**Django**

* Python framework to develop a web application
* Uses MVT (Model View Template) architecture

pip install django

django-admin startproject broadway .

python manage.py runserver

**Web Application**

* **Presentation Layer(UI/UX)**

***HTML*** – to define contents of a page

***CSS*** – to apply style to a page

***Bootstrap***– CSS library to make responsive pages

***JavaScript*** – to add user events to a page (to make a page interactive)

***JQuery*** – JavaScript library

***AJAX*** – Technique to load data to a particular component without reloading full page

***React JS, Vue JS, React Native, Flutter***

* **Business layer**
* **Database**

**HTML**

* Hypertext Markup Language
* Used to create web pages
* defines contents of a web page
* **tag** based language (<html></html>)
* tag has its properties called **attributes**
* it has tree like document structure called DOM (Document Object Model)

**Tags**

1. Heading Tags (h1-h6)
2. Paragraph Tag(p)
3. Formatting tags (b, I, u, strong, strike, em, pre, sub, sup)
4. Lists
   1. Unordered List (<ul><li></li></ul>)
   2. Ordered List (<ol><li></li></ol>)
   3. Definition List(<dl><dt></dt><dd></dd></dl>)
5. Multimedia
   1. Image (<img src=”” alt=”” />)

(<figure>

<img src=”” alt=”” />

<figcaption>Text</figcaption>

</figure>)

* 1. Audio (<audio controls><source src=”” type=”” /></audio>)
  2. Video (<video controls><source src=”” type=”” /></video>)

1. Hyperlink (anchor tag (<a href=””>Click</a>))
   1. Relative URL
   2. Absolute URL
   3. In page Link
2. Table in HTML

<table>

<tr>

<td></td>/<th></th>

</tr>

</table>

1. Division and Span (<div></div> <span></span>)
2. HTML5 Semantic elements

header, nav, section, article, aside, footer

1. Iframe
2. Form in HTML
   1. <form method=”” action=”” enctype=””></form>
   2. <label>Text</label>
   3. <input type=”” name=”” value=”” />

type=text/date/number/password/radio/checkbox/range/file/email/url/submit/button/reset

* 1. <select name=””>

<option value=””></option>

</select>

* 1. <textarea name=””></textarea>
  2. <button type=” submit/button/reset”></button>

**CSS - Selectors**

|  |  |
| --- | --- |
| Tag Selector | **p{} h1{} ul{}** |
| Class Selector | **.class\_name {}** |
| Id Selector | **#id\_name {}** |
| Descendant selector | **#id\_name p {} nav ul li{}** |
| Child Selector | **ul > li {} .div\_class > p {}** |
| Pseudo Element  Selector | **h1::first-letter{} p::first-line{}**  **h1::after{} h1::before{}** |
| Pseudo Class Selector | **a:hover{} a:active{} a:visited{} a:link{}** |
| Sibling selector | **div ~ p {}** |
| Adjacent sibling selector | **div + p {}** |
| nth child selector | **div:nth-child(3)** |
| Universal selector | **\*** |
| Multiple Element Sel | **P, h1, #id\_name {}** |

**Text Properties**

color, font-size, font-family, text-align, text-indent,

font-weight, text-decoration, font-style, text-shadow

**Box Properties**

height, width, background-color, background, margin, padding, border, border-radius

**Floating & Positioning**

* float: left/right
* clear: left/right/both
* position: absolute/fixed/relative/sticky/static
* display: none/block/inline/grid/flex

**Animation & Styles**

* transform: rotate/skew/scale
* animation

animation-name:

animation-duration:

* transition

**Database**

Stores data in the form of table. Provides different tools to manage data.

**CRUD Operation**

* CREATE DATABASE db\_name;
* SHOW DATABASES;
* USE db\_name;
* CREATE TABLE users(id int PRIMARY KEY AUTO\_INCREMENT, name varchar(100), dob Date, phone varchar(20), email varchar(100), username varchar(20) UNIQUE KEY, password varchar(20), type ENUM('admin', 'staff', 'customer'), status boolean);
* CREATE TABLE products (id int PRIMARY KEY AUTO\_INCREMENT,

category\_id int, name varchar(255),

details mediumtext, price double,

stock int,

FOREIGN KEY (category\_id) REFERENCES categories(id))

* CREATE TABLE orders (id int PRIMARY KEY AUTO\_INCREMENT,

user\_id int, product\_id int, qty int,

ordered\_date date, delivery\_address varchar(255),

FOREIGN KEY(user\_id) REFERENCES users(id),

FOREIGN KEY(product\_id) REFERENCES products(id))

* INSERT INTO users (name,dob,phone,email,username,password,type,status)

VALUES ('ram kumar', '2000-01-01', '9812345678', 'ram@email.com', 'ram123', 'ram123', 'admin', 1)

* INSERT INTO users

VALUES ('', 'ram kumar', '2000-01-01', '9812345678', 'ram@email.com', 'ram123', 'ram123', 'admin', 1)

* UPDATE users SET name='ram prasad', dob='2001-08-10' WHERE id=1
* DELETE FROM users WHERE id=3
* SELECT \* FROM users
* SELECT name, type FROM users
* SELECT \* FROM `users` WHERE name LIKE 'r%' AND dob < '2002-01-01';
* SELECT \* FROM products

JOIN categories ON categories.id=products.category\_id

* SELECT products.name, categories.title, products.price FROM products

JOIN categories ON categories.id=products.category\_id

WHERE categories.title = 'Kitchen Ware'

* TRUNCATE TABLE product
* DROP TABLE product
* ALTER : see yourself

**Web application**

**News Portal**

* **user**
  + name
  + email
  + username
  + password
  + type
  + status
* **category**
  + title
  + image
  + status
* **news/article**
  + category\_id
  + title
  + date
  + details
  + author
  + image
  + status
  + user\_id
* **advertisement**
  + company
  + product
  + title
  + image
  + details
  + category
  + status
  + user\_id
* **comment**
  + user\_id
  + news\_id
  + datetime
  + status
* **about**
  + title
  + details
  + status

pip install django

django-admin startproject broadway .

python manage.py runserver

django-admin startapp module\_name

python manage.py makemigrations

python manage.py migrate

python manage.py createsuperuser