# WEEK – 1

Aim: To understand the Basic concepts the HTML language and heading tags

Description:

**1.What is web programming?**

Web programming refers to the writing, markup and coding involved in Web development, which includes Web content, Web client and server scripting and network security. The most common languages used for Web programming are XML, HTML, JavaScript, Perl 5 and PHP. Web programming is different from just programming, which requires interdisciplinary knowledge on the application area, client and server scripting, and database technology.

To improve user experience and related functionalities on the client side, JavaScript is usually used. It is an excellent client-side platform for designing and implementing Web applications.

HTML5 and CSS3 supports most of the client-side functionality provided by other application frameworks.

**2.what is full form of HTML ? and what is its version?**

* HYPER TEXT MARKUP LANGUAGE .
* current version is HTML5

**3.How do we write code in HTML?**

<html>

<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>

</html>

4.Difference b/w LAN,MAN & VAN?

* A LAN (local area network) is a group of computers and network devices conected together, usually within the same building.
* A MAN (metropolitan area network) is a larger network that usually spans several buildings in the same city or town.
* A WAN (wide area network), in comparison to a MAN, is not restricted to a geographical location, although it might be confined within the bounds of a stateor country. A WAN connects several LANs, and may be limited to an enterprise (a corporation or an organization) or accessible to the public. The technology is high speed and relatively expensive. The Internet is an example of a worldwide public WAN.

**5. Difference between physical address and logical address**

* The logical address is a virtual address and can be viewed by the user. ... The fundamental difference between logical and physical address is that logical address is generated by CPU during a program execution whereas, the physical address refers to a location in the memory unit.

**6.What are h1-h6 tags in HTML?**

<!DOCTYPE html>

<html>

<body>

<h1>Heading 1</h1>

<h2>Heading 2</h2>

<h3>Heading 3</h3>

<h4>Heading 4</h4>

<h5>Heading 5</h5>

<h6>Heading 6</h6>

</body>

</html>

**7. How many types of tags are there in HTML and what are they ?**

**1. Paired and Unpaired Tags**

**Paired Tags** : <p> This text is a paragraph . </p>

**Unpaired Tags :**

<p> This is a paragraph </p>

<hr>

<i> <b> This is a bold and italicized text </b> </i>

**2. Self-Closing Tags**

<img src="a.jpg" alt="This is an alternate text">

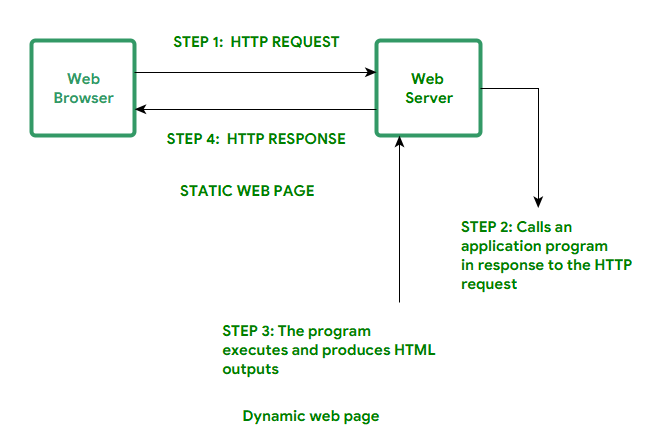
**3. Utility-Based Tags:**

* This is done using tags like <font>, <b>, <u>, etc. Tables, divisions, and span tags are also those tags that help format a web page or document and set the layout of the page.

**Control Tags**

* Another category of tags that can be created is ‘Control Tags’. The Script tags, radio buttons or checkboxes, the Form tags, etc., forms the control tags

**8. what is a web page and what are the different types of web pages?**

****

* web page is a document available on world wide web. Web Pages are stored on web server and can be viewed using a web browser. A web page can cotain huge information including text, graphics, audio, video and hyper links. These hyper links are the link to other web pages

# -> Static Web pages:

Static Web pages are very simple. It is written in languages such as HTML, JavaScript, CSS, etc. For static web pages when a server receives a request for a web page, then the server sends the response to the client without doing any additional process. And these web pages are seen through a web browser. In static web pages, Pages will remain the same until someone changes it manually.

## -> Dynamic Web Pages:

Dynamic Web Pages are written in languages such as CGI, AJAX, ASP, ASP.NET, etc. In dynamic web pages, the Content of pages is different for different visitors. It takes more time to load than the static web page. Dynamic web pages are used where the information is changed frequently, for example, stock prices, weather information, etc.

**9.what is a website ?**

A website is a collection of web pages and related content that is identified by a common domain name and published on at least one web server. Notable examples are wikipedia.org, google.com, and amazon.com. All publicly accessible websites collectively constitute the World Wide Web

**Program:**

<html>

<h1> M.J.N.V.SAI </h1>

<h2> M.J.N.V.SAI </h2>

<h3> M.J.N.V.SAI </h3>

<h4> M.J.N.V.SAI </h4>

<h5> M.J.N.V.SAI </h5>

<h6> M.J.N.V.SAI </h6>

</html>

**OUTPUT:**

****

# WEEK – 2

Aim: To design the homepage with image tags and paragraph,bold,italic tags.

Description:

1. **What is the difference between b/n heading , title , body?**

Heading fonts and body fonts are part of the templates available to you as part of your brand.

Heading fonts are used as larger, higher impact text, such as in a title or section header

Body fonts are usually used with longer text, like a paragraph or secondary text.  We recommend an easy-to-read font that pairs well with your heading font. You may want to choose an easy to read font that compliments your heading font choice. Body fonts also support bold and italics styles.  If you’d like you can select your regular font, and we will build the italics and bold styles automatically for you, but your text will look its best when you also upload the font files all the text styling options.

1. **What is the difference between web app****lication and mobile application**

## Native apps:

They are installed onto the device itself and are developed especially for a particular mobile operating system. These apps are available on app stores such as Apple App Store, Google Play Store, etc.

Native apps work in specific mobile operating systems such as Apple iOS or Android OS.

If an app made for Android OS then it will not work on Apple iOS or Windows OS. We have to build separate apps for each operating system if we want to work our app across all major operating systems. This means we have to spend more money and more effort (time, resources).

* Native apps are faster than web apps.
* Native apps can access system/device resources such as a GPS or camera.
* These apps can work without an internet connections

## Web Apps:

Web Apps can be accessed through the mobile device’s Web browser, Web Apps are based on internet-enabled applications. In order to access we don’t need to download and install the app onto a mobile device.

The app is developed as web pages in HTML and CSS, interactive parts are developed in JQuery, JavaScript, or similar language. The single web app can be used on most devices capable of surfing the web, it does not depend upon the operating system they use.

* Web Apps do not work without an internet connection.
* Web apps work slower than mobile apps.
* It is difficult to discover web apps since they aren’t hosted in a specific database like an app store

1. **List out various types of tags?.**

## Various types of Tags:

* **Title tag:**

The <title> tag defines the title of the document. The title must be text-only, and it is shown in the browser's title bar or in the page's tab.

The <title> tag is required in HTML documents!

The contents of a page title is very important for search engine optimization (SEO)! The page title is used by search engine algorithms to decide the order when listing pages in search results.

The <title> element:

* defines a title in the browser toolbar
* provides a title for the page when it is added to favorites
* displays a title for the page in search-engine resume
* **Strong Tag:**

The <strong> tag is used to define text with strong importance. The content inside is typically displayed in **bold**.

* **Center Tag:**

<center> ……………….. </center> : places at the center of web-page

* **Small tags:**

1. <b> ………………. </b> : bolds the string
2. <U> ………………………….. </U> : underlines the string
3. <i> …………………………. </i> : string will be in italic style

**Image tag:**

<img src “pic url” align=”right” or align=”left” or align=”center” width=”100” height=”100” >

Src : image source

Width & height : they are image dimensions

**Hyper – link tag:**

<a href=”link url”> link-name </a>

**Program:**

|  |
| --- |
|  |
|  | <html> |
|  | <head> |
|  | <title> RESUME: M.Sai </title> |
|  |  |
|  | </head> |
|  |  |
|  | <body style="background-color:#30bc9242"> |
|  |  |
|  | <center> <h1 style="color:blue"> <b> <U> <i> RESUME </i> </U> </b> </h1> </center> |
|  |  |
|  | <h2> <strong> MOTAMARRI JAYA NAGA VENKATA SAI </strong> </h2> |
|  |  |
|  | <img src = "<https://bit.ly/3qVrg6X>" align = "right" width = "500" height = "500"> |
|  |  |
|  | <p> <b> Email: sai.doc45@gmail.com </b> </p> |
|  | <p> Contact number: 8331062449 </p> |
|  | <p> Codechef username: msai\_45 </p> |
|  | <p> Facebook username: mjnv Sai </p> |
|  |  |
|  | <p> <a href="<https://www.facebook.com>"> myfacebook.com </a> </p> |
|  | <p> <a href="<https://mjnvsai.blogspot.com/>"> myblog.com </a> </p> |
|  | <p> <a href="<https://github.com/MJNVSai>"> mygithub.com </a> </p> |
|  |  |
|  | <p> At present i am studying 2nd year of B.Tech at V.R.S.E.C in the Branch of Information Technology After then i have completed my intermediate in K.B.N Shine college and also i have completed 10th Class in Ravindra Bharathi Public School in Bhavanipuram Branch, at present My short-term goal is to Become Best in Coding and My middle-term goal is To get best rank in any Coding platform and My long-term goal is to get placed in AMAZON Company and my hobbies are listning music, watching Anime and movies, Imporving My coding skills in leishure time...</p> |
|  |  |
|  | <p> DOB: 15-01-2002 </p> |
|  | <p> Age: 19 </p> |
|  | <p> Fathers name: M.Krishna Rao </p> |
|  | <p> Address: 17-76-B/1,kothapeta, 1Town </p> |
|  |  |
|  | </body> |
|  |  |
|  | </html> |

**OUTPUT:**



# WEEK-3

Aim: Create an Ordered and unordered Lists and Insert an Video,audio Into an webpage

Description:

## Lists:

What are lists in HTML language and types of Lists?

There are 2 types of lists and they are:

* Ordered Lists
* Unordered Lists

**Ordered Lists:**

* An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.The list items will be marked with numbers by default.
* The HTML <ol> tag defines an ordered list. An ordered list can be numerical or alphabetical.

**Type attributes in Ordered list:**

The type attribute of the <ol> tag, defines the type of the list item marker:

|  |  |
| --- | --- |
| **Type** | **Description** |
| type="1" | The list items will be numbered with numbers (default) |
| type="A" | The list items will be numbered with uppercase letters |
| type="a" | The list items will be numbered with lowercase letters |
| type="I" | The list items will be numbered with uppercase roman numbers |
| type="i" | The list items will be numbered with lowercase roman numbers |

**Syntax: (ordered list)**

<ol type=” ”>

<li>………………………………….</li>

<li> …………………………………..</li>

<li>………………………………….</li>

<ol>

* Use the HTML <ol> element to define an ordered list
* Use the HTML type attribute to define the numbering type
* Use the HTML <li> element to define a list item
* Lists can be nested
* List items can contain other HTML elements

## Unordered List:

* An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.
* The list items will be marked with bullets (small black circles) by default
* The HTML <ul> tag defines an unordered (bulleted) list.

**Type attributes in unordered lists:**

* The type attribute of the <ul> tag, defines the type of the list item marker:

|  |  |
| --- | --- |
| **Value** | **Description** |
| disc | Sets the list item marker to a bullet (default) |
| circle | Sets the list item marker to a circle |
| square | Sets the list item marker to a square |
| none | The list items will not be marked |

**Unordered lists syntax:**

<ul type=””>

<li>……………………………….</li>

<li>………………………………..</li>

<li>…………………………………</li>

<ul>

**Video Tag:**

The <video> tag is used to embed video content in a document, such as a movie clip or other video streams.

The <video> tag contains one or more [<source>](https://www.w3schools.com/tags/tag_source.asp) tags with different video sources. The browser will choose the first source it supports.

The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.

There are three supported video formats in HTML: MP4, WebM, and OGG.

**Attributes in video tag:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| [autoplay](https://www.w3schools.com/tags/att_video_autoplay.asp) | autoplay | Specifies that the video will start playing as soon as it is ready |
| [controls](https://www.w3schools.com/tags/att_video_controls.asp) | controls | Specifies that video controls should be displayed (such as a play/pause button etc). |
| [height](https://www.w3schools.com/tags/att_video_height.asp) | *pixels* | Sets the height of the video player |
| [loop](https://www.w3schools.com/tags/att_video_loop.asp) | loop | Specifies that the video will start over again, every time it is finished |
| [muted](https://www.w3schools.com/tags/att_video_muted.asp) | muted | Specifies that the audio output of the video should be muted |
| [poster](https://www.w3schools.com/tags/att_video_poster.asp) | *URL* | Specifies an image to be shown while the video is downloading, or until the user hits the play button |
| [preload](https://www.w3schools.com/tags/att_video_preload.asp) | auto metadata none | Specifies if and how the author thinks the video should be loaded when the page loads |
| [src](https://www.w3schools.com/tags/att_video_src.asp) | *URL* | Specifies the URL of the video file |
| [width](https://www.w3schools.com/tags/att_video_width.asp) | *pixels* | Sets the width of the video player |

**Syntax for Video tag:**

<video width = “ “ height = “ ” poster = “.png/.jpeg/url“ controls>

<source src = “url or path of file.mp4” type = “video/mp4”>

(or)

<source src = “url or path of file.ogg” type = “video/ogg”>

</video>

**Audio Tag:**

The <audio> tag is used to embed sound content in a document, such as music or other audio streams.

The <audio> tag contains one or more [<source>](https://www.w3schools.com/tags/tag_source.asp) tags with different audio sources. The browser will choose the first source it supports.

The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element.

There are three supported audio formats in HTML: MP3, WAV, and OGG.

**Attributes in Audio tag:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| [autoplay](https://www.w3schools.com/tags/att_audio_autoplay.asp) | autoplay | Specifies that the audio will start playing as soon as it is ready |
| [controls](https://www.w3schools.com/tags/att_audio_controls.asp) | controls | Specifies that audio controls should be displayed (such as a play/pause button etc) |
| [loop](https://www.w3schools.com/tags/att_audio_loop.asp) | loop | Specifies that the audio will start over again, every time it is finished |
| [muted](https://www.w3schools.com/tags/att_audio_muted.asp) | muted | Specifies that the audio output should be muted |
| [preload](https://www.w3schools.com/tags/att_audio_preload.asp) | auto metadata none | Specifies if and how the author thinks the audio should be loaded when the page loads |
| [src](https://www.w3schools.com/tags/att_audio_src.asp) | *URL* | Specifies the URL of the audio file |

**Syntax for Audio Tag:**

<audio controls>

<source src = “url or path of file.mp3” type = “audio/mpeg”>

(or)

<source src = “url or path of file.ogg” type = “audio/ogg”>

</audio>

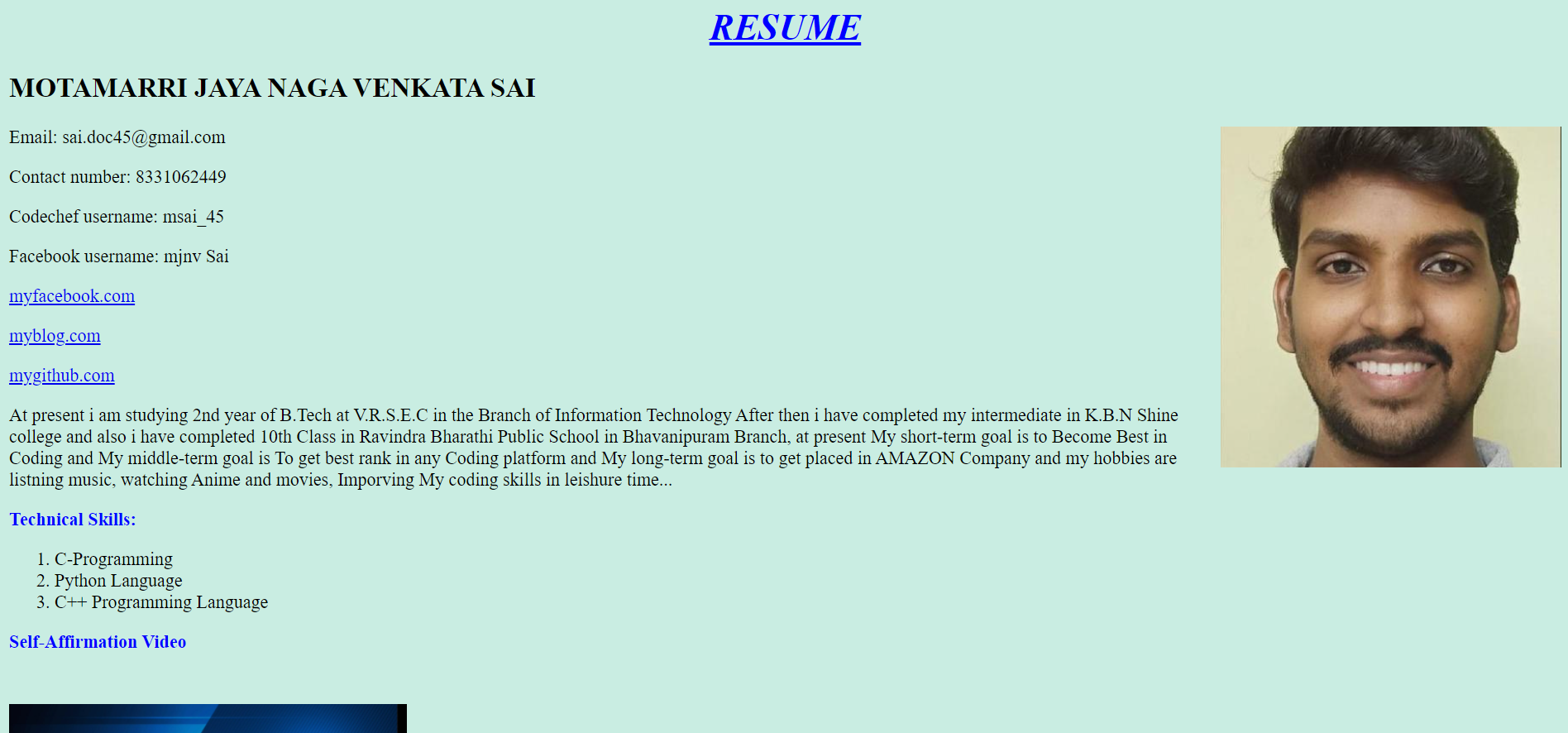
**Program-1:**

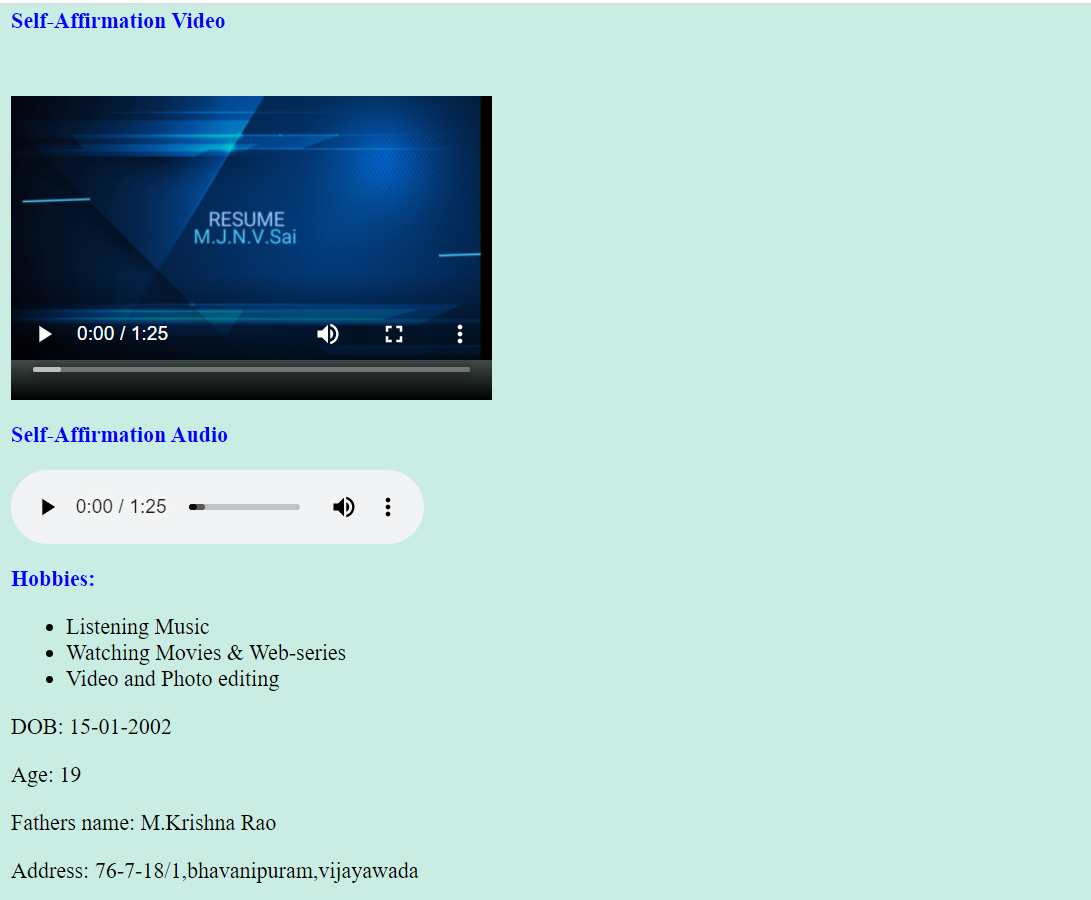
|  |
| --- |
|  |
|  | <html> |
|  | <head> |
|  | <title> RESUME: M.Sai </title> |
|  |  |
|  | </head> |
|  |  |
|  | <body style="background-color:#30bc9242"> |
|  |  |
|  | <center> <h1 style="color:blue"> <b> <U> <i> RESUME </i> </U> </b> </h1> </center> |
|  |  |
|  | <h2> <strong> MOTAMARRI JAYA NAGA VENKATA SAI </strong> </h2> |
|  |  |
|  | <img src = "<https://bit.ly/3qVrg6X>" align = "right" width = "300" height = "300"> |
|  |  |
|  | <p> Email: sai.doc45@gmail.com </p> |
|  | <p> Contact number: 8331062449 </p> |
|  | <p> Codechef username: msai\_45 </p> |
|  | <p> Facebook username: mjnv Sai </p> |
|  |  |
|  | <p> <a href="[https://www.facebook.com](https://www.facebook.com/)"> myfacebook.com </a> </p> |
|  | <p> <a href="<https://mjnvsai.blogspot.com/>"> myblog.com </a> </p> |
|  | <p> <a href="<https://github.com/MJNVSai>"> mygithub.com </a> </p> |
|  |  |
|  | <p> At present i am studying 2nd year of B.Tech at V.R.S.E.C in the Branch of Information Technology After then i have completed my intermediate in K.B.N Shine college and also i have completed 10th Class in Ravindra Bharathi Public School in Bhavanipuram Branch, at present My short-term goal is to Become Best in Coding and My middle-term goal is To get best rank in any Coding platform and My long-term goal is to get placed in AMAZON Company and my hobbies are listning music, watching Anime and movies, Imporving My coding skills in leishure time...</p> |
|  |  |
|  | <p style="color:blue"> <b> Technical Skills: </b> </p> |
|  | <ol type="1"> |
|  | <li> C-Programming </li> |
|  | <li> Python Language </li> |
|  | <li> C++ Programming Language </li> |
|  | </ol> |
|  |  |
|  | <p style="color:blue"> <b> Self-Affirmation Video </b> </p> |
|  | <video width = "350" height = "250" poster = "https://bit.ly/3lT6NME" controls> |
|  | <source src = "<https://mjnvsai.github.io/VRSEC/Web-programming/HTML/self.mp4>" type = "video/mp4"> |
|  | </video> |
|  |  |
|  | <p style="color:blue"> <b> Self-Affirmation Audio </b> </p> |
|  | <audio controls> |
|  | <source src = "<https://mjnvsai.github.io/VRSEC/Web-programming/HTML/self-introduct-sai.mp3>" type = "audio/mp3"> |
|  | </audio> |
|  |  |
|  | <p style="color:blue"> <b> Hobbies: </b> </p> |
|  | <ul type="disc"> |
|  | <li> Listening Music </li> |
|  | <li> Watching Movies & Web-series </li> |
|  | <li> Video and Photo editing</li> |
|  | </ul> |
|  |  |
|  | <p> DOB: 15-01-2002 </p> |
|  | <p> Age: 19 </p> |
|  | <p> Fathers name: M.Krishna Rao </p> |
|  | <p> Address: 76-7-18/1,bhavanipuram,vijayawada </p> |
|  |  |
|  | </body> |
|  |  |
|  | </html> |

**Program-2:**

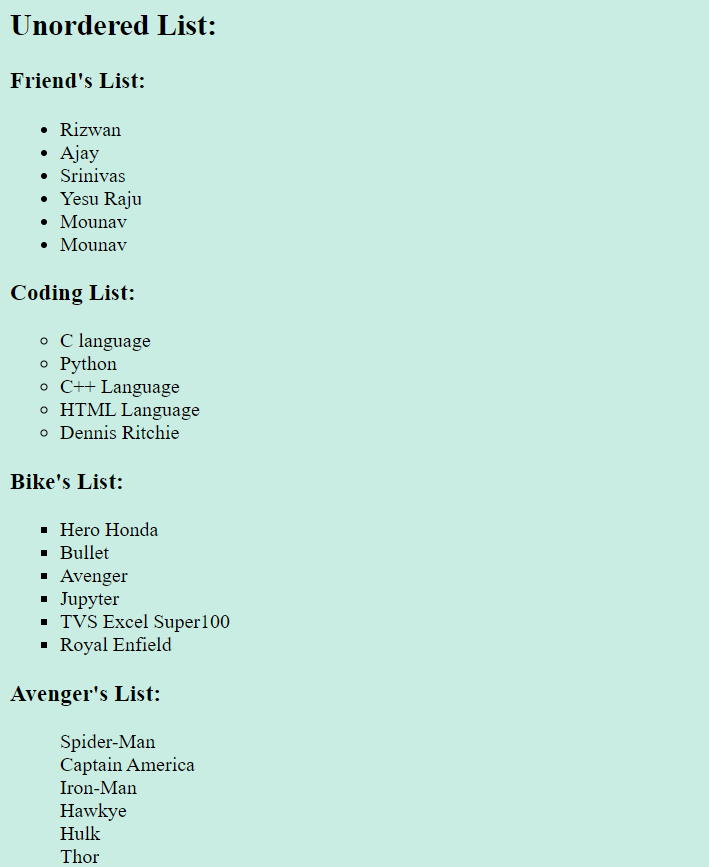
|  |
| --- |
|  |
|  | <html>  <body style="background-color:#30bc9242"> |
|  |  |
|  | <h1 style="color:green"> <center> <U> Different Types of list's in HTML </U> </center> </h1> |
|  |  |
|  | <h2> Ordered List: </h2> |
|  |  |
|  | <h3> Friend's List: </h3> |
|  | <ol type="1"> |
|  | <li> Rizwan </li> |
|  | <li> Ajay </li> |
|  | <li> Srinivas </li> |
|  | <li> Yesu Raju </li> |
|  | <li> Mounav </li> |
|  | <li> Mounav </li> |
|  | </ol> |
|  |  |
|  | <h3> Coding List: </h3> |
|  | <ol type="A"> |
|  | <li> C language </li> |
|  | <li> Python </li> |
|  | <li> C++ Language </li> |
|  | <li> HTML Language </li> |
|  | <li> Dennis Ritchie </li> |
|  | </ol> |
|  |  |
|  | <h3> Bike's List: </h3> |
|  | <ol type="I"> |
|  | <li> Hero Honda </li> |
|  | <li> Bullet </li> |
|  | <li> Avenger </li> |
|  | <li> Jupyter </li> |
|  | <li> TVS Excel Super100 </li> |
|  | <li> Royal Enfield </li> |
|  | </ol> |
|  |  |
|  | <h3> Avenger's List: </h3> |
|  | <ol type="a"> |
|  | <li> Spider-Man </li> |
|  | <li> Captain America </li> |
|  | <li> Iron-Man </li> |
|  | <li> Hawkye </li> |
|  | <li> Hulk </li> |
|  | <li> Thor </li> |
|  | </ol> |
|  |  |
|  | <h2> Unordered List: </h2> |
|  |  |
|  | <h3> Friend's List: </h3> |
|  | <ul> |
|  | <li> Rizwan </li> |
|  | <li> Ajay </li> |
|  | <li> Srinivas </li> |
|  | <li> Yesu Raju </li> |
|  | <li> Mounav </li> |
|  | <li> Mounav </li> |
|  | </ul> |
|  |  |
|  | <h3> Coding List: </h3> |
|  | <ul type="circle"> |
|  | <li> C language </li> |
|  | <li> Python </li> |
|  | <li> C++ Language </li> |
|  | <li> HTML Language </li> |
|  | <li> Dennis Ritchie </li> |
|  | </ul> |
|  |  |
|  | <h3> Bike's List: </h3> |
|  | <ul type="square"> |
|  | <li> Hero Honda </li> |
|  | <li> Bullet </li> |
|  | <li> Avenger </li> |
|  | <li> Jupyter </li> |
|  | <li> TVS Excel Super100 </li> |
|  | <li> Royal Enfield </li> |
|  | </ul> |
|  |  |
|  | <h3> Avenger's List: </h3> |
|  | <ul type="none"> |
|  | <li> Spider-Man </li> |
|  | <li> Captain America </li> |
|  | <li> Iron-Man </li> |
|  | <li> Hawkye </li> |
|  | <li> Hulk </li> |
|  | <li> Thor </li> |
|  | </ul> |
|  | </body> |
|  | </html> |

**OUTPUT:**









# Week – 4

Aim: Create an your Resume by using framset tag and using HTML language.

Description:

## Frameset:

The <frameset> tag in HTML is used to define the frameset. The

<frameset> element contains one or more frame elements. It is used to specify the number of rows and columns in frameset with their pixel of spaces. Each element can hold a separate document.

HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized: into rows and columns.

## 

## Attributes in Frameset:

**cols**: The cols attribute is used to create vertical frames in a web browser. This attribute is basically used to define the no. of columns and their size inside the frameset tag.

**rows:** The rows attribute is used to create horizontal frames in the web browser. This attribute is used to define the no. of rows and their size inside the frameset tag.

**border:** This attribute of frameset tag defines the width of the border of each frame in pixels. Zero value is used for no border.

**frameborder:** This attribute of frameset tag is used to specify whether a three-dimensional border should be displayed between the frames or not for this use two values 0 and 1, where 0 defines no border and value 1 signifies for yes there will be a border.

**framespacing:** This attribute of frameset tag is used to specify the amount of spacing between the frames in a frameset. This can take any integer value as a parameter which basically denotes the value in pixel.

**Syantax For Column’s Division:**

<frameset cols = “n1%, n2%, n3%>

<frame name = “top” src = “first file.html” />

<frame name = “main” src = “second file.html” />

<frame name = “bottom” src = “third file.html” />

.

.

.

.

<noframes>

<body>

You can write here documentation.

n1,n2,n3 are the division of web-page percentages.

</body>

</noframes>

</frameset>

**Syantax For Row’s Division:**

<frameset rows = “n1%, n2%, n3%>

<frame name = “top” src = “first file.html” />

<frame name = “main” src = “second file.html” />

<frame name = “bottom” src = “third file.html” />

.

.

.

.

<noframes>

<body>

You can write here documentation.

n1,n2,n3 are the division of web-page percentages.

</body>

</noframes>

</frameset>

**Program:**

|  |
| --- |
|  |
|  | <html>  <head> |
|  | <title> RESUME </title> |
|  | </head> |
|  |  |
|  |  |
|  | <frameset cols = "25%, 39%, 36%"> |
|  | <frame name = "top" src = "<https://mjnvsai.github.io/VRSEC/Web-programming/HTML/frameset/sai1.html>" /> |
|  | <frame name = "main" src = "<https://mjnvsai.github.io/VRSEC/Web-programming/HTML/frameset/paragraph.html>" /> |
|  | <frame name = "bottom" src = "<https://mjnvsai.github.io/VRSEC/Web-programming/HTML/frameset/personal.html>" /> |
|  | <noframes> |
|  | <body> this browser is working </body> |
|  | </noframes> |
|  | </frameset> |
|  | </html> |

**OUTPUT:**



# WEEK – 5

Aim: To create an Academic Qualifications Table using Table tag in HTMl

## Description:

## Tables:

* HTML tables allow web developers to arrange data into rows and columns.
* The HTML tables are created using the **<table>** tag in which the **<tr>** tag is used to create table rows and **<td>** tag is used to create data cells. The elements under <td> are regular and left aligned by default

**Table Heading:**

* Table heading can be defined using **<th>** tag. This tag will be put to replace <td> tag, which is used to represent actual data cell. Normally you will put your top row as table heading as shown below, otherwise you can use <th> element in any row. Headings, which are defined in <th> tag are centered and bold by default.

**Cellpadding and Cellspacing:**

* There are two attributes called cellpadding and cellspacing which you will use to adjust the white space in your table cells. The cellspacing attribute defines space between table cells, while cellpadding represents the distance between cell borders and the content within a cell.

**Table Height and Width:**

* You can set a table width and height using width and height attributes. You can specify table width or height in terms of pixels or in terms of percentage of available screen area.

**Table Backgrounds:**

* You can set table background using one of the following two ways −
* bgcolor attribute − You can set background color for whole table or just for one cell.
* background attribute − You can set background image for whole table or just for one cell.
* You can also set border color also using bordercolor attribute

Table Header: use <th> ……………………………… </th>

Table rows or colums: use <tr> ………………………… </tr>

Table Data: use <tc> …………………………………….. </tc>

## Attributes in Table Tag:

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<table>](https://www.w3schools.com/tags/tag_table.asp) | Defines a table |
| [<th>](https://www.w3schools.com/tags/tag_th.asp) | Defines a header cell in a table |
| [<tr>](https://www.w3schools.com/tags/tag_tr.asp) | Defines a row in a table |
| [<td>](https://www.w3schools.com/tags/tag_td.asp) | Defines a cell in a table |
| [<caption>](https://www.w3schools.com/tags/tag_caption.asp) | Defines a table caption |
| [<colgroup>](https://www.w3schools.com/tags/tag_colgroup.asp) | Specifies a group of one or more columns in a table for formatting |
| [<col>](https://www.w3schools.com/tags/tag_col.asp) | Specifies column properties for each column within a <colgroup> element |
| [<thead>](https://www.w3schools.com/tags/tag_thead.asp) | Groups the header content in a table |
| [<tbody>](https://www.w3schools.com/tags/tag_tbody.asp) | Groups the body content in a table |
| [<tfoot>](https://www.w3schools.com/tags/tag_tfoot.asp) | Groups the footer content in a table |

**Syntax For Table:**

<table border = “ ” width = “ ” height = “ ” cellspacing = “ ” bordercolor = “ ” bgcolor = “ ” >

<tr>

<th> ……………………………………….. </th>

.

.

.

</tr>

<tr>

<td> ………………………………………… </td>

<td> ……………………………………….. </td>

.

.

.

.

</tr>

.

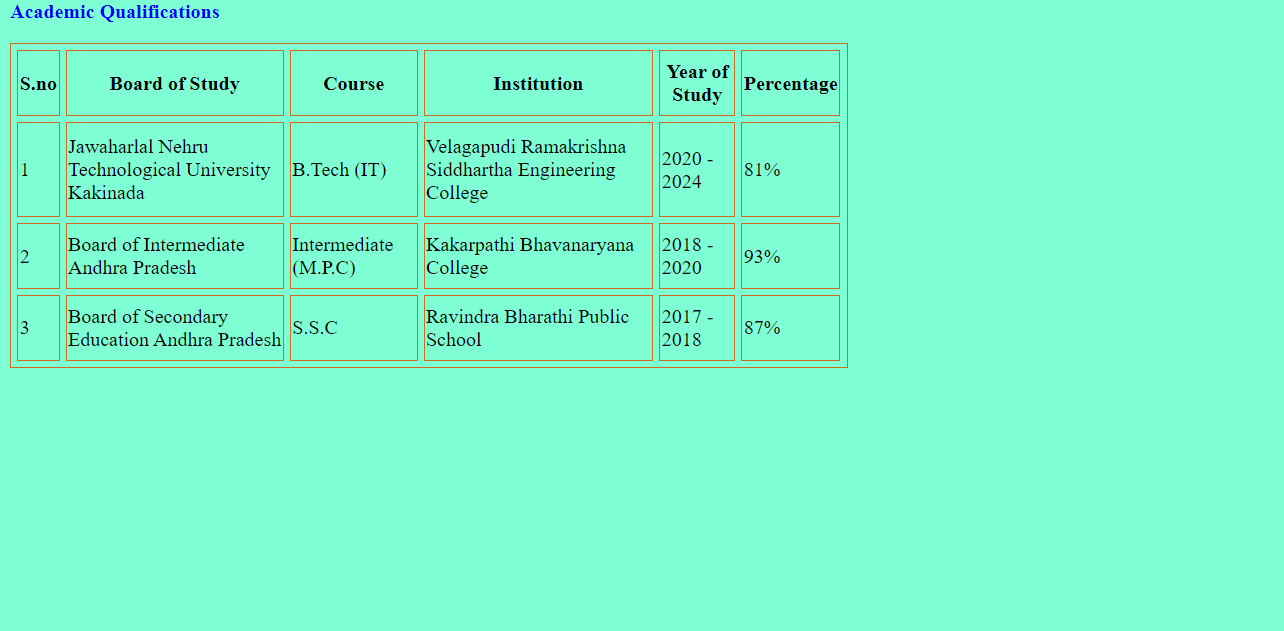
.

</table>

**Program:**

|  |  |
| --- | --- |
|  | |
|  | <html>  <body bgcolor="Aquamarine"> | |
|  | <p style="color:blue"> <b> Academic Qualifications </b> </p> | |
|  |  | |
|  | <table border = "1" width="670" height="260" cellspacing = "5" bordercolor = "Chocolate"> | |
|  | <tr> | |
|  | <th> S.no </th> | |
|  |  | |
|  | <th> Board of Study </th> | |
|  | <th> Course </th> | |
|  | <th> Institution </th> | |
|  | <th> Year of Study </th> | |
|  | <th> Percentage </th> | |
|  | </tr> | |
|  |  | |
|  | <tr> | |
|  | <td> 1 </td> | |
|  | <td> Jawaharlal Nehru Technological University Kakinada</td> | |
|  | <td> B.Tech (IT) </td> | |
|  | <td> Velagapudi Ramakrishna Siddhartha Engineering College</td> | |
|  | <td> 2020 - 2024 </td> | |
|  | <td> 81% </td> | |
|  | </tr> | |
|  |  | |
|  | <tr> | |
|  | <td> 2 </td> | |
|  | <td> Board of Intermediate Andhra Pradesh </td> | |
|  | <td> Intermediate (M.P.C) </td> | |
|  | <td> Kakarpathi Bhavanaryana College </td> | |
|  | <td> 2018 - 2020 </td> | |
|  | <td> 93% </td> | |
|  | </tr> | |
|  |  | |
|  | <tr> | |
|  | <td> 3 </td> | |
|  | <td> Board of Secondary Education Andhra Pradesh </td> | |
|  | <td> S.S.C</td> | |
|  | <td> Ravindra Bharathi Public School </td> | |
|  | <td> 2017 - 2018 </td> | |
|  | <td> 87% </td> | |
|  | </tr> | |
|  | </table> | |
|  |  | |
|  | </body> | |
|  | </html> | |

**OUTPUT:**



# WEEK – 6

**Aim** : To create simple Resume HTML form

**Description**:

HTML Forms are required, when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc.

A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like text fields, textarea

fields, drop-down menus, radio buttons, checkboxes, etc.

**Form – Tag Attributes:**

|  |  |
| --- | --- |
| **Sr.No** | **Attribute & Description** |
| 1 | **action**  Backend script ready to process your passed data. |
| 2 | **method**  Method to be used to upload data. The most frequently used are GET and POST methods. |
| 3 | **target**  Specify the target window or frame where the result of the script will be displayed. It takes values like \_blank, \_self, \_parent etc. |
| 4 | **enctype**  You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are −  **application/x-www-form-urlencoded** − This is the standard method most forms use in simple scenarios.  **mutlipart/form-data** − This is used when you want to upload binary data in the form of files like image, word file etc. |

**HTML Form Controls:**

There are different types of form controls that you can use to collect data using HTML form −

* Text Input Controls
* Checkboxes Controls
* Radio Box Controls
* Select Box Controls
* File Select boxes
* Hidden Controls
* Clickable Buttons
* Submit and Reset Button
* Text Input Controls

There are three types of text input used on forms −

* **Single-line text input** controls : This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.
* **Password input controls** : This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTMl <input> tag.
* **Multi-line text input controls** : This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.
* **Checkbox Control** : Checkboxes are used when more than one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to checkbox..
* **Radio Button Control** : Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to radio.
* **Select Box Control** : A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.
* **File Upload Box** : If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the <input> element but type attribute is set to file.
* **Button Controls** : There are various ways in HTML to create clickable buttons. You can also create a clickable button using <input>tag by setting its type attribute to button.
* **Hidden Form Controls** : Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page. For example, following hidden form is being used to keep current page number. When a user will click next page then the value of hidden control will be sent to the web server and there it will decide which page will be displayed next based on the passed current page.

**Type & Attributes:**

|  |  |
| --- | --- |
| <input type="text"> | Displays a single-line text input field |
| <input type="radio"> | Displays a radio button (for selecting one of many choices) |
| <input type="checkbox"> | Displays a checkbox (for selecting zero or more of many choices) |
| <input type="submit"> | Displays a submit button (for submitting the form) |
| <input type="button"> | Displays a clickable button |
| <input type=”password”> | Encodes the text into symbols |
| <input type=”reset”> | It will Resets the Form |
| <input type=”image”> | It will create an image button |
| <select> </select> | It will create drop down box |
| <option> <option> | It is a part of select tag |
| <input type=”file”> | User can upload the files |

**Syntax:**

<form action = “ …….. “ enctype = “……” method = “POST” >

<label for = “……”> ………………….. </label>

<input type = “….” Id = “…” name = “……” placeholder = “…….” >

<select>

<option> …………………………………….. </option>

.

.

.

</select>

<textarea name = “….” Rows = “….” Cols = “….” Placeholder = “…….” > </textarea>

</form>

**Program:**

<html>

<head>

<title> Resume </title>

</head>

<body bgcolor = "aquamarine">

<h1 style="color:blue"> <strong> <i> <U> <center> RESUME </center> </U> </i> </strong> </h1>

<form action="https://formsubmit.co/cb2b7f7afbc855bb5ec6f9f81cec6345" enctype="multipart/form-data" method="POST" >

<label for="fname" > First Name: </label>

<input type="text" id="fname" name="fname" placeholder="Enter your First name" required> <br> <br>

<label for="mname" > Middle Name: </label>

<input type="text" id="mname" name="middle name" placeholder="Enter your Middle name" required> <br> <br>

<label for="lname" > Last Name: </label>

<input type="text" id="lname" name="last name" placeholder="Enter your Last name" required> <br> <br>

<label for="mail" > Email: </label>

<input type="mail" id="mail" name="email-id" placeholder="abcde@\*\*\*" required> <br> <br>

<label for="fb" > Facebook: </label>

<input type="url" id="fb" name="facebook" placeholder="Enter your facebook link" required> <br> <br>

<label for="insta" > Instagram: </label>

<input type="url" id="insta" name="instagram" placeholder="Enter your instagram link" required> <br> <br>

<label for="hack" > Hackerank: </label>

<input type="url" id="hack" name="hackerrank" placeholder="Enter your hackerank link" required> <br> <br>

<label for="git" > Github: </label>

<input type="url" id="git" name="Github" placeholder="Enter your Github link" required> <br> <br>

<label for="descrip" > About Yourself: </label> <br>

<textarea name="message" rows="10" cols="30"> </textarea> <br> <br>

<label for="vresume" > Upload Video-resume: </label>

<input type="file" id="vresume" name="video-resume" required > <br> <br>

<label for="aresume" > Upload Audio-resume: </label>

<input type="file" id="aresume" name="Audio-resume" required > <br> <br>

<h2 style="color:blue"> Academic Qualifications </h2>

<table border = "1" width="670" height="260" cellspacing = "5" bordercolor = "Chocolate">

<tr>

<th> S.no </th>

<th> Board of Study </th>

<th> Course </th>

<th> Institution </th>

<th> Year of Study </th>

<th> Percentage </th>

</tr>

<tr>

<td> 1 </td>

<td> <input type="text" size="30" required> </td>

<td> <input type="text" size="15" required> </td>

<td> <input type="text" size="30" required> </td>

<td> <input type="text" size="10" required> </td>

<td> <input type="text" size="10" required> </td>

</tr>

<tr>

<td> 2 </td>

<td> <input type="text" size="30" required> </td>

<td> <input type="text" size="15" required> </td>

<td> <input type="text" size="30" required> </td>

<td> <input type="text" size="10" required> </td>

<td> <input type="text" size="10" required> </td>

</tr>

<tr>

<td> 3 </td>

<td> <input type="text" size="30" required> </td>

<td> <input type="text" size="15" required> </td>

<td> <input type="text" size="30" required> </td>

<td> <input type="text" size="10" required> </td>

<td> <input type="text" size="10" required> </td>

</tr>

</table>

<br>

<br>

<label for="dept" style="color:blue" > <b> Department: </b> </label> <br> <br>

<select>

<option> NONE </option>

<option> Artificial - Intelligence (AI) </option>

<option> Computer - Science (CSE) </option>

<option> Information - Technology (IT) </option>

<option> Electronics and Instrumentation (EIE) </option>

<option> Electrical and Electronics (EEE) </option>

<option> Electronics and Communication (ECE) </option>

<option> Mechanical Engineering (ME) </option>

<option> Civil Engineering (CE) </option>

</select>

<br>

<br>

<label for="ts" style="color:blue" > <b> Technical - Skills: </b> </label> <br> <br>

<input type="checkbox" id="technical-skills" C-language> C-language

<input type="checkbox" Python with OOPS> Python with OOPS

<input type="checkbox" OOPS with C++> OOPS with C++

<input type="checkbox" Web-programming> Web-programming

<input type="checkbox" Data-structures> Data-structures

<br>

<br>

<label for="nts" style="color:blue" > <b> Non-Technical - Skills: </b> </label> <br> <br>

<input type="checkbox" id="non-tecnical-skills" Singing> Singing

<input type="checkbox" Dancing> Dancing

<input type="checkbox" Playing> Playing

<input type="checkbox" Watching movies & series> Watching movies & series

<input type="checkbox" Reading Books> Reading Books

<input type="checkbox" Editing photos & videos> Editing photos & videos

<h2 style="color:blue"> Communication: </h2>

<label for="father" > Father's Name: </label>

<input type="text" id="father" size="25" placeholder="Enter your father's name" required> <br> <br>

<label for="foccup" > Father's Occupation: </label>

<input type="text" id="foccup" size="25" placeholder="Enter your fathers job name" required> <br> <br>

<label for="dob" > Date of Birth: </label>

<input type="date" id="dob" required> <br> <br>

<label for="gender" > Gender: </label>

<input type="radio" name = "Gender" required > Male

<input type="radio" name = "Gender" required > Female

<input type="radio" name = "Gender" required > Others

<br>

<br>

<label for="ph.no" > Phone-number: </label>

<select>

<option> +91 </option>

<option> +672 </option>

<option> +61 </option>

<option> +880 </option>

<option> +1 </option>

<option> +33 </option>

<option> +49 </option>

<option> +852 </option>

<option> +39 </option>

<option> +81 </option>

</select>

<input type="phone" size=20 name="phone number" placeholder="8331\*\*\*\*\*\*" required > <br> <br>

<label for="addres" > <b> Address: </b> </label> <br>

<textarea name="message" rows="10" cols="30" placeholder="Write your permanent address" required> </textarea> <br> <br>

<br>

<label for="place" > <b> Place: </b> </label>

<input type="text" id="place" placeholder="Enter your place" required> <br> <br>

<label for="pdate" > <b> Date: </b> </label>

<input type="date" id="pdate" required> <br> <br>

<h3 align="right" > SIGNATURE <label for="sign"</label> <br> <input type="text" id="sign" required> <br> </h3>

<br>

<input type="hidden" name="\_template" value="table">

<br>

<input type="hidden" name="\_autoresponse" value="Successfully submitted the form">

<br>

<input type="submit" value = "Submit" name = "submit button" >

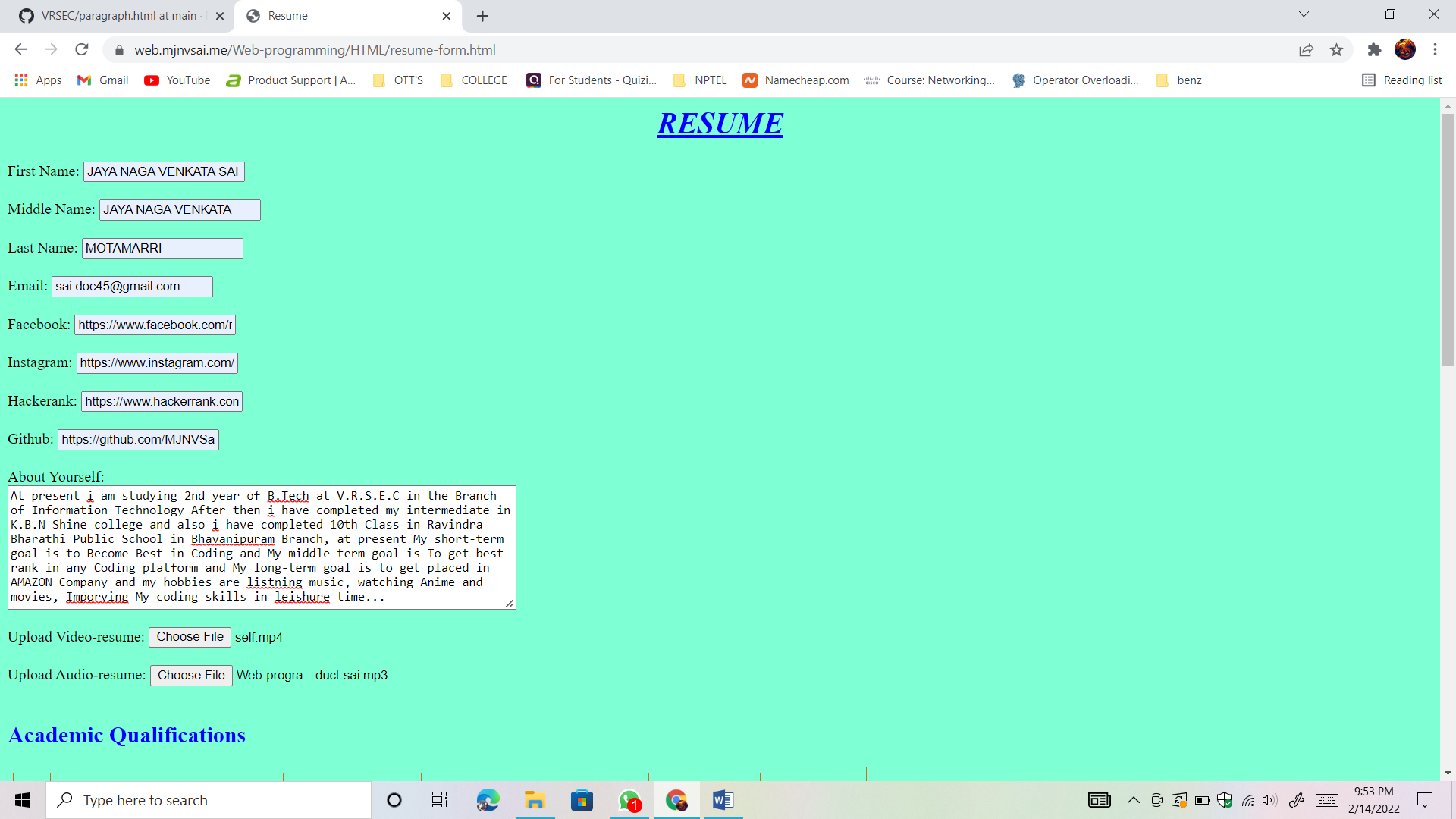
<input type="reset" value = "Reset" name = "reset button" >

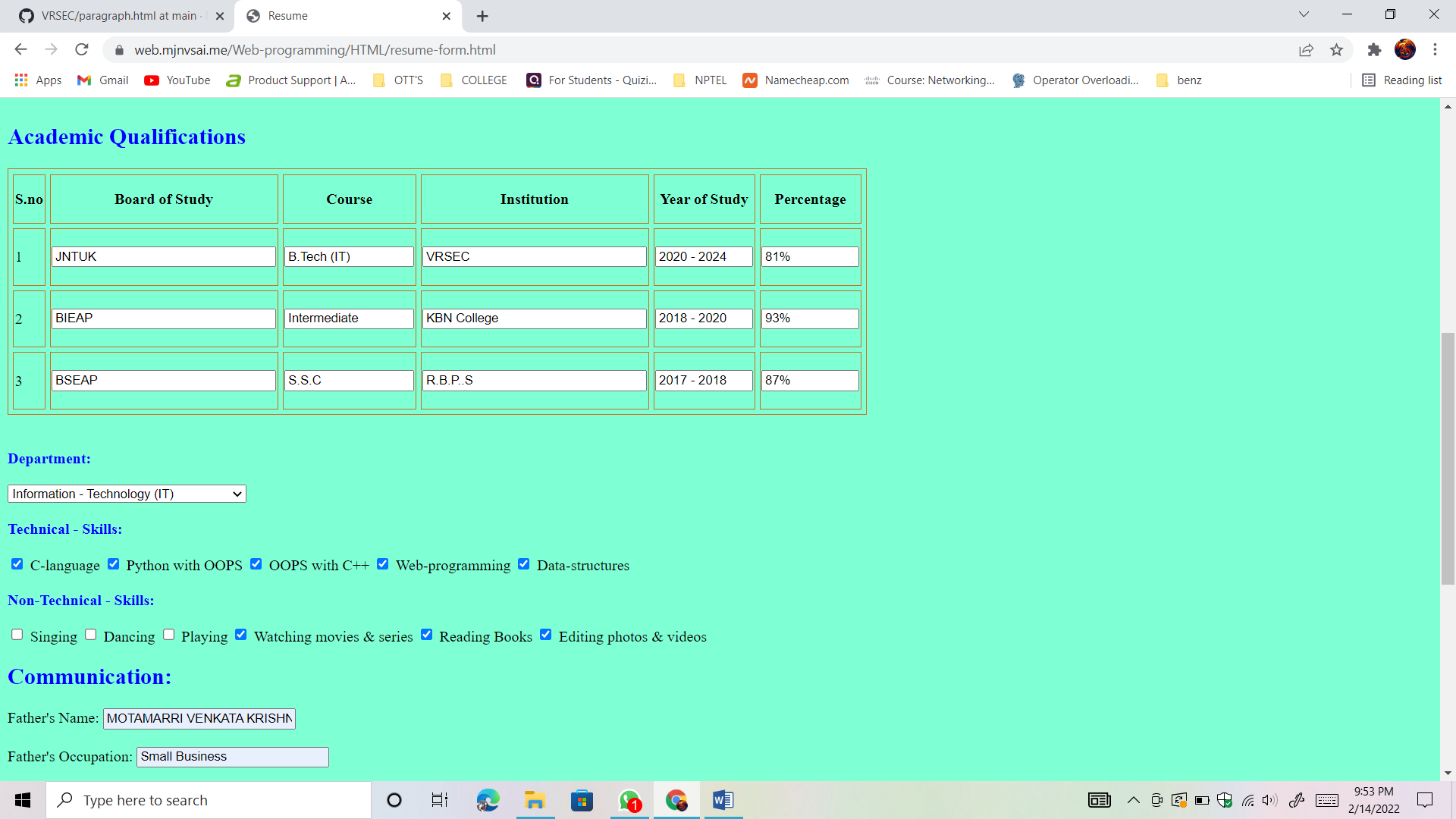
</form>

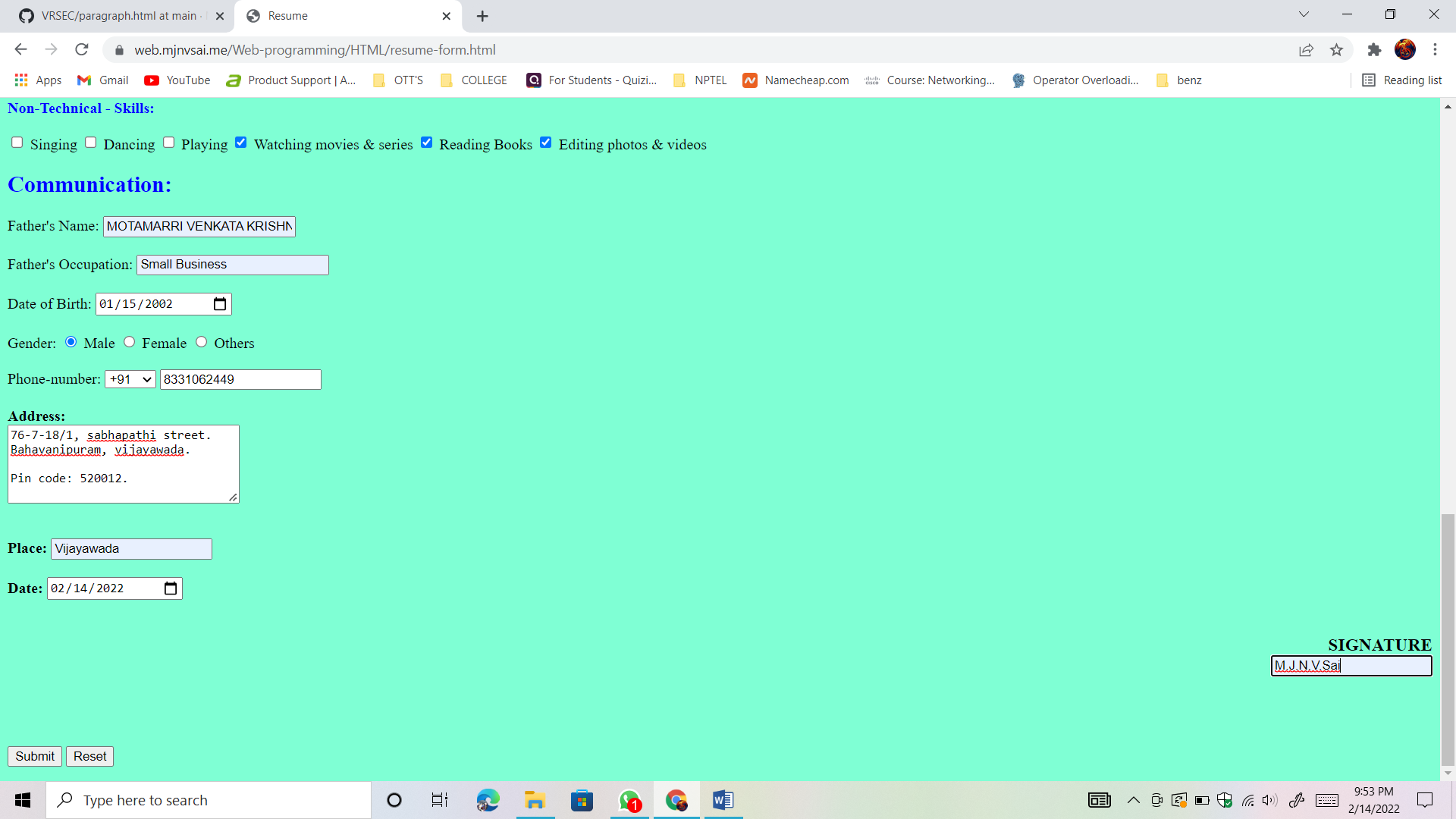
</body>

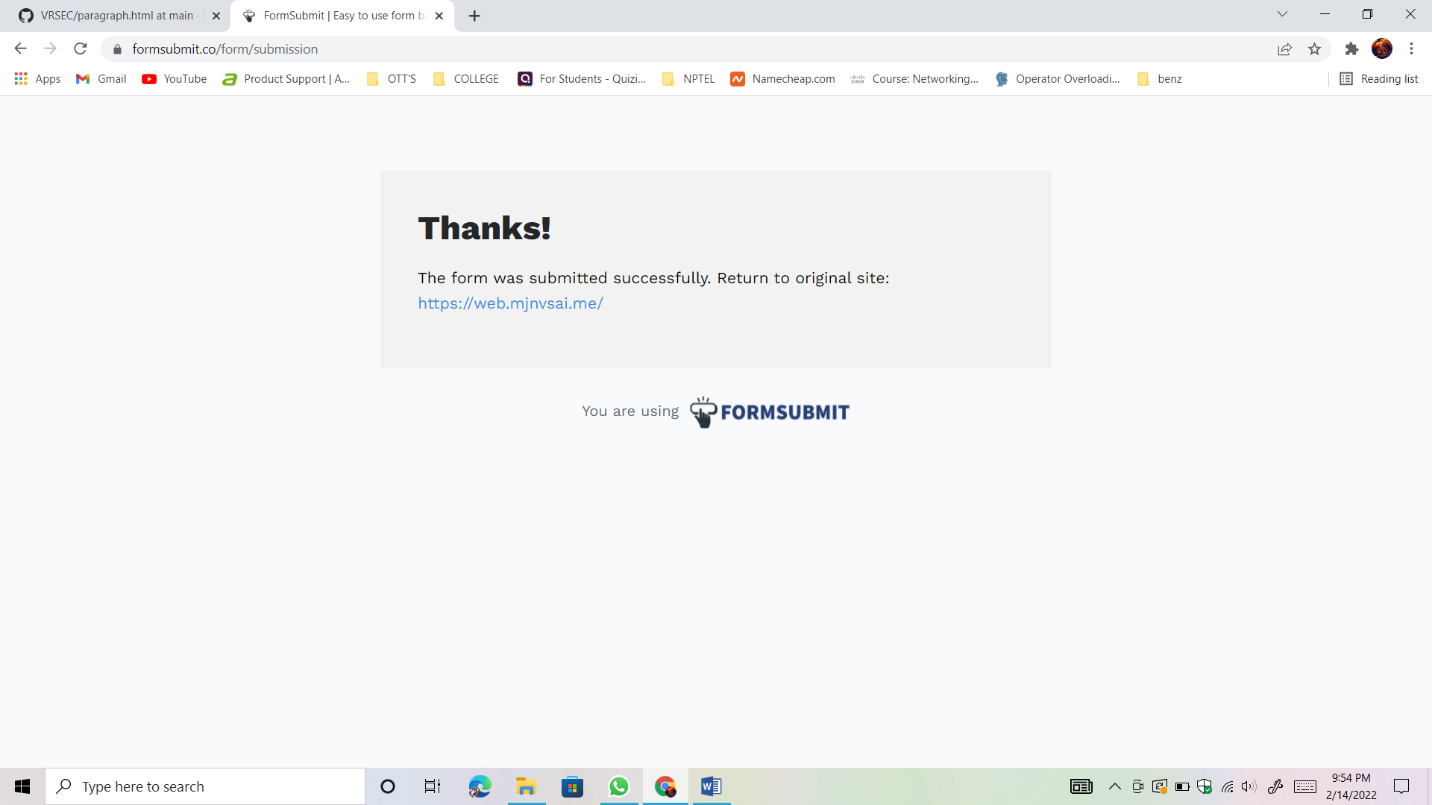
</html>

**OUTPUT:**

****







Result : Successfully Executed the program..

# WEEK – 7

**Aim** : Introduction to Django and Django installation

**Description:**

1) what is django how it is useful?

A) 1. Django is a high-level Python web framework that enables rapid development of secure and maintainable websites.

2. Django helps developers avoid many common security mistakes by providing a framework that has been engineered to

"do the right things" to protect the website automatically.

1. Django is an open-source framework for backend web applications based on Python — one of the top web development languages.Its main goals are simplicity, flexibility, reliability, and scalability.

2) what is the framework used in Django?

A) It is a Python-based free and open-source web framework that follows the model–template–views (MTV) architectural pattern.

It is maintained by the Django Software Foundation (DSF).

3) How to install Django?

A) step-1 => install latest Version the python.

step-2 => open command prompt in system and run the below command : python -m pip install Django

step-3 => Make sure the your is pip is upgraded in python.

step-4 => after running the command you will get the message "Successfully installed the package".

4) what is the Data-base and which data-base is used in Django?

A) 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).

Most of the databases use structured query language (SQL) for writing and querying data.

==> Django officially supports the following databases: PostgreSQL. MariaDB. MySQL.

5) How to create an environment to create an project in Django?

A) Steps to Create an Environment in Python:

step-1 => Open a terminal.

step-2 => Setup the pip package manager.

step-3 => Install the virtualenv package.

step-4 => Create the virtual environment.

step-5 => Activate the virtual environment.

Create A New Django Project

1. Change to the Django projects folder and run the following command to create a new project.

Replace the PROJECT placeholder with the name of your project in this and all subsequent commands.

**{ cd installdir/apps/django/django\_projects/PROJECT**

**django-admin.py startproject PROJECT}.**

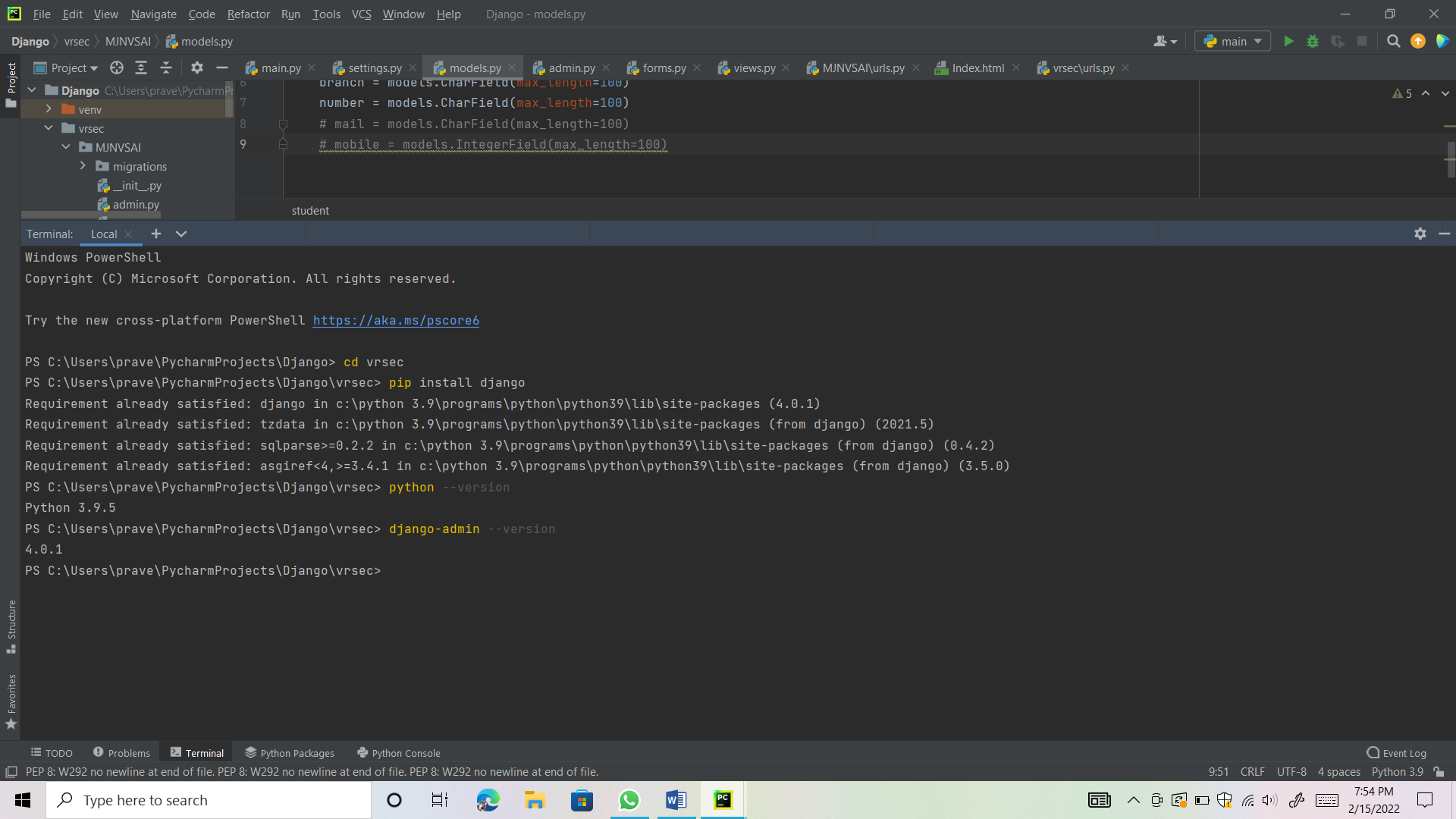
2. If you wish to create a new application inside the project, execute the following commands as well.

Replace the APP placeholder with the name of your application in this and all subsequent commands.

**{ cd PROJECT**

**python3 manage.py startapp APP}**

**OUTPUT:**



Result: Sucessfully installed Django

# WEEK – 8

**Aim** : Django install and make it work in our systems , eg ;Creating a

form**Program** :

**settings.py**

"""

Django settings for vrsec project.

Generated by 'django-admin startproject' using Django 4.0.1.

For more information on this file, see

https://docs.djangoproject.com/en/4.0/topics/settings/

For the full list of settings and their values, see

https://docs.djangoproject.com/en/4.0/ref/settings/

"""

from pathlib import Path

# Build paths inside the project like this: BASE\_DIR / 'subdir'.

BASE\_DIR = Path(\_file\_).resolve().parent.parent

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/4.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = 'django-insecure-r1-ys-et!zy!)!@g(5u^ghk)s+@%8h$tgg=n-3@tt$+(+0$-&e'

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED\_HOSTS = []

# Application definition

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

'MJNVSAI',

]

MIDDLEWARE = [

'django.middleware.security.SecurityMiddleware',

'django.contrib.sessions.middleware.SessionMiddleware',

'django.middleware.common.CommonMiddleware',

'django.middleware.csrf.CsrfViewMiddleware',

'django.contrib.auth.middleware.AuthenticationMiddleware',

'django.contrib.messages.middleware.MessageMiddleware',

'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'vrsec.urls'

TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates',

'DIRS': [BASE\_DIR/"templates"],

'APP\_DIRS': True,

'OPTIONS': {

'context\_processors': [

'django.template.context\_processors.debug',

'django.template.context\_processors.request',

'django.contrib.auth.context\_processors.auth',

'django.contrib.messages.context\_processors.messages',

],

},

},

]

WSGI\_APPLICATION = 'vrsec.wsgi.application'

# Database

# https://docs.djangoproject.com/en/4.0/ref/settings/#databases

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.sqlite3',

'NAME': BASE\_DIR / 'db.sqlite3',

}

}

# Password validation

# https://docs.djangoproject.com/en/4.0/ref/settings/#auth-password-validators

AUTH\_PASSWORD\_VALIDATORS = [

{

'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

},

]

# Internationalization

# https://docs.djangoproject.com/en/4.0/topics/i18n/

LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/4.0/howto/static-files/

STATIC\_URL = 'static/'

# Default primary key field type

# https://docs.djangoproject.com/en/4.0/ref/settings/#default-auto-field

DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

**models.py :**

from django.db import models

# Create your models here.

class student(models.Model):

name = models.CharField(max\_length=100)

branch = models.CharField(max\_length=100)

number = models.CharField(max\_length=100)

mail = models.CharField(max\_length=100)

mobile = models.IntegerField(max\_length=100)

**admin.py :**

from django.contrib import admin

from MJNVSAI.models import student

# Register your models here.

admin.site.register(student)

**forms.py :**

from django import forms

from MJNVSAI.models import student

class Login(forms.ModelForm):

class Meta:

model = student

fields = '\_\_all\_\_'

**views.py :**

from django.shortcuts import render

from django.http import HttpResponse

from MJNVSAI.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})

**MJNVSAI\urls.py :**

from django.urls import path

from . import views

urlpatterns = [path('form', views.form\_view, name='form\_view')]

**templates\Index.html :**

<!DOCTYPE html>

<html>

<body>

<form method="POST" >

{%csrf\_token %}

{{objects.as\_p}}

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

</form>

</body>

</html>

**vrsec\urls.py :**

"""vrsec 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 MJNVSAI import views

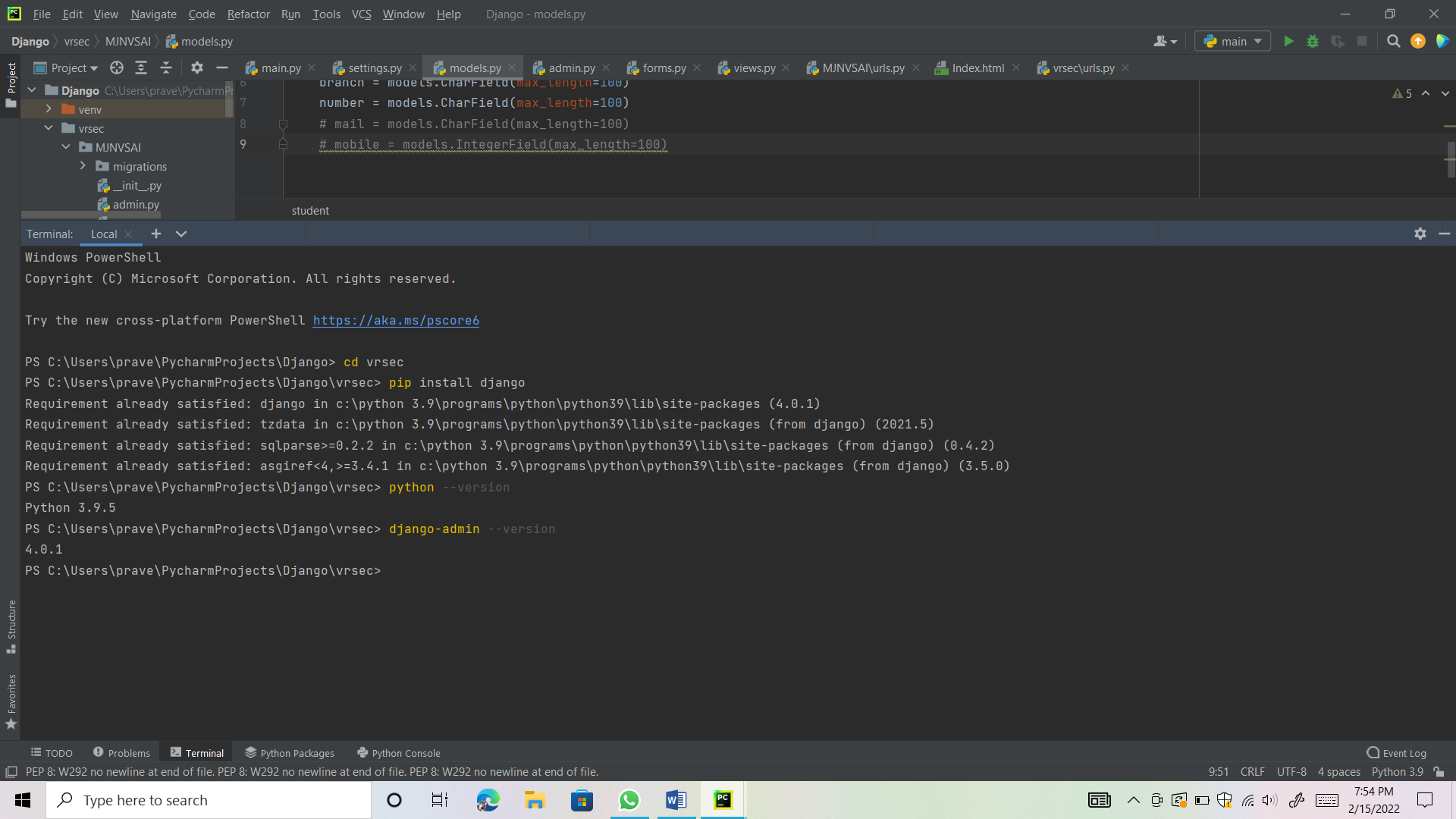
urlpatterns = [

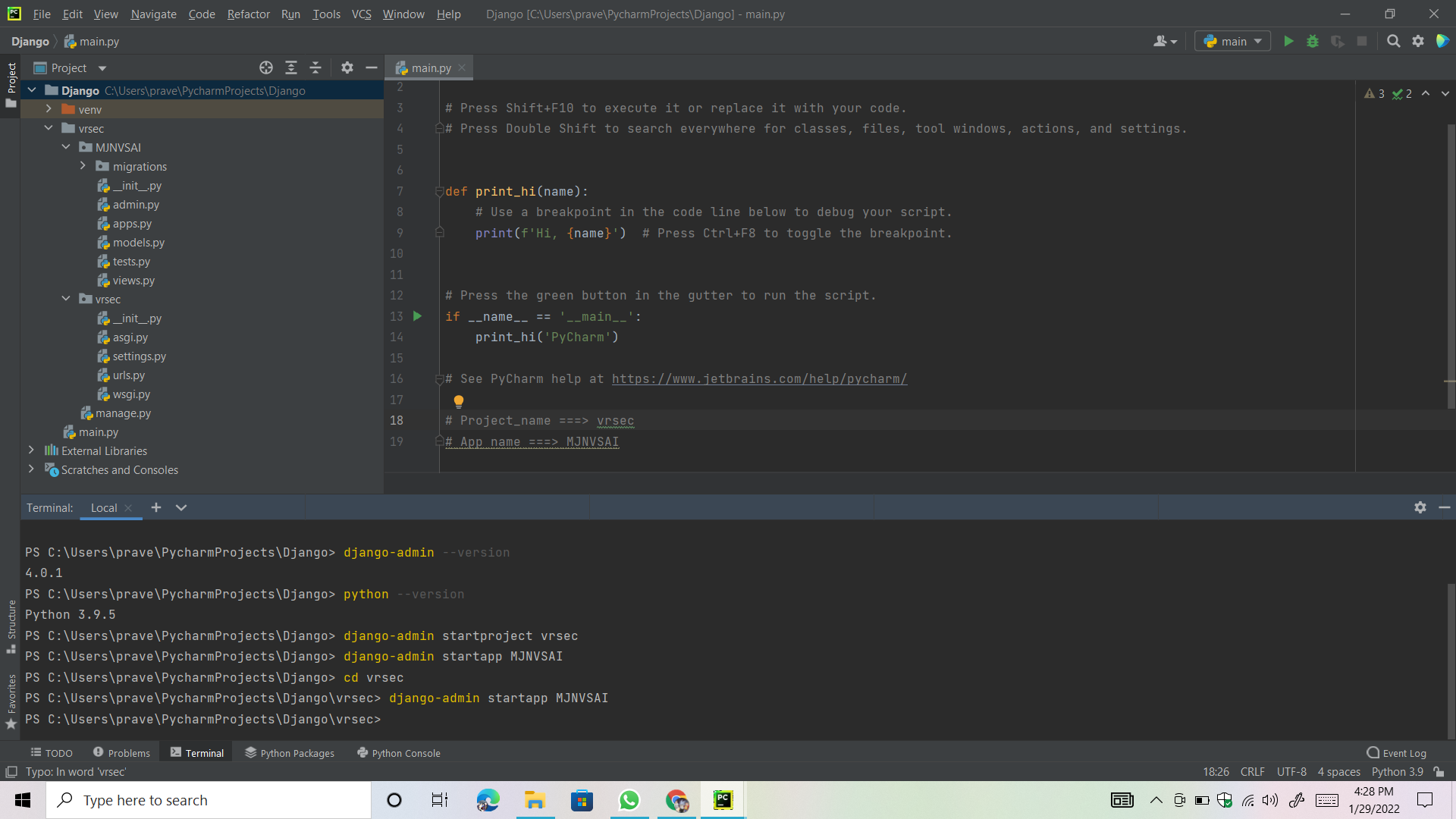
path('admin/', admin.site.urls),

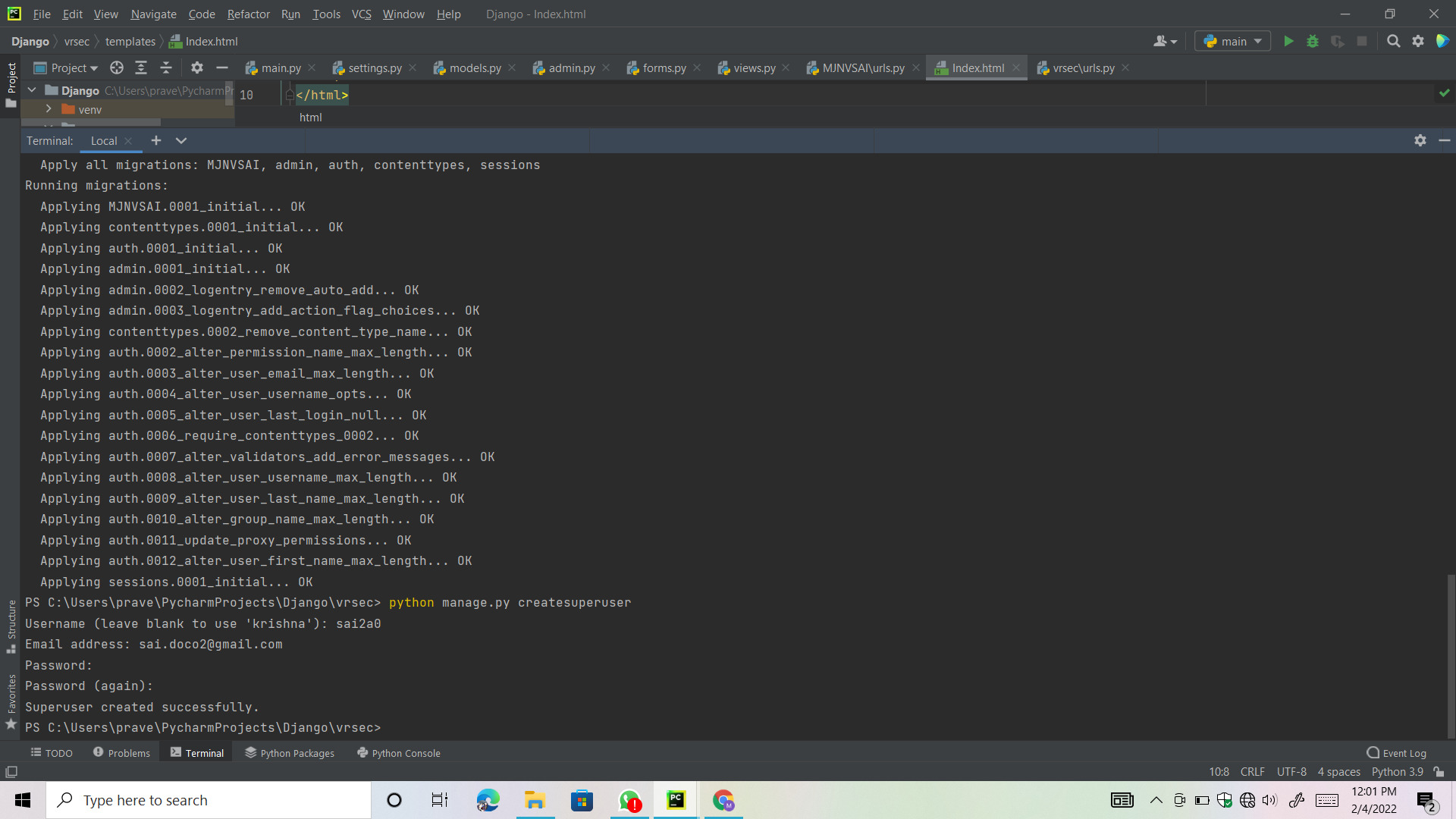
path(r'',include('MJNVSAI.urls'))

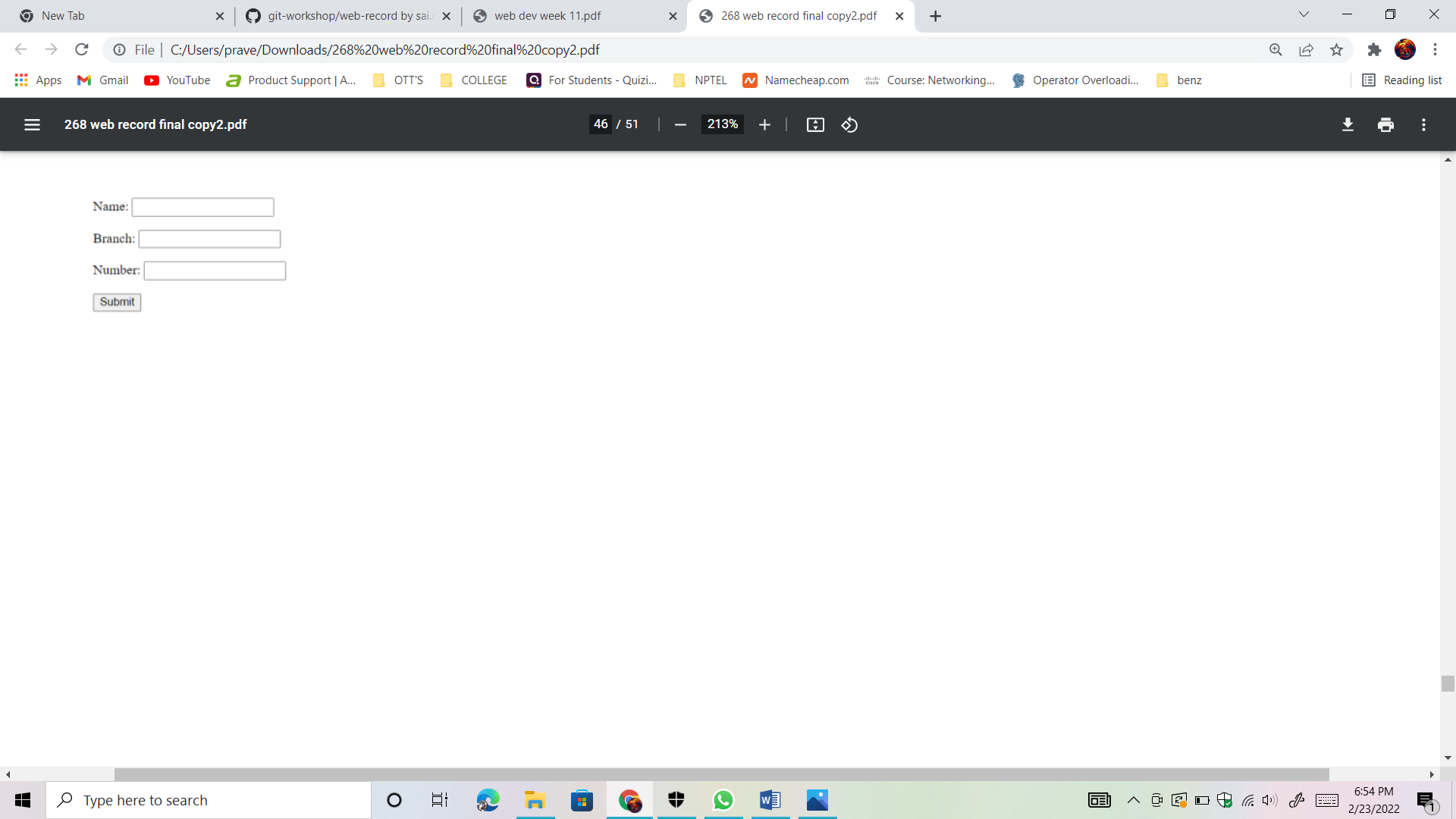
]

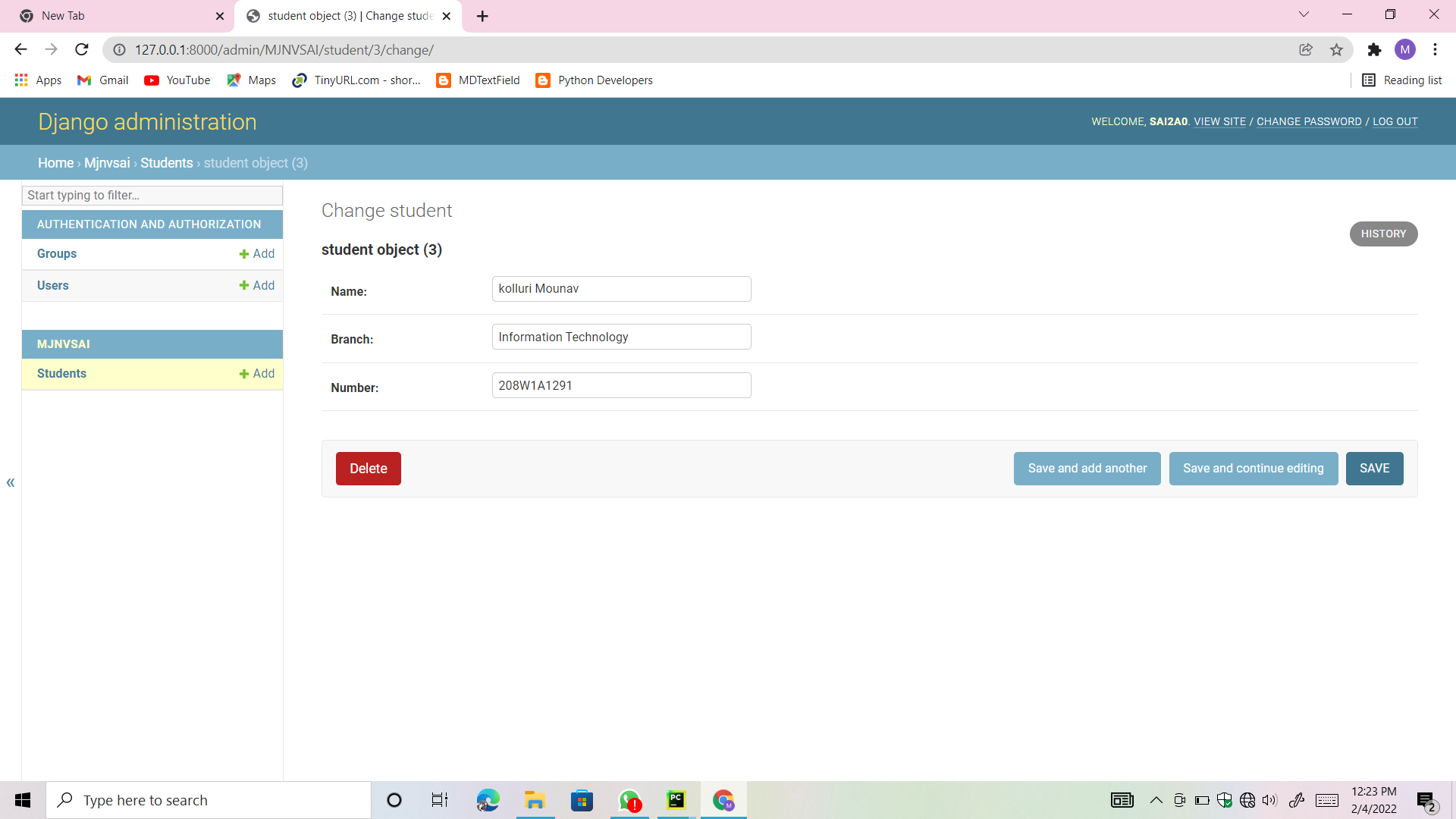
**OUTPUT :**

****









# WEEK – 9

**Aim :** Apply CRUD operations in Django form

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.

C === > CREATE ===> INSERT

R ===> READ ===> SELECT

U ===> UPDATE ===> EDIT

D ===> DELETE ===> DELETE

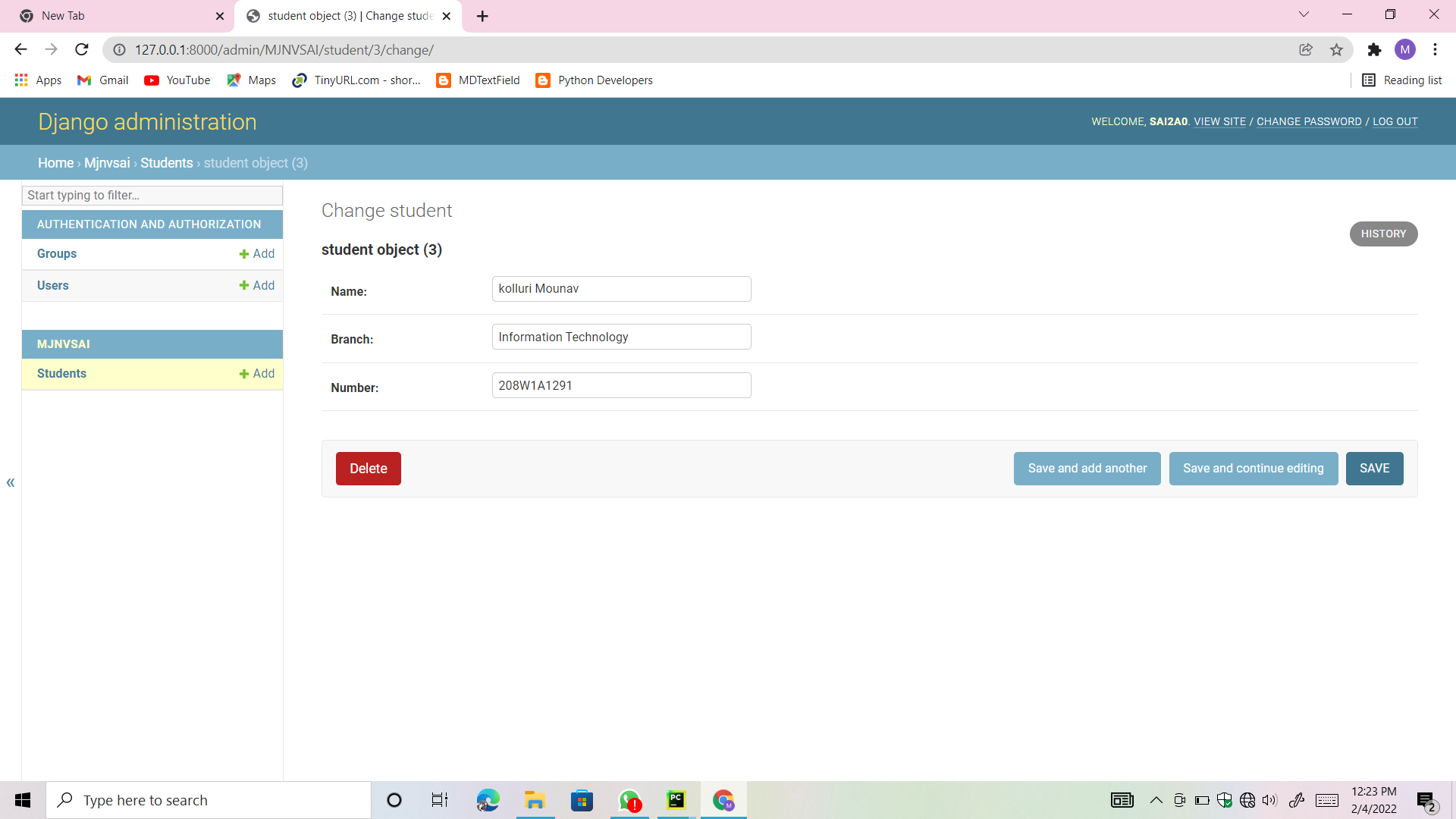
READ :



Update :

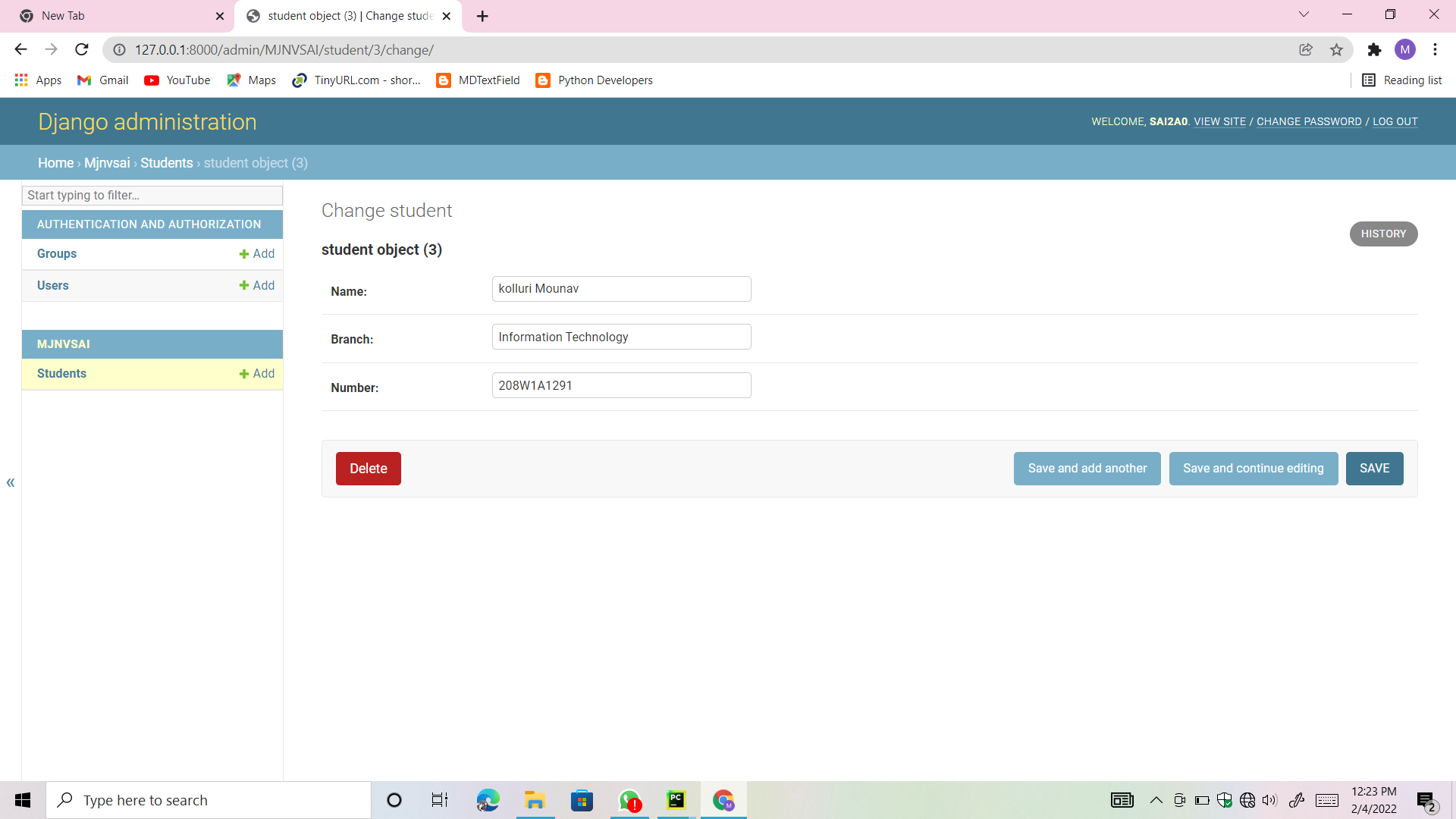
Step 1: Select and open one of the objects created.

(say student object(1))

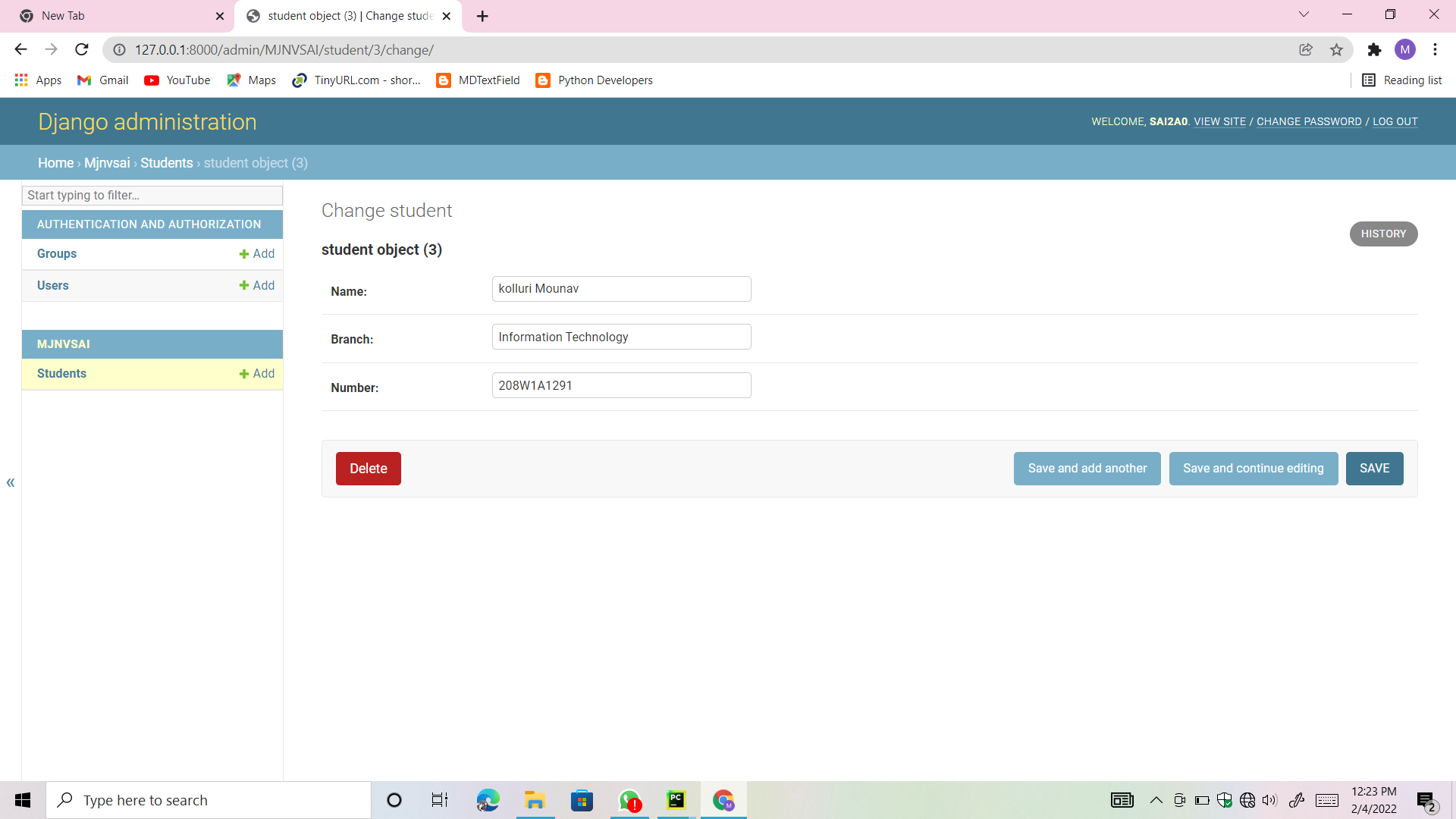


Step 2: Select the Save and continue editing button to update the

object.

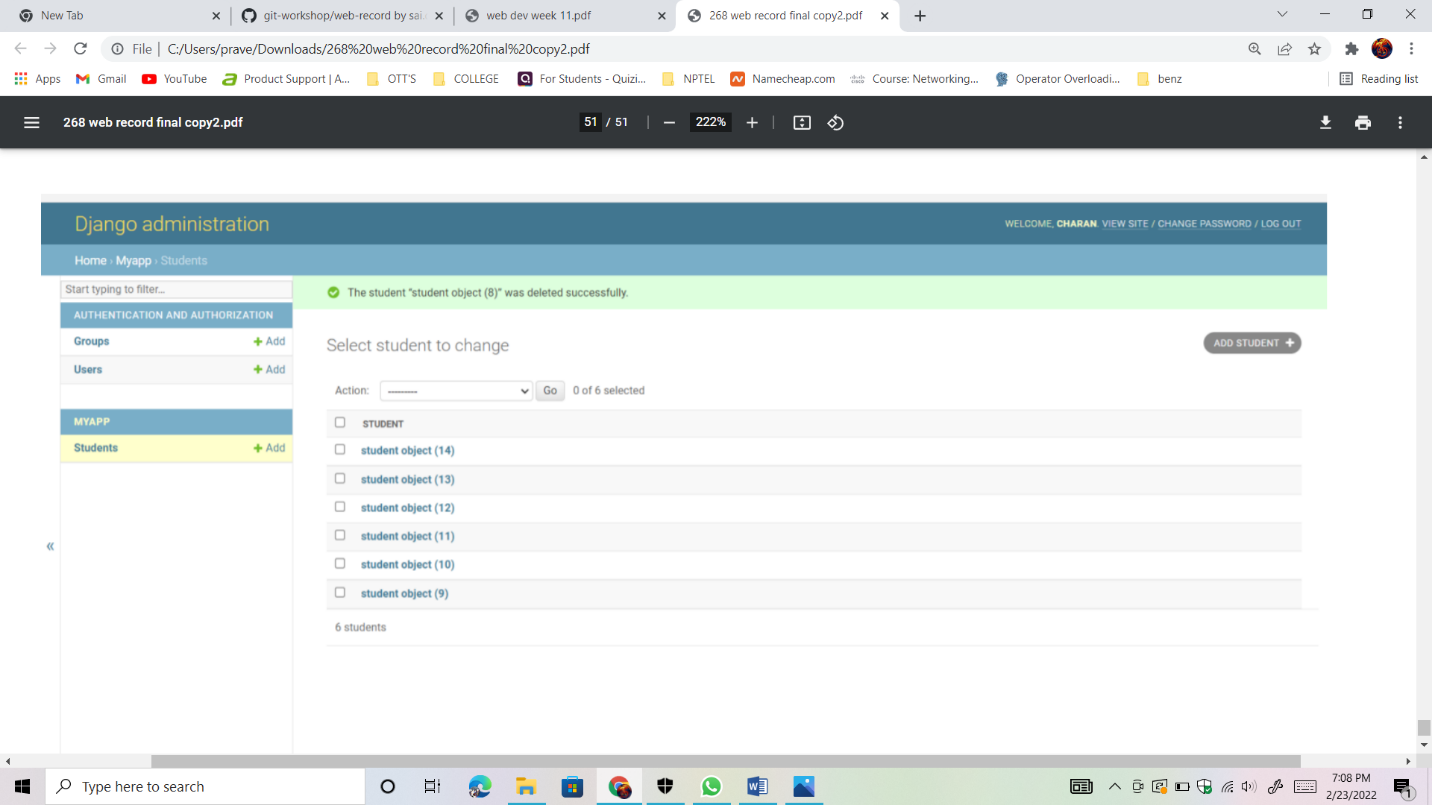


Step 3: The change is updated successfully.



DELETE:

Step 1: Select the student object to delete. (say student object(9))



# WEEK –11

**Aim:** Django Project ( URL Shortner )

# Code:

**Main file ( shortner) admin.py:**

from django.contrib import admin from .models import Url

# Register your models here. admin.site.register(Url)

# apps.py:

from django.apps import AppConfig

class ShortnerConfig(AppConfig): name = 'shortner'

**models.py:**

from django.db import models

# Create your models here. class Url(models.Model):

link = models.CharField(max\_length=10000)

uuid = models.CharField(max\_length=10)

# urls.py:

from django.urls import path from . import views

urlpatterns = [

path('', views.index, name='index'), path('create', views.create, name='create'), path('<str:pk>', views.go, name='go')

]

**views.py:**

from django.shortcuts import render, redirect import uuid

from .models import Url

from django.http import HttpResponse

# Create your views here. def index(request):

return render(request, 'index.html')

def create(request):

if request.method == 'POST': link = request.POST['link'] uid = str(uuid.uuid4())[:5]

new\_url = Url(link=link,uuid=uid) new\_url.save()

return HttpResponse(uid)

def go(request, pk):

url\_details = Url.objects.get(uuid=pk) return redirect('https://'+url\_details.link)

# URLSHORTNER:

**urls.py:**

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

path('admin/', admin.site.urls), path('', include('shortner.urls'))

]

# OUTPUT:

