot of things have been brought to perfection and many new things have been added. Most importantly, when you’re trying to figure out how something should work in Django, you can usually find the answer. Thousands of people must have already solved any issue you’re dealing with, and you can find a solution provided by the passionate Django community.

Admin panel by default

Admin panels are designed to help you manage your application. A Django admin panel is generated automatically from Python code, whereas creating an admin panel manually would take a lot of time and be absolutely pointless.

There’s a lot of room for customization in the Django admin panel thanks to third-party applications. Additionally, Django allows you to modify the interface with third-party wrappers and add dashboards unique to your needs.

Good for SEO

Python is famous for having human-readable code, and that’s an advantage if you want your site to rank high in search results. With Django, you can generate

readable website URLs and links using the most relevant keywords and search engine optimization (SEO) best practices.

After all, a domain name is just a “human-readable” string that maps to a “computer friendly” set of numbers, known as an IP address. People fixate on getting the right domain name, but tend to neglect what the URL slug is—Django can fix that.

Pluggable

Django is pluggable by nature and can be extended with plugins. Plugins are software components that allow developers to add a specific feature to an app, leaving a lot of scope for customization. There are hundreds of packages to help you add Google Maps, create complex permissions, or connect to Stripe to process payments. And if you need to scale your project in the future, you can unplug some components and replace them with others that meet your current needs.

Libraries

Every programming language comes with its own set of libraries for solving common tasks. A software library includes prewritten code, classes, procedures, scripts, configuration data, and more. As a rule, a library is added to a program to provide more functionality or to automate a process without manually writing new code. This reduces time to market.

Django allows developers to use libraries when building any project. Some popular libraries include the Django REST framework, which is responsible for building application programming interfaces (APIs); Django CMS, which is designed to manage website content; and Django-allauth, which is an integrated set of Django applications for authentication, registration, account management, and third-party (social) account authentication.

ORM

Django is valued for its object-relational mapper that helps developers interact with databases. An object-relational mapper (ORM) is a library that automatically transfers data stored in databases such as PostgreSQL and MySQL into objects commonly used in application code.

2. What are the prerequisites to implement Django.

There are many prerequisites to learn Django:

The basic syntax of the Python

Python functions

importing external modules

understanding of Python Path concepts

conditional operators and loops

string manipulation operations

Regular Expression (RE)

object-oriented concepts

Learn REST APIs and JSON

Database Management and SQL queries

3. How to install and run Django

Install Django¶

Django can be installed easily using pip.

In the command prompt, execute the following command: pip install django. This will download and install Django.

After the installation has completed, you can verify your Django installation by executing django-admin --version in the command prompt.

Use the Django admin console

Create a superuser. You will be prompted to enter a username, email, and password. python manage. py createsuperuser.

Start a local web server: python manage. py runserver.

Log in to the admin site using the username and password you used when you ran createsuperuser .

4. Write a very basic program using Django.

django_admin startproject hello_world_project

hello_world_project/

manage.py

helloworld_project/

cd hello_world_project

python manage.py migrate

python manage.py runserver__init__.py

settings.py

urls.py

asgi.py

migrating and testing project

5. What is a database.Which database is by default in Django.

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.

Data within the most common types of databases in operation today is typically modeled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data.

By default, the configuration uses SQLite. If you're new to databases, or you're just interested in trying Django, this is the easiest choice. SQLite is included in Python, so you won't need to install anything else to support your database.

6. What is the framework and advantages of Django.

Rich ecosystem

Read Django like a system, developers say. What they mean is that there are many third-party applications that come with Django. These applications can be integrated depending on project requirements. To imagine this better, think of Legos. There are many different Lego blocks. In app development, an authorization “block” or email sending “block” is present in almost every project. Django consists of many applications — such as for authorization and sending emails — that can easily be plugged into a system.

Maturity. Django has been around for 11 years and has gone through stages of significant improvement. A lot of things have been brought to perfection and many new things have been added. Most importantly, when you’re trying to figure out how something should work in Django, you can usually find the answer. Thousands of people must have already solved any issue you’re dealing with, and you can find a solution provided by the passionate Django community.

Admin panel by default

Admin panels are designed to help you manage your application. A Django admin panel is generated automatically from Python code, whereas creating an admin panel manually would take a lot of time and be absolutely pointless.

There’s a lot of room for customization in the Django admin panel thanks to third-party applications. Additionally, Django allows you to modify the interface with third-party wrappers and add dashboards unique to your needs.

Good for SEO

Python is famous for having human-readable code, and that’s an advantage if you want your site to rank high in search results. With Django, you can generate

readable website URLs and links using the most relevant keywords and search engine optimization (SEO) best practices.

After all, a domain name is just a “human-readable” string that maps to a “computer friendly” set of numbers, known as an IP address. People fixate on getting the right domain name, but tend to neglect what the URL slug is—Django can fix that.

Pluggable

Django is pluggable by nature and can be extended with plugins. Plugins are software components that allow developers to add a specific feature to an app, leaving a lot of scope for customization. There are hundreds of packages to help you add Google Maps, create complex permissions, or connect to Stripe to process payments. And if you need to scale your project in the future, you can unplug some components and replace them with others that meet your current needs.

Libraries

Every programming language comes with its own set of libraries for solving common tasks. A software library includes prewritten code, classes, procedures, scripts, configuration data, and more. As a rule, a library is added to a program to provide more functionality or to automate a process without manually writing new code. This reduces time to market.

Django allows developers to use libraries when building any project. Some popular libraries include the Django REST framework, which is responsible for building application programming interfaces (APIs); Django CMS, which is designed to manage website content; and Django-allauth, which is an integrated set of Django applications for authentication, registration, account management, and third-party (social) account authentication.

ORM

Django is valued for its object-relational mapper that helps developers interact with databases. An object-relational mapper (ORM) is a library that automatically transfers data stored in databases such as PostgreSQL and MySQL into objects commonly used in application code.

7. How to create an environment to start a new project in Django.

1. Install Package. First, install python3-venv package by using the following command.

2. Create a Directory. $ mkdir djangoenv. ...

3. Create Virtual Environment. $ python3 -m venv djangoenv. ...

4. Activate Virtual Environment.

8. What is the difference between project and app.

A *project* refers to the entire application and all its parts.

An *app* refers to a submodule of the project. It's self-sufficient and not intertwined with the other apps in the project such that, in theory, you could pick it up and plop it down into another project without any modification. An *app* typically has its own *models.py* (which might actually be empty). You might think of it as a standalone python module. A simple project might only have one app.

For your example, the *project* is the whole website. You might structure it so there is an *app* for articles, an *app* for ranking tables, and an *app* for fixtures and results. If they need to interact with each other, they do it through well-documented public classes and accessor methods.

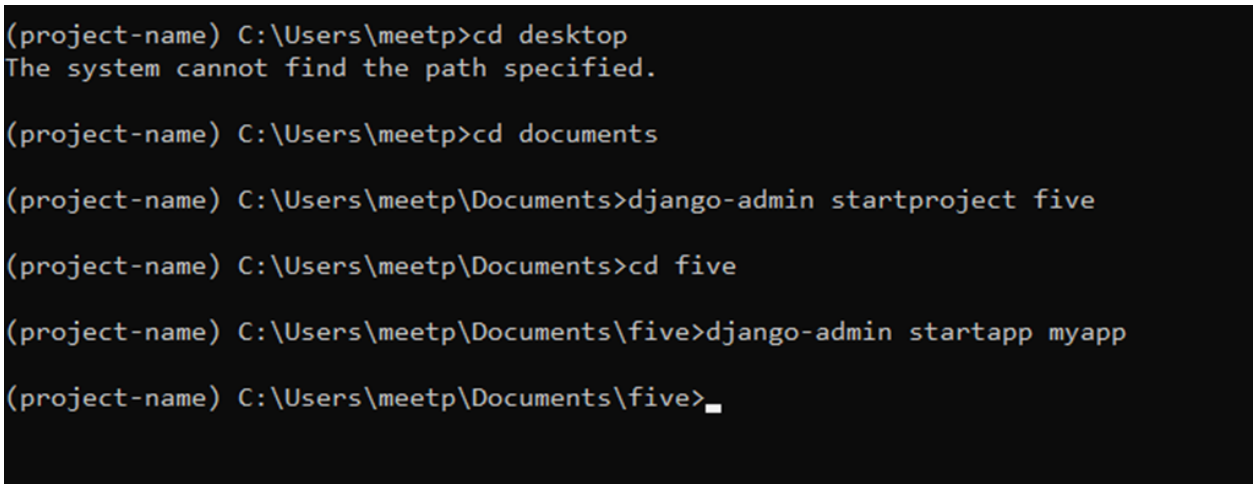
The main thing to keep in mind is this level of interdependence between the *apps*. In practice it's all one *project*, so there's no sense in going overboard, but keep in mind how co-dependent two apps are. If you find one app is solving two problems, split them into two apps. If you find two apps are so intertwined you could never reuse one without the other, combine them into a single app.

WEEK - 10

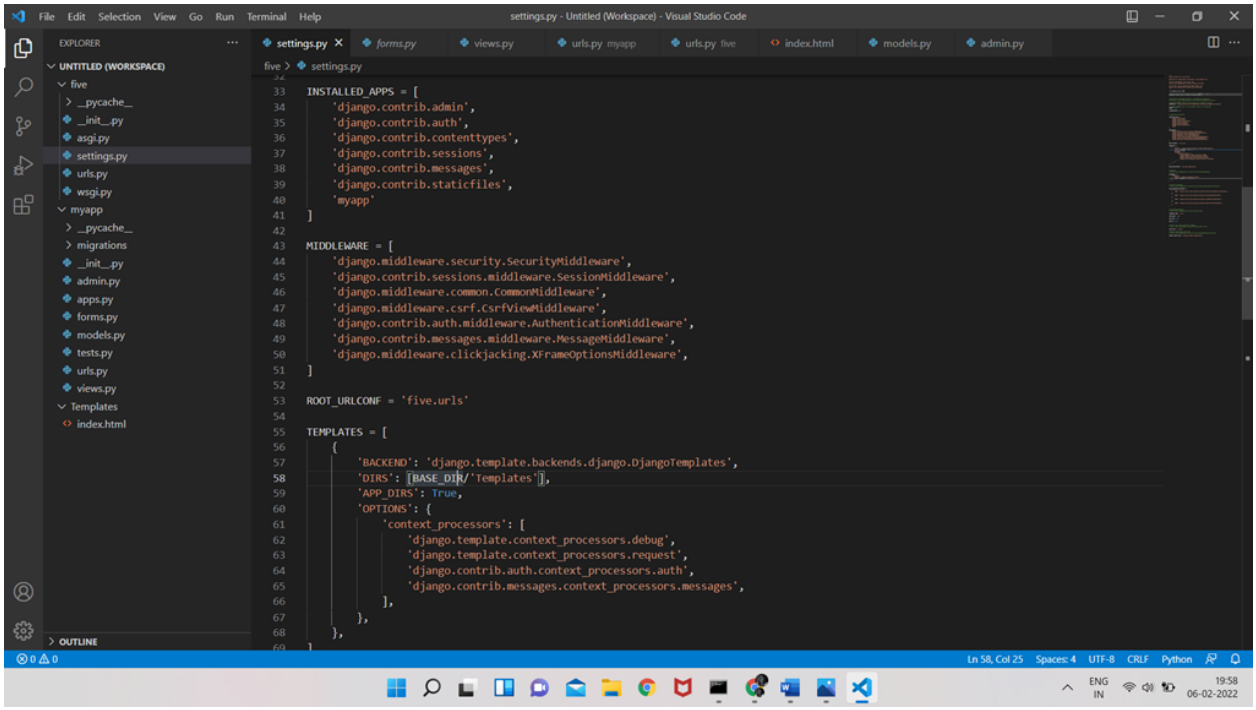
AIM- How to implement a form in django

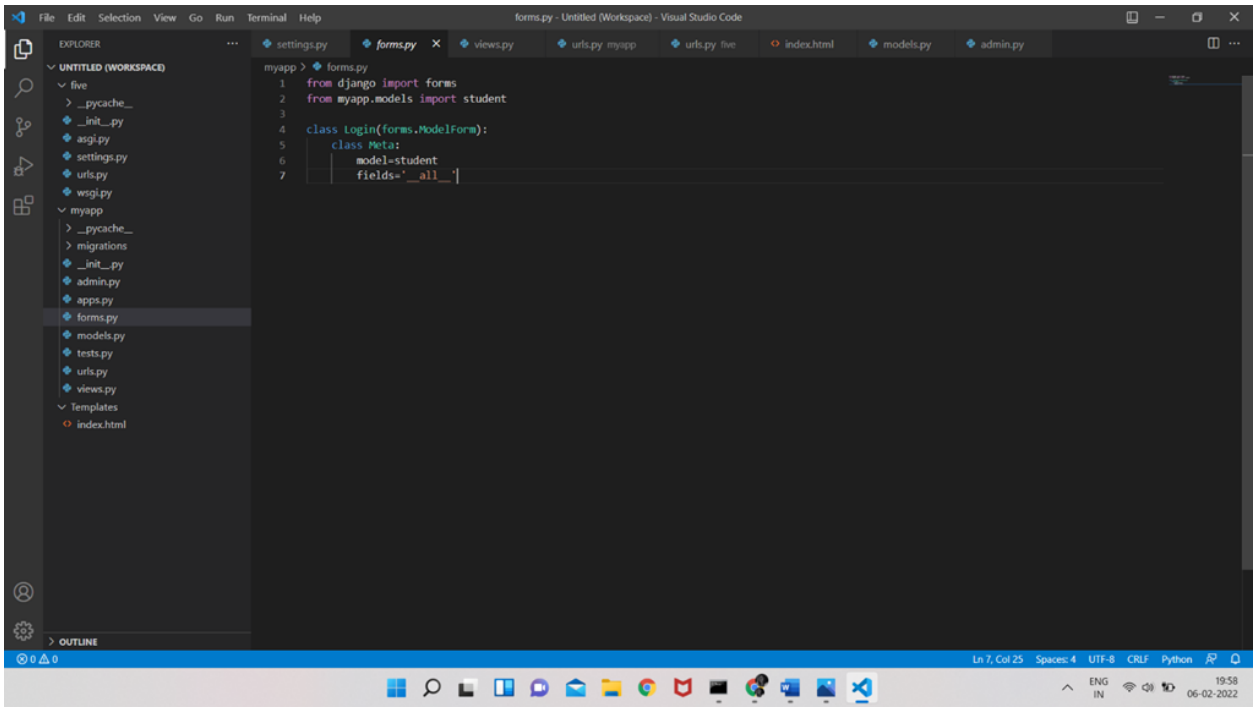
Step:1 - install django in command prompt

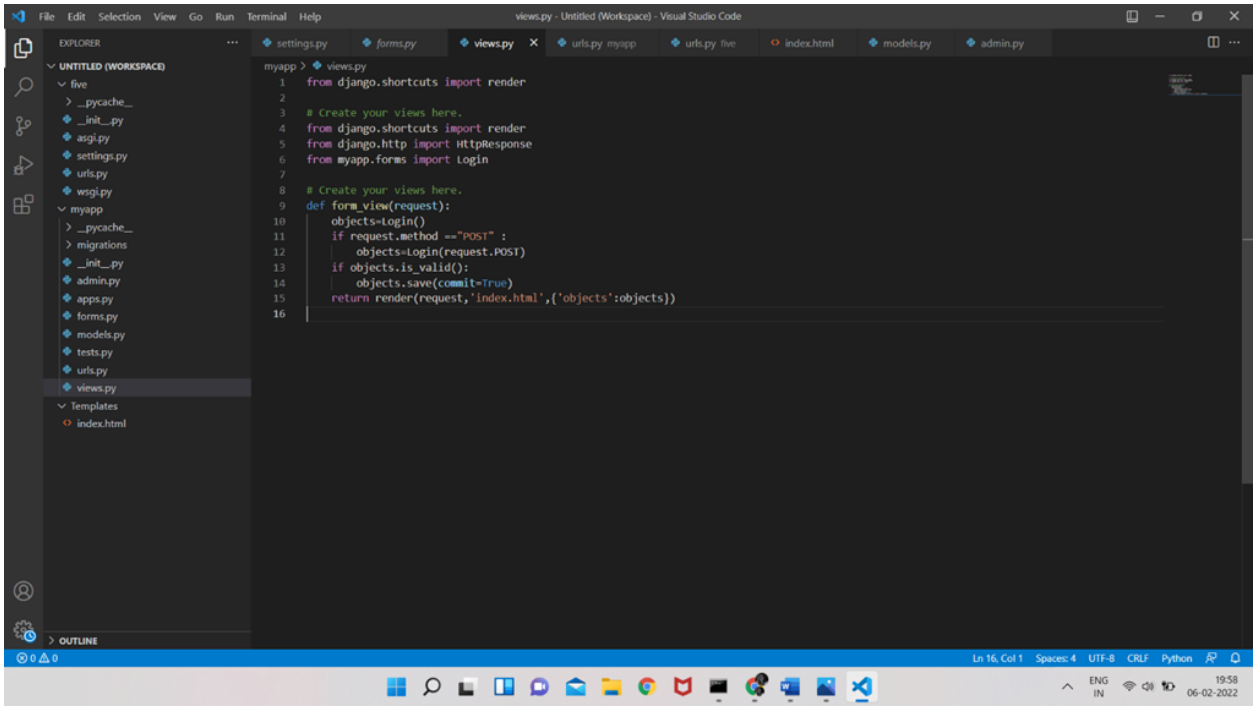
Step:2 - create an environment

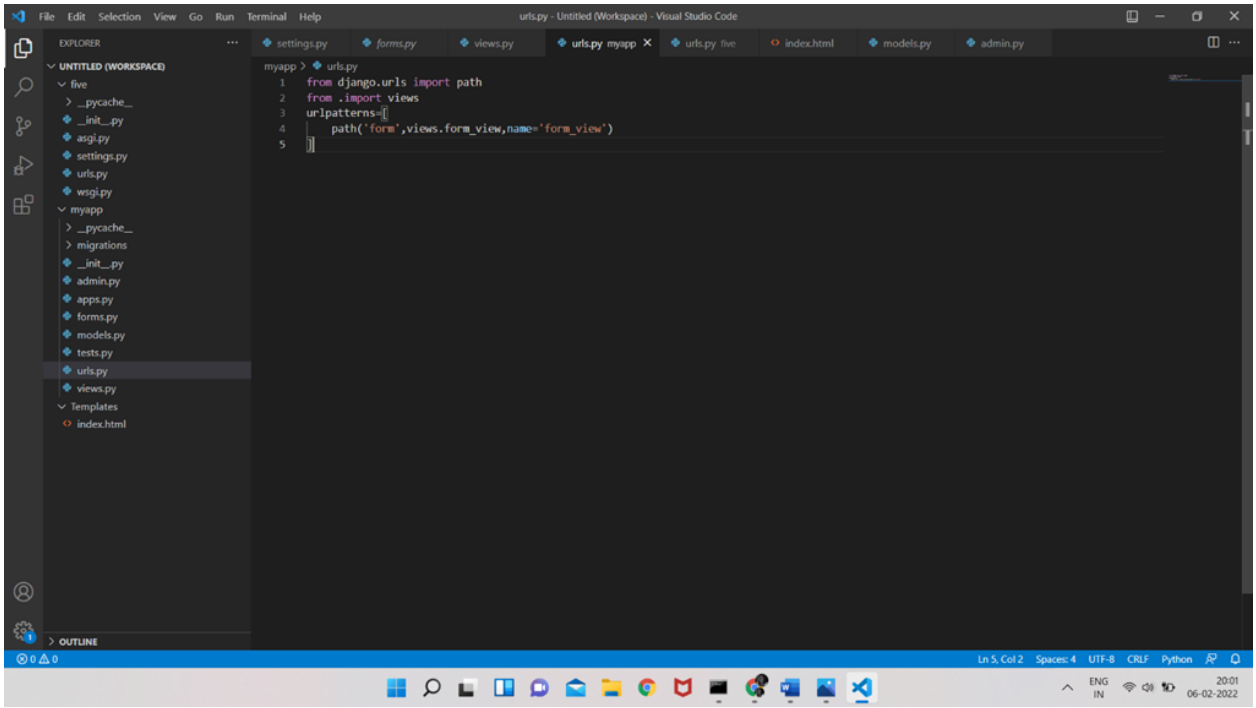


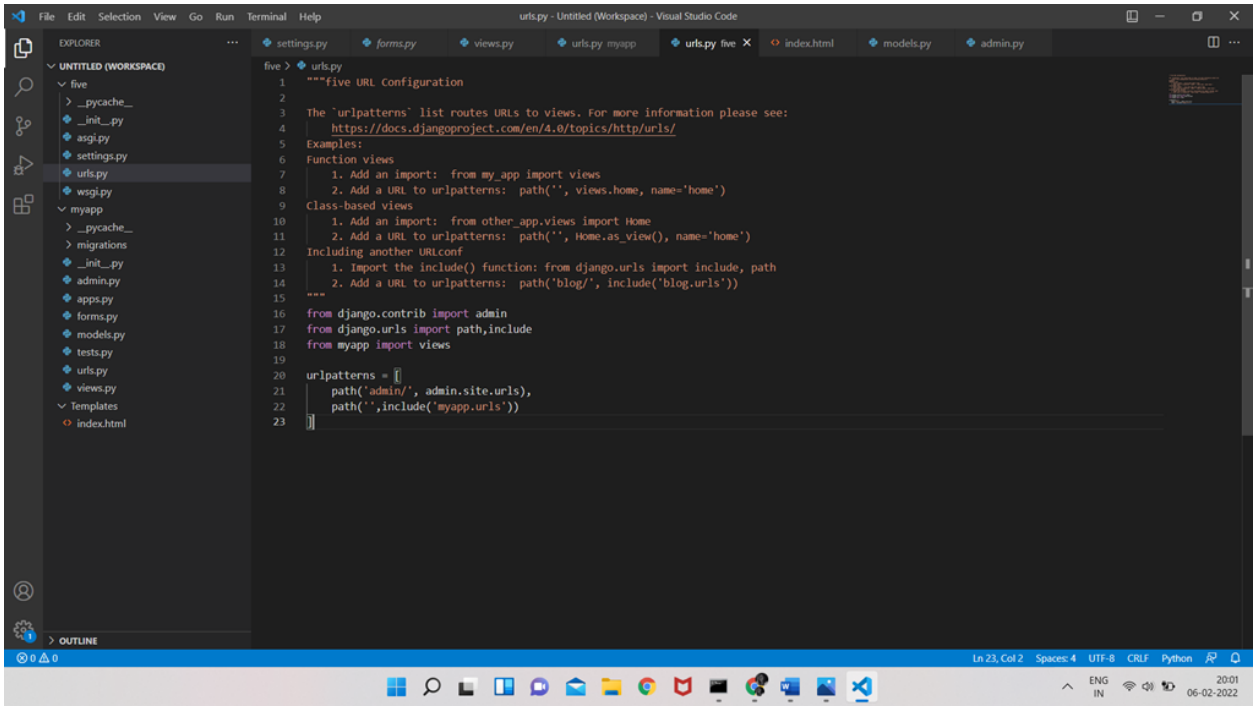
Step:3 - open visual studio , folder five (as per me) in visual studio ,files will be uploaded

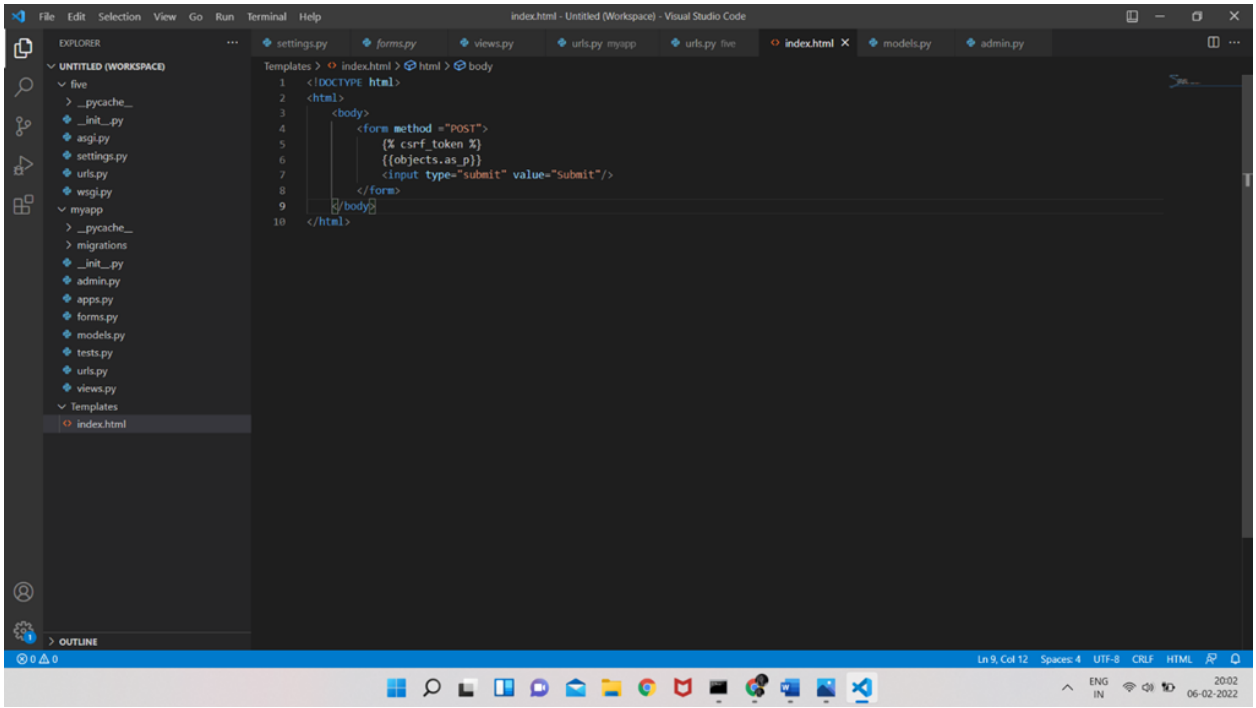


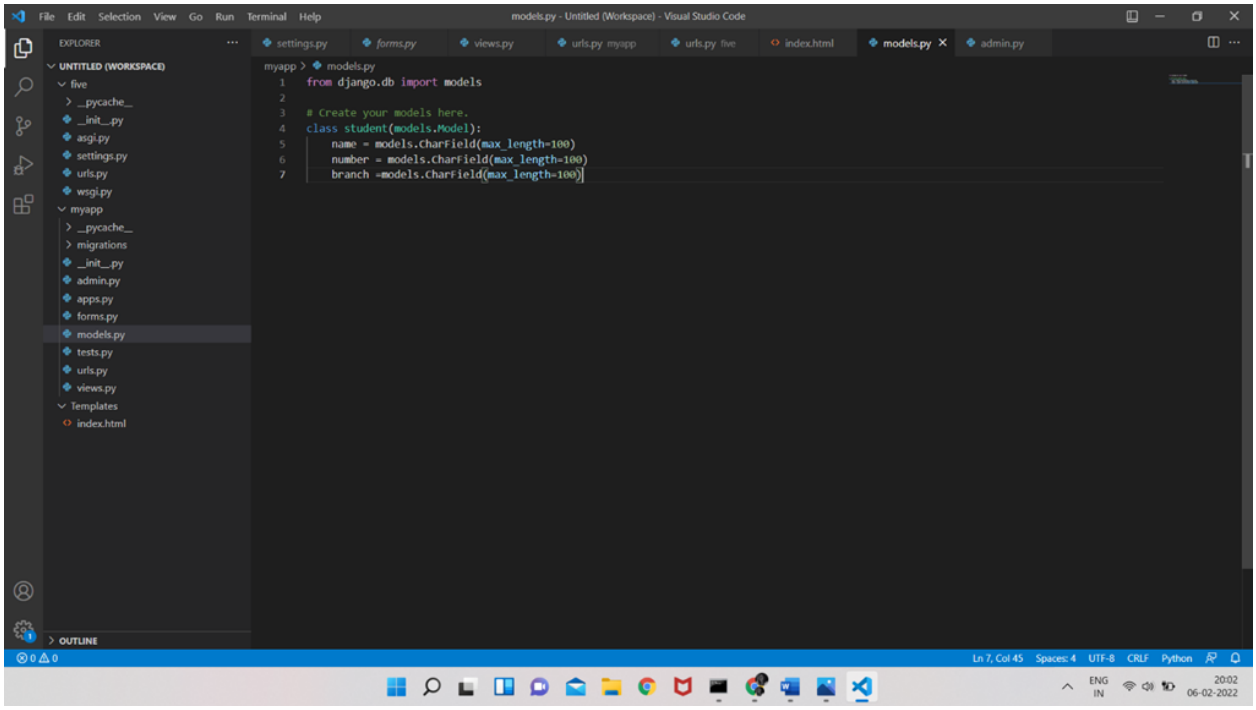


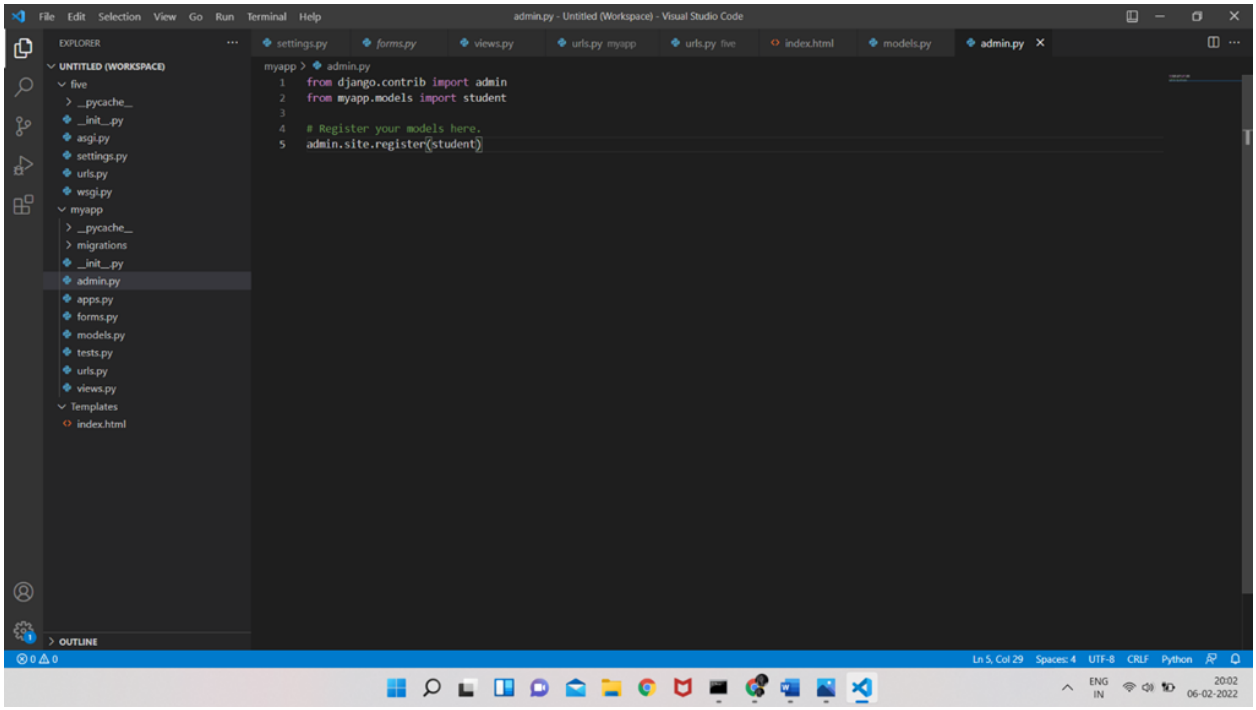

File Edit Selection View Go Run Terminal Help
views.py - Untitled (Workspace) - Visual Studio Code
EXPLORER
UNTITLED (WORKSPACE)
five
__pycache__
__init__.py
asgi.py
settings.py
urls.py
wsgi.py
myapp
__pycache__
migrations
__init__.py
admin.py
apps.py
forms.py
models.py
tests.py
urls.py
views.py
Templates
index.html
settings.py
forms.py
views.py
urls.py myapp
urls.py five
index.html
models.py
admin.py
myapp > views.py
from django.shortcuts import render
# Create your views here.
from django.shortcuts import render
from django.http import HttpResponse
from myapp.forms import Login
# Create your views here.
def form_view(request):
objects=Login()
if request.method =="POST" :
objects=Login(request.POST)
if objects.is_valid():
objects.save(commit=True)
return render(request,'index.html',{'objects':objects})
OUTLINE
Ln 16, Col 1 Spaces: 4 UTF-8 CRLF Python
ENG IN
19:58
06-02-2022


File Edit Selection View Go Run Terminal Help
urls.py - Untitled (Workspace) - Visual Studio Code
EXPLORER
UNTITLED (WORKSPACE)
five
__pycache__
__init__.py
asgi.py
settings.py
urls.py
wsgi.py
myapp
__pycache__
migrations
__init__.py
admin.py
apps.py
forms.py
models.py
tests.py
urls.py
views.py
Templates
index.html
settings.py
forms.py
views.py
urls.py myapp
urls.py five
index.html
models.py
admin.py
myapp > urls.py
from django.urls import path
from .import views
urlpatterns=[
path('form',views.form_view,name='form_view')
]
OUTLINE
Ln 5, Col 2 Spaces: 4 UTF-8 CRLF Python
ENG IN
20:01
06-02-2022


urls.py - Untitled (Workspace) - Visual Studio Code
EXPLORER
UNTITLED (WORKSPACE)
five
__pycache__
__init__.py
asgi.py
settings.py
urls.py
wsgi.py
myapp
__pycache__
migrations
__init__.py
admin.py
apps.py
forms.py
models.py
tests.py
urls.py
views.py
Templates
index.html
OUTLINE
settings.py
forms.py
views.py
urls.py myapp
urls.py five
index.html
models.py
admin.py
five > urls.py
"""five URL Configuration
The `urlpatterns` list routes URLs to views. For more information please see:
https://docs.djangoproject.com/en/4.0/topics/http/urls/
Examples:
Function views
1. Add an import: from my_app import views
2. Add a URL to urlpatterns: path('', views.home, name='home')
Class-based views
1. Add an import: from other_app.views import Home
2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
Including another URLconf
1. Import the include() function: from django.urls import include, path
2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path,include
from myapp import views
urlpatterns = [
path('admin/', admin.site.urls),
path('',include('myapp.urls'))
]
Ln 23, Col 2
Spaces: 4
UTF-8
CRLF
Python
ENG
IN
20:01
06-02-2022


index.html - Untitled (Workspace) - Visual Studio Code
EXPLORER
UNTITLED (WORKSPACE)
five
__pycache__
__init__.py
asgi.py
settings.py
urls.py
wsgi.py
myapp
__pycache__
migrations
__init__.py
admin.py
apps.py
forms.py
models.py
tests.py
urls.py
views.py
Templates
index.html
OUTLINE
settings.py
forms.py
views.py
urls.py myapp
urls.py five
index.html
models.py
admin.py
Templates > index.html > html > body
<!DOCTYPE html>
<html>
<body>
<form method ="POST">
{% csrf_token %}
{{objects.as_p}}
<input type="submit" value="Submit"/>
</form>
</body>
</html>
Ln 9, Col 12
Spaces: 4
UTF-8
CRLF
HTML
ENG
IN
20:02
06-02-2022


File Edit Selection View Go Run Terminal Help
models.py - Untitled (Workspace) - Visual Studio Code
EXPLORER
UNTITLED (WORKSPACE)
five
__pycache__
__init__.py
asgi.py
settings.py
urls.py
wsgi.py
myapp
__pycache__
migrations
__init__.py
admin.py
apps.py
forms.py
models.py
tests.py
urls.py
views.py
Templates
index.html
settings.py
forms.py
views.py
urls.py myapp
urls.py five
index.html
models.py
admin.py
myapp > models.py
1 from django.db import models
2
3 # Create your models here.
4 class student(models.Model):
5 name = models.CharField(max_length=100)
6 number = models.CharField(max_length=100)
7 branch =models.CharField(max_length=100)
OUTLINE
Ln 7, Col 45 Spaces: 4 UTF-8 CRLF Python
ENG IN
20:02 06-02-2022


File Edit Selection View Go Run Terminal Help
admin.py - Untitled (Workspace) - Visual Studio Code
EXPLORER
UNTITLED (WORKSPACE)
five
__pycache__
__init__.py
asgi.py
settings.py
urls.py
wsgi.py
myapp
__pycache__
migrations
__init__.py
admin.py
apps.py
forms.py
models.py
tests.py
urls.py
views.py
Templates
index.html
settings.py
forms.py
views.py
urls.py myapp
urls.py five
index.html
models.py
admin.py
myapp > admin.py
1 from django.contrib import admin
2 from myapp.models import student
3
4 # Register your models here.
5 admin.site.register(student)
OUTLINE
Ln 5, Col 29 Spaces: 4 UTF-8 CRLF Python
ENG IN
20:02 06-02-2022

Step:4 - For execution of the form go to command prompt

type: python manage.py makemigrations

thereafter type: python manage.py migrate

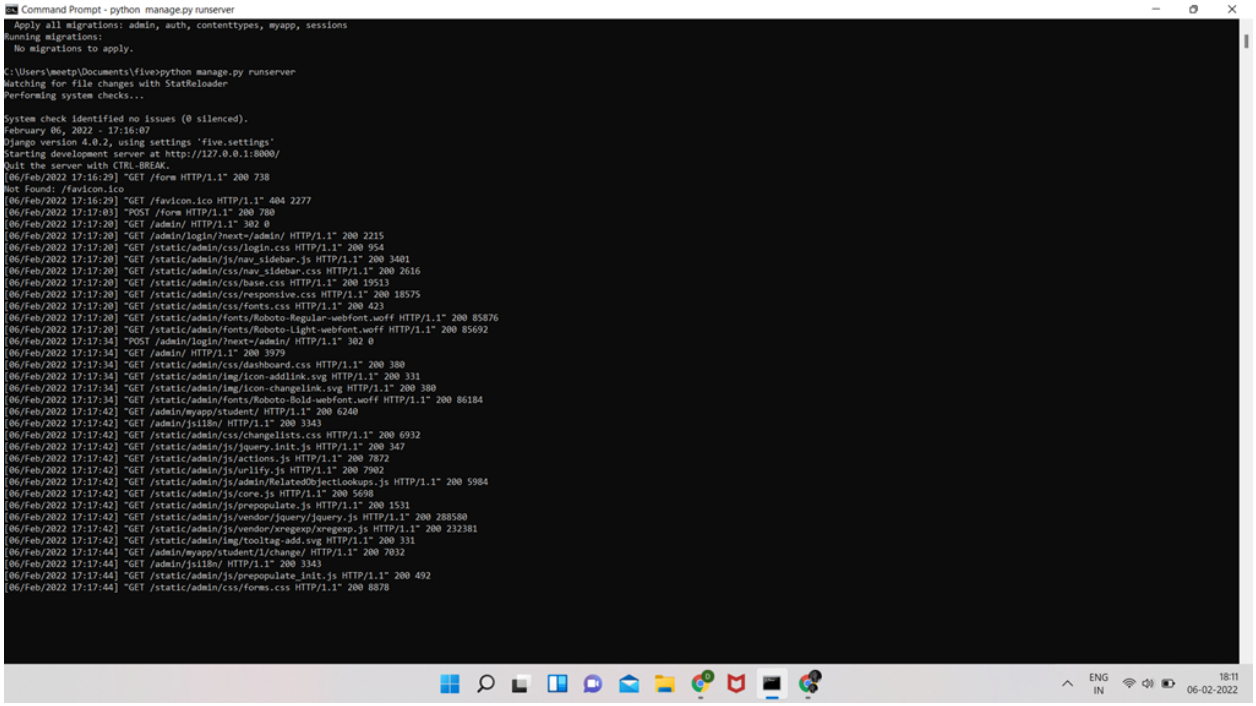
after type:python manage.py runserver

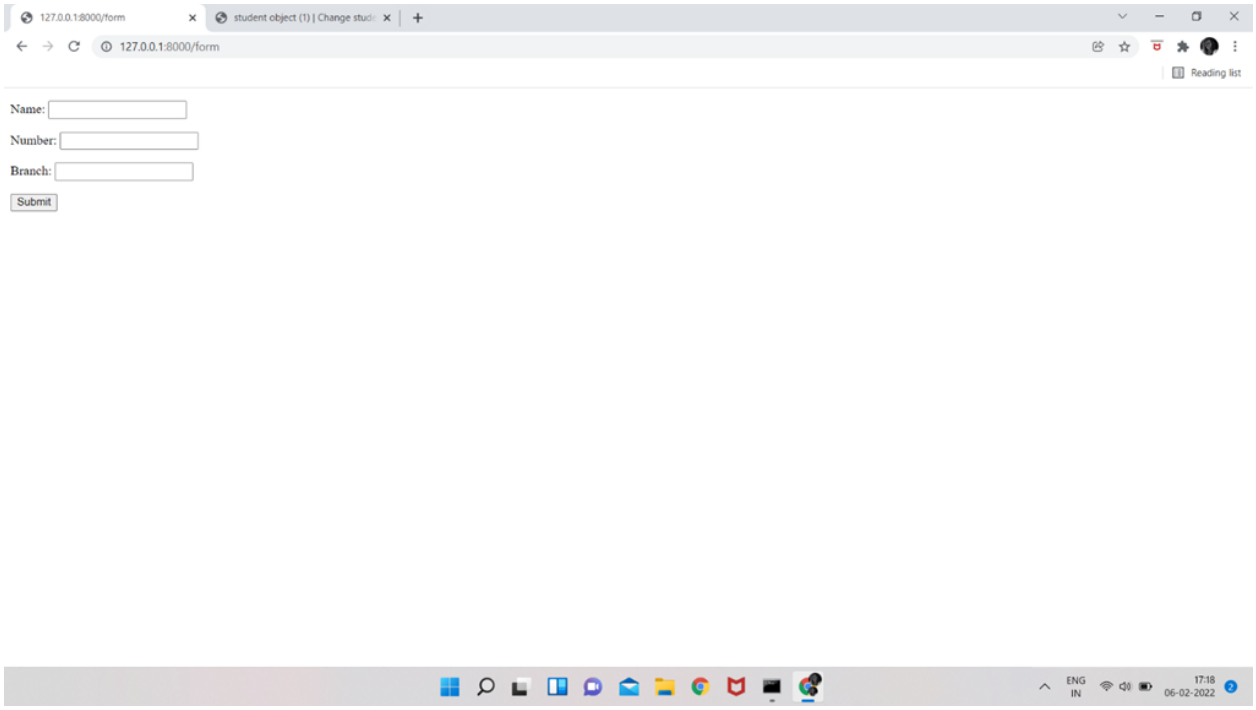
we will get an ip address

copy that and paste it in chrome

add : /form to the address

http://127.0.0.1:8000/form

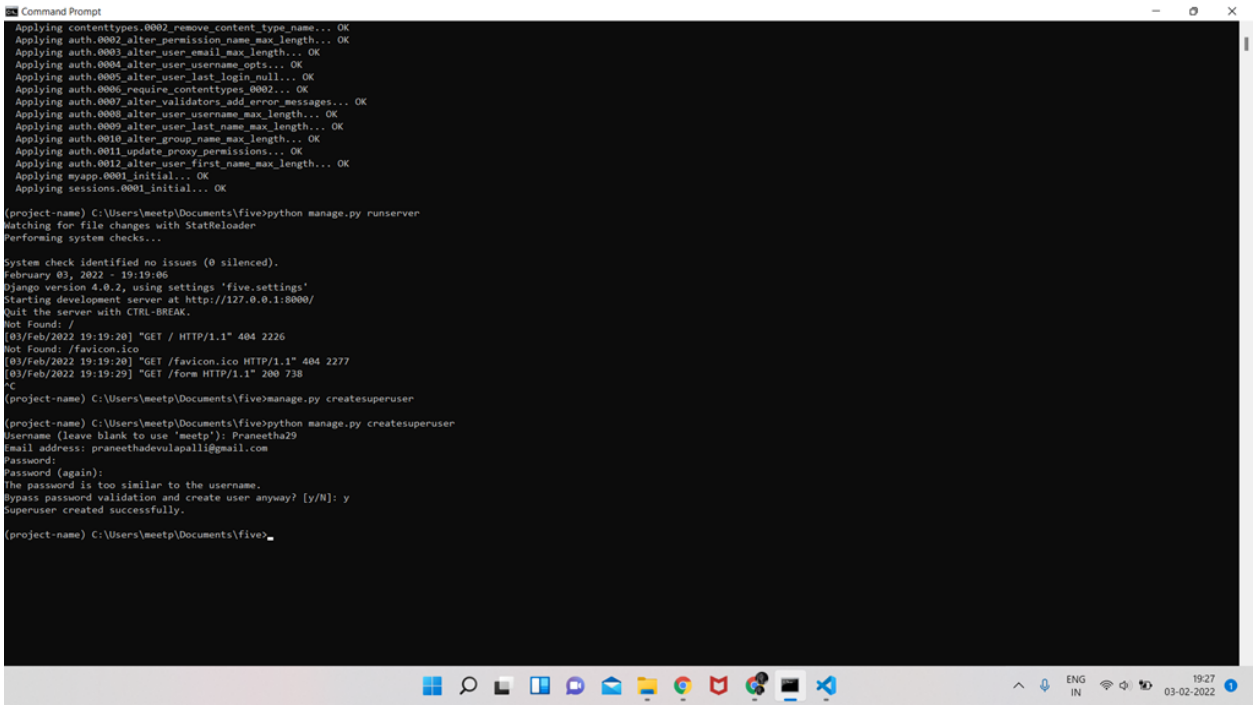



127.0.0.1:8000/form
student object (1) | Change stude
127.0.0.1:8000/form
Reading list
Name:
Number:
Branch:
Submit
ENG
IN
17:18
06-02-2022

Step: 5 - We will get a form like this

thereafter create a user id for django

by typing in cmd prompt : python manage.py createsuperuser



Step:6 - http://127.0.0.1:8000/admin with/admin

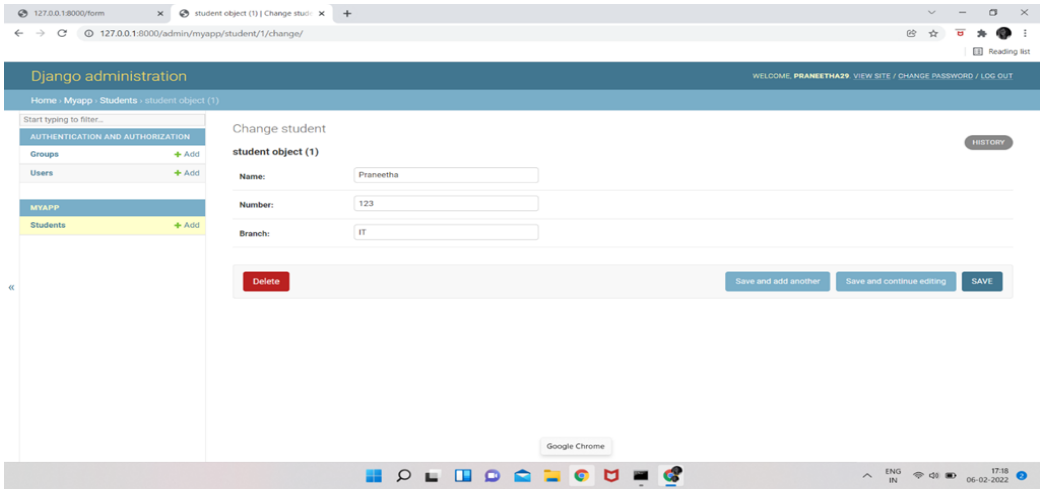
again go to chrome and paste the address

will get a site consist of username and passowed

will get an Django administration site

login

in student option we can cretae as many as student objects and we can delete too



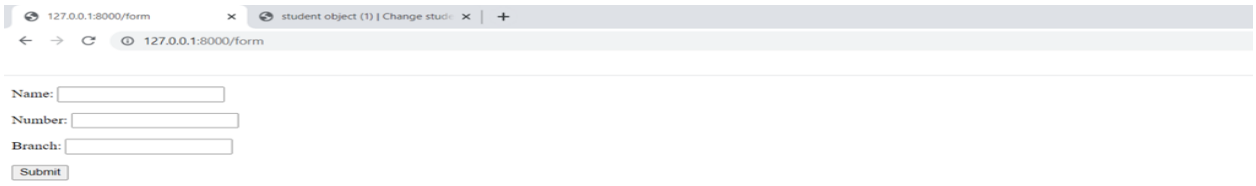
WEEK - 11

1. CRUD operations on the form.

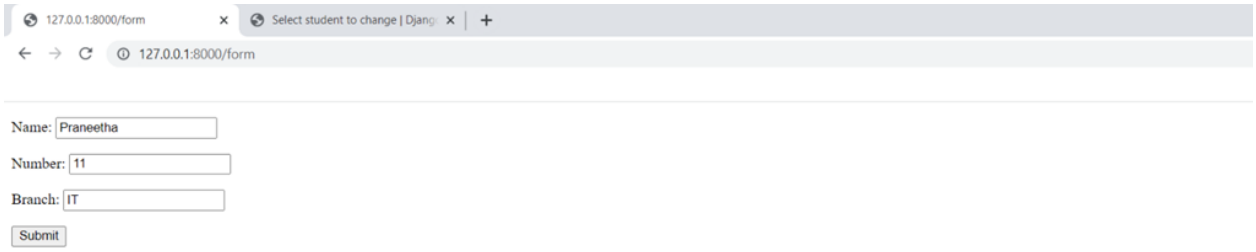
CRUD Meaning: CRUD is an acronym that comes from the world of computer programming and refers to the four functions that are considered necessary to implement a persistent storage application: create, read, update and delete.

Letter	Operation	Function
C	Create	Insert
R	Read	Select
U	Update	Edit
D	Delete	Delete

CREATE :

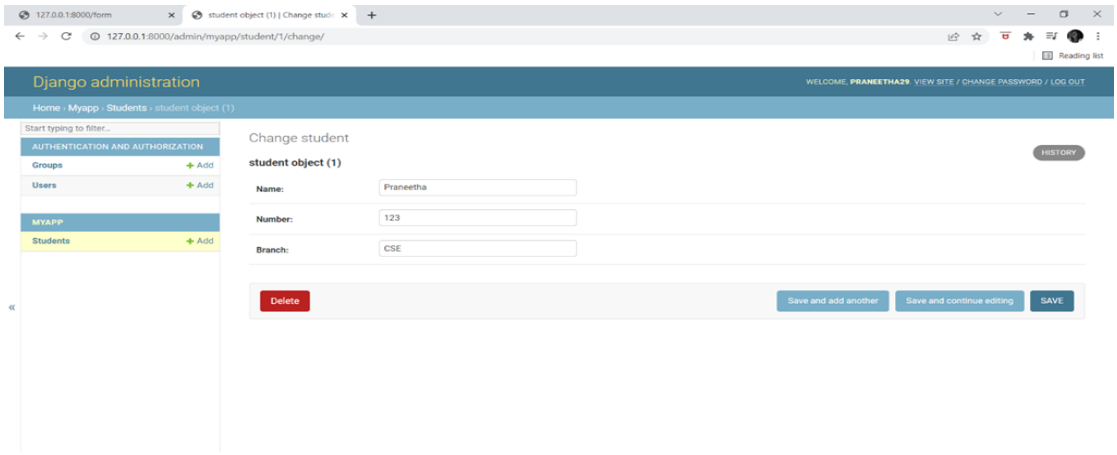

127.0.0.1:8000/form
student object (1) | Change stude
127.0.0.1:8000/form
Name:
Number:
Branch:
Submit

READ

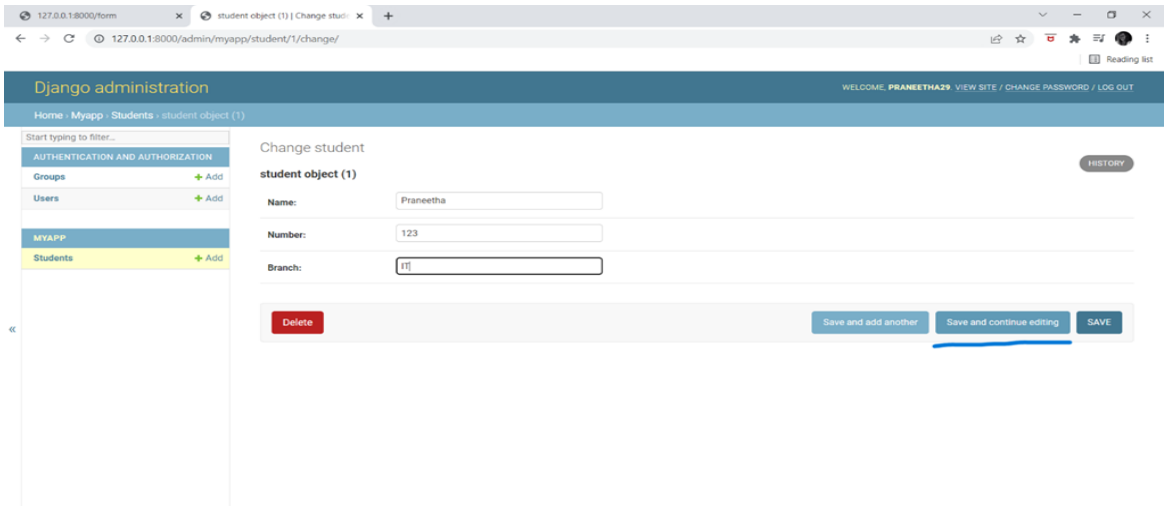

127.0.0.1:8000/form
Select student to change | Djang
127.0.0.1:8000/form
Name: Praneetha
Number: 11
Branch: IT
Submit

UPDATE

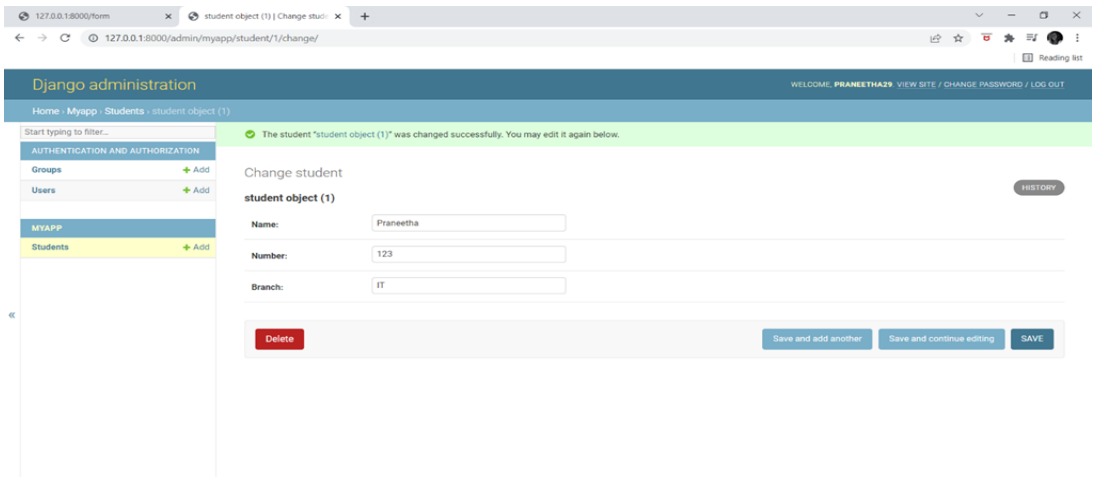
Step 1: Select and open one of the objects created.(say student object (1))



Step 2: Select the Save and continue editing button to update the object.

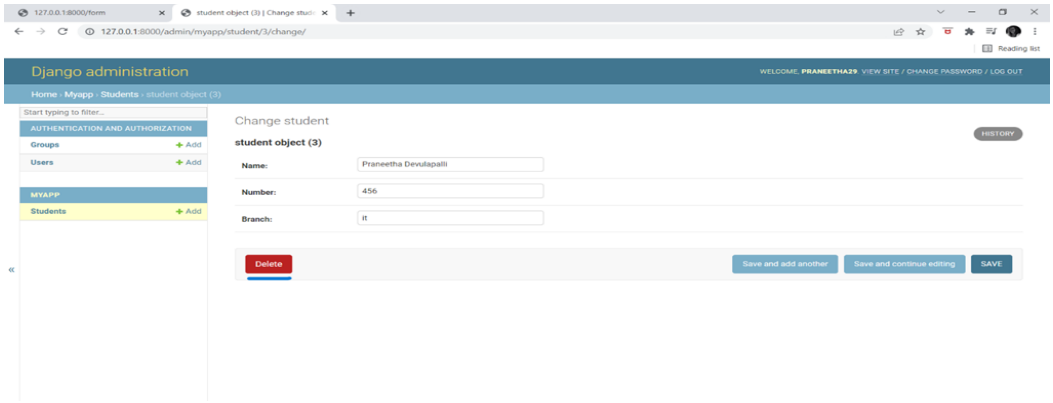


Step 3: The change is updated successfully.



DELETE :

Step 1: Select the student object to delete. (say student object(3))



Step 2: The student “student object(3)” was deleted successfully.