

---

# Model Deployment

## Creating a Frontend/ UI



Democratizing Data Science Learning

# Learning Objectives

---

**Frontend and  
Backend  
Development**

**HTML, CSS, &  
Javascript**

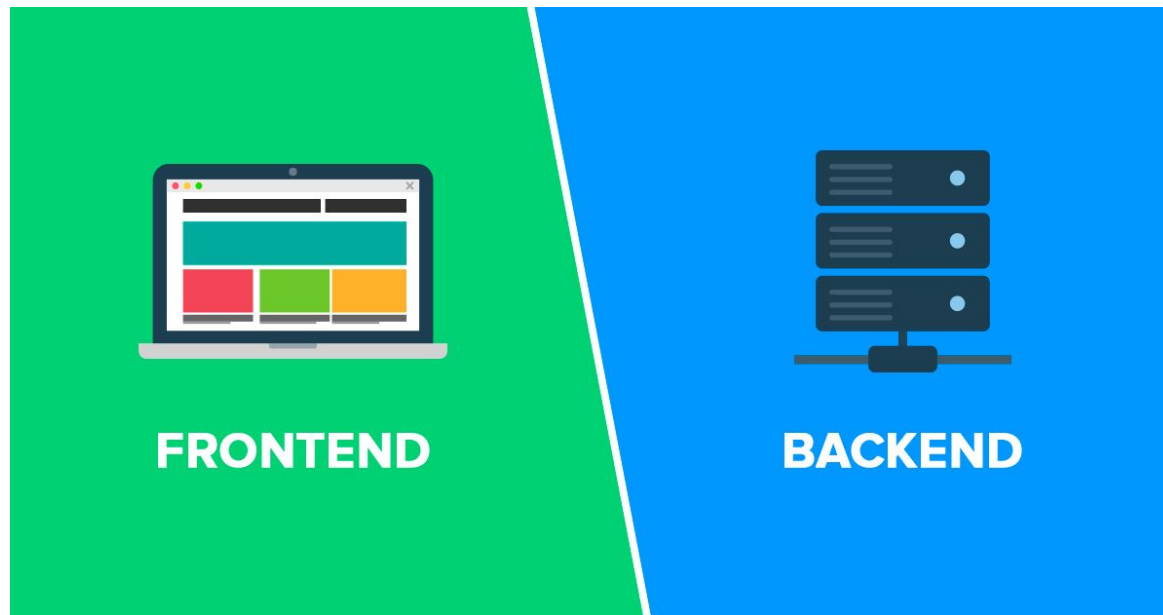
**Creating a  
Website/UI**

# Frontend and Backend Development

---

Frontend and Backend are two most popular terms used in web development. These terms are very crucial for web development but are quite different from each other. Each side needs to communicate and operate effectively with the other as a single unit to improve the website's functionality.

Languages used for front end are HTML, CSS, Javascript while those used for backend include Java, Ruby, Python, .Net .



# Frontend

---

The part of a website that user interacts with directly is termed as front end.

It is also referred to as the 'client side' of the application.

It includes everything that users experience directly: text colors and styles, images, graphs and tables, buttons, colors, and navigation menu.

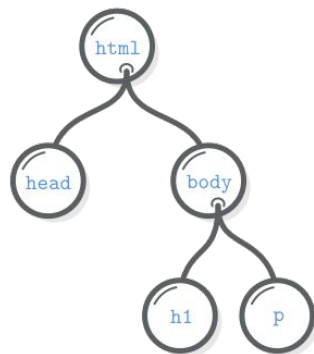
User Interface (UI) development is the part of Frontend development.

# HTML, CSS, & Javascript

---

HyperText Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript are the languages that run the web. They're very closely related, but they're also designed for very specific tasks.

- **HTML** provides the basic structure of sites, which is enhanced and modified by other technologies like CSS and JavaScript.
- **CSS** is used to control presentation, formatting, and layout.
- **JavaScript** is used to control the behavior of different elements.



HTML



CSS

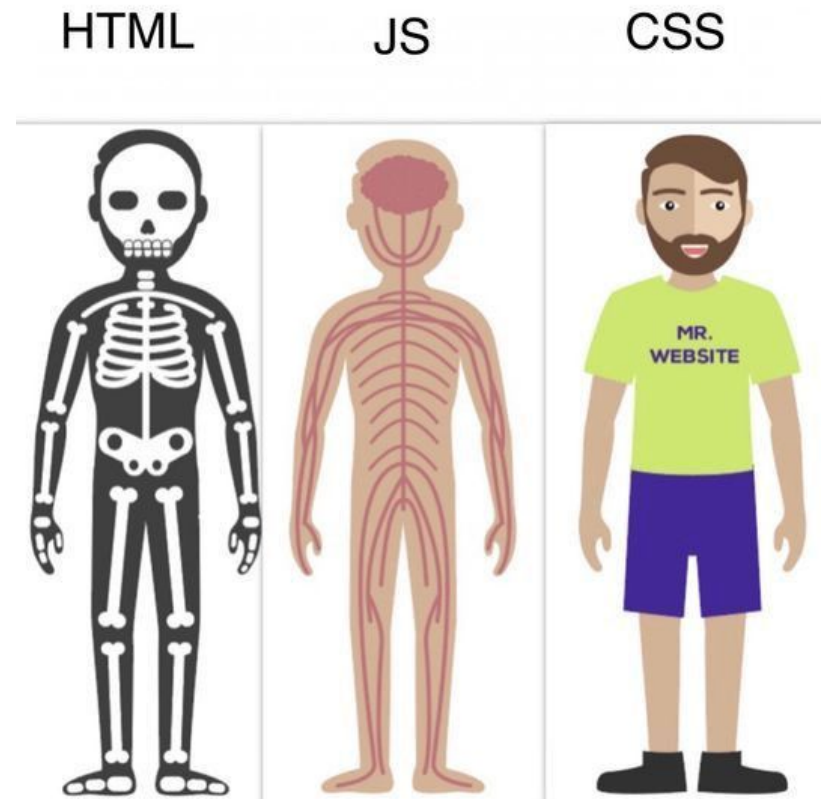


JAVASCRIPT

# HTML, CSS, & Javascript

Let's understand this with a simple analogy.

- HTML is the bare bones that give a structure to our body.
- Javascript is the vessels that enable blood flow and thus help our body function.
- CSS is just our appearance.



# HTML, CSS, & Javascript

---

These concepts are actually the building blocks of web development.

In the tutorial, understanding a basic overview of 'what elements are performing which function' will be sufficient.

However, if you wish to learn these concepts properly, you can go through the free resource here:

[https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)

# The directory structure

As a refresher, this is the folder structure for our application:

```
webapp/
├── server/
│   ├── artifacts/
│   │   ├── banglore_home_prices_model.pickle
│   │   └── columns.json
│   ├── server.py
│   └── util.py
├── client/
│   ├── app.html
│   ├── app.css
│   └── app.js
└── model/
    └── model.ipynb
```

We previously worked with the server folder. Now, we'll be working with the client folder.

You'll find all the files that'll be used in the tutorial here:

[https://github.com/dphi-official/Micro-Courses/tree/master/Introduction Model Deployment/client](https://github.com/dphi-official/Micro-Courses/tree/master/Introduction%20Model%20Deployment/client)



# Creating a Website/UI

## Data Science Project

Real Estate

Price Prediction

7

# Integrating Frontend and Backend

---

Till now, what all has been implemented is: You need to run the `server.py` file using `python server.py` but it doesn't run your frontend automatically. The HTML File needs to be opened separately for that purpose.

Now, we'll be making some changes such that on running the Flask application, our website runs automatically.

In the HTML File, the 6th and 7th lines are changed to

```
<script src="{{ url_for('static',filename='app.js') }}"></script>  
<link rel="stylesheet" href= "{{ url_for('static',filename='app.css') }}">
```

# Integrating Frontend and Backend

---

Next, we'll be linking the Flask application to the frontend files.

We'll simply edit the creating the Flask app line in the server.py file for this purpose:

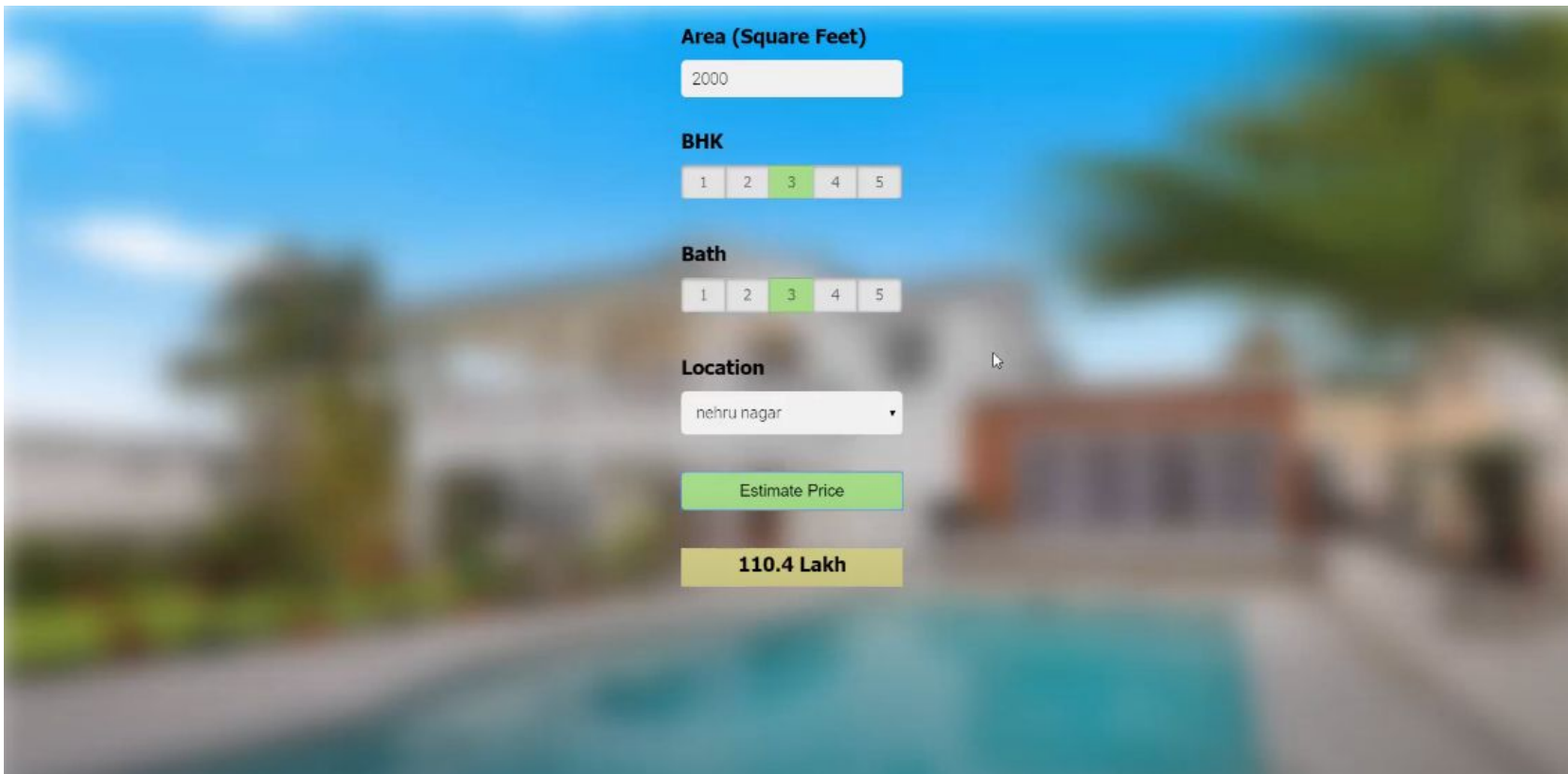
In the server.py File, the 4th line is changed to  
`app = Flask(__name__, static_url_path="/client", static_folder='../client',  
template_folder="../client")`

# Final Product

---

And that's it!

You should now be able to run your application locally at <http://127.0.0.1:5000> and it would look something like this:



The image shows a web application interface for estimating property prices. The background is a blurred image of a swimming pool and buildings. The form is centered and contains the following elements:

- Area (Square Feet)**: A text input field with the value "2000".
- BHK**: A row of five buttons labeled 1, 2, 3, 4, and 5. The button labeled "3" is highlighted in green.
- Bath**: A row of five buttons labeled 1, 2, 3, 4, and 5. The button labeled "3" is highlighted in green.
- Location**: A dropdown menu with the text "nehru nagar" and a downward arrow.
- Estimate Price**: A green button with the text "Estimate Price".
- 110.4 Lakh**: A yellow box displaying the estimated price.

# What Next?

---

Please note: CSS and Javascript are optional elements for creating a website. You could've just created a bare form that takes in input and prints an output as well.

So now that you are able to use your website, why not make it available to the world so that anyone and everyone can use it? That's what we will cover in the last unit of this course.

# References

---

- <https://www.geeksforgeeks.org/frontend-vs-backend/>

# Slide Download Link

---

You can download this unit from the below link:

[https://docs.google.com/presentation/d/1YmPfv86Yo4N7WU\\_Mjyv\\_rJ6ktvImY4VM12EQR\\_ax5pRQ/edit?usp=sharing](https://docs.google.com/presentation/d/1YmPfv86Yo4N7WU_Mjyv_rJ6ktvImY4VM12EQR_ax5pRQ/edit?usp=sharing)

---

That's it for this unit. Thank you!

Feel free to post any queries on [Discuss](#).