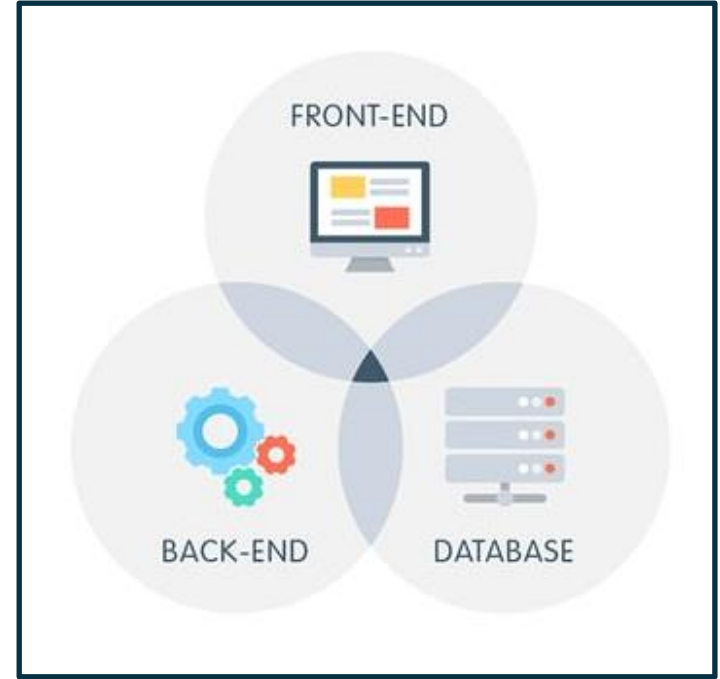


Full Stack Web Development

Roshan David Jathanna

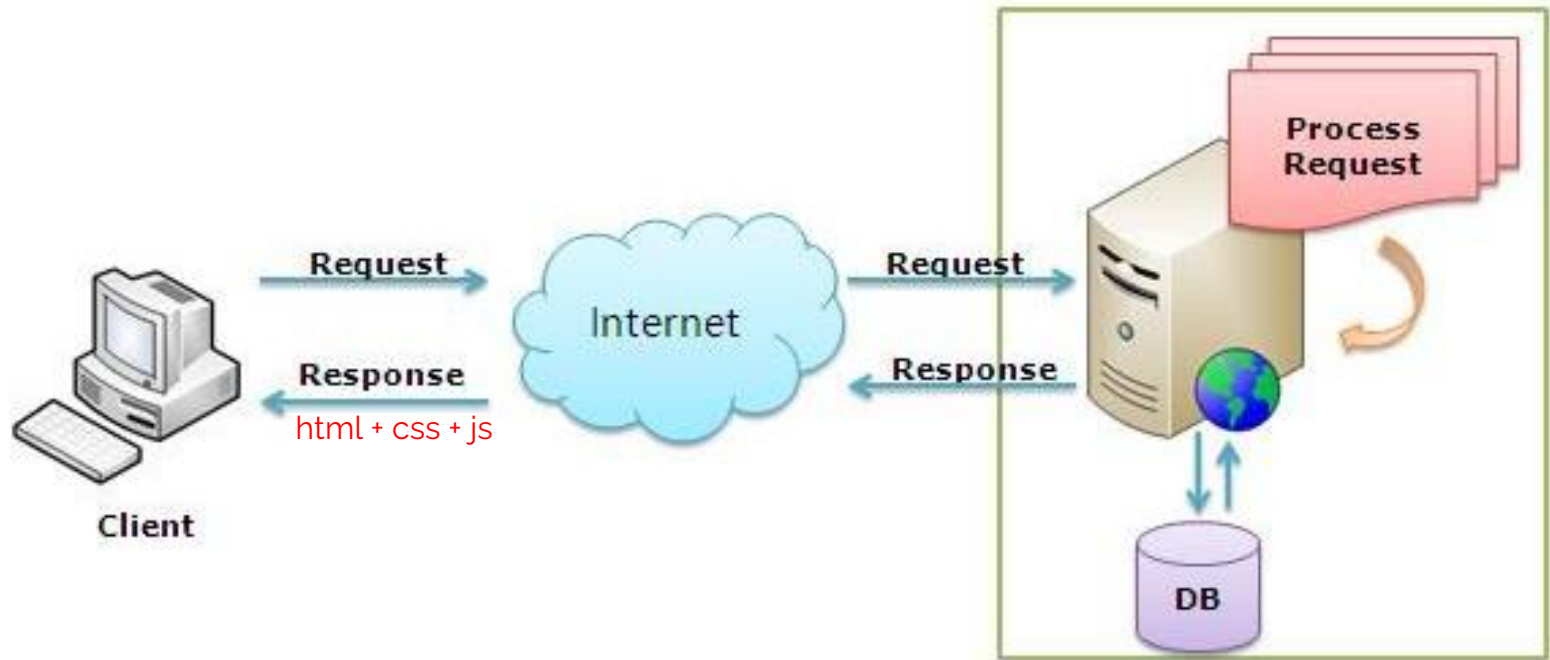
roshan.jathanna@manipal.edu



► Course Objectives

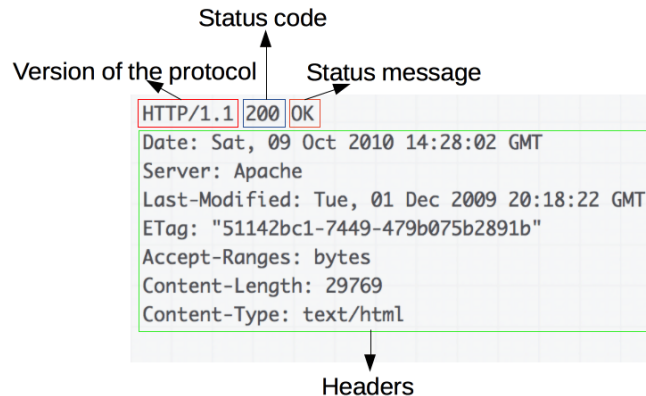
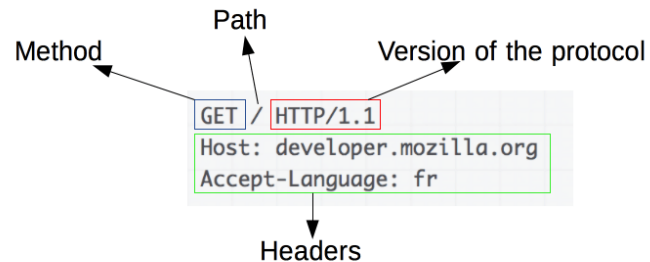
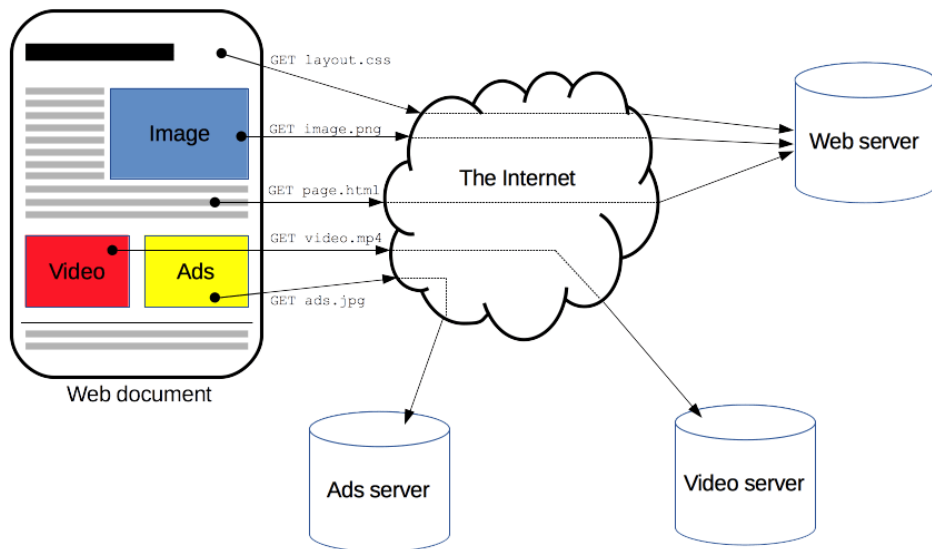
1. Understand and apply the foundational principles of web development using HTML, CSS, and JavaScript.
2. Utilize front-end web UI frameworks such as Bootstrap for creating responsive and visually appealing interfaces.
3. Develop dynamic and interactive user interfaces using React
4. Develop server-side applications using NodeJS and Express
5. Integrate MongoDB as the database, utilizing Mongoose ODM for efficient data storage and retrieval

Web Vs Windows Application











HTTP Protocol



► HTTP Methods & Status Code

 GET	 POST	 PUT	 DELETE	 PATCH	 HEAD
retrieve data from server	add data to an existing file or resource	update(replace) an existing file or resource in server	delete data from server	update a resource partially (modify)	retrieve the resource's headers

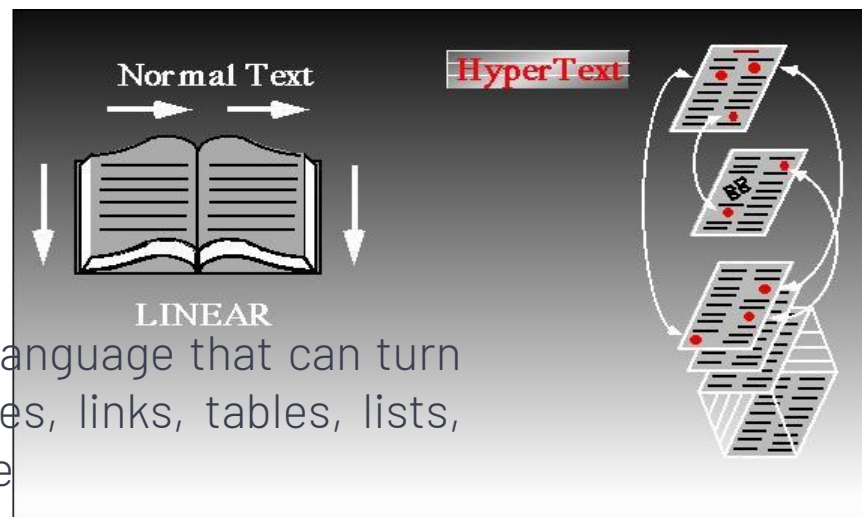
Informational	100-199	Level 200 (Success) 200 : OK 201 : Created 203 : Non-Authoritative Information 204 : No Content	Level 400 400 : Bad Request 401 : Unauthorized 403 : Forbidden 404 : Not Found 409 : Conflict	Level 500 500 : Internal Server Error 503 : Service Unavailable 501 : Not Implemented 504 : Gateway Timeout 599 : Network timeout 502 : Bad Gateway
Successful	200 to 299			
Redirection	300 to 399			
Client error	400 to 499			
Server error	500 to 599			

► HTML



- ▶ H is for Hyper
 - ▶ T is for Text
- } Text With Links

- ▶ M is for Markup
 - ▶ L is for Language
- } programming language that can turn text into images, links, tables, lists, and much more



► HTML Elements

<p> This is my first paragraph. </p>

**Opening
Tag**

+ Content +

**Closing
Tag**

Some self closing tags:
 <hr> <embed> <input> <link> <meta>

► Skeleton of the HTML Page

```
<!DOCTYPE html>
<html>
  <head>
    <title>My First Page</title>
  </head>
  <body>
    <p>This is a paragraph.</p>
  </body>
</html>
```

- ▶ First Line is `<!DOCTYPE html>`
 - ▶ This tells the browser what language it's reading (in this case, HTML).
- ▶ Next line is `<html>`
 - ▶ This starts the HTML document.
- ▶ Last line is `</html>`
 - ▶ This ends the HTML document.
- ▶ Html file contains two parts head and body
 - ▶ Head - information about the webpage, such as its title
 - ▶ Body - information to be displayed in the web page

VS Code

The image shows the Visual Studio Code website on the left and a screenshot of the VS Code application interface on the right.

Visual Studio Code Website:

- Navigation links: Docs, Updates, Blog, API, Extensions, FAQ, Learn.
- Search Docs and Download buttons.
- Message: Version 1.76 is now available! Read about the new features and fixes from February.
- Headline: Code editing. Redefined.
- Text: Free. Built on open source. Runs everywhere.
- Download for Windows (Stable Build) button.
- Text: Web, Insiders edition, or other platforms.
- Text: By using VS Code, you agree to its license and privacy statement.


VS Code Application Interface:

- File Explorer: Extensions Marketplace, @sortinstalls.
- Search Bar: Python, C/C++, ESLint, Debugger for Chrome, Language Support, vscode-icons, Vetur.
- Code Editor: JS serviceWorker.js, register, window.addEventListener('load') callback, checkValidServiceWorker(swUrl, config);, // Add some additional logging to localhost, p..., // service worker/PWA documentation., navigator.serviceWorker.ready.then(() => { { product, productSub, removeSiteSpecificTrackingException, removeWebWideTrackingException, requestMediaKeySystemAccess, sendBeacon, serviceWorker (property) Navigator.serviceWorker..., storage, storeSiteSpecificTrackingException, storeWebWideTrackingException, userAgent, vendor }, function registerValidSW(swUrl, config) { navigator.serviceWorker .register(swUrl) .then(registration => {
- Terminal: You can now view create-react-app in the browser.


<https://code.visualstudio.com/>




VS Code Extensions



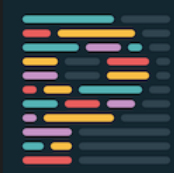
Live Server v5.7.9

Ritwick Dey |  31,909,007 | ★★★★★ (431)




Launch a development local Server with live reload feature for static & dynamic pages

[Disable](#) [Uninstall](#) 


This extension is enabled globally.




Prettier - Code formatter v9.10.4

Prettier  prettier.io |  29,953,555 | ★★★★★☆ (382) |  Sponsor



Code formatter using prettier

[Disable](#) [Uninstall](#) 


This extension is enabled globally.



Simple React Snippets v1.2.7








Burke Holland  burkeholland.dev |  2,583,863 | ★★★★★ (27)

Dead simple React snippets you will actually use

[Disable](#) [Uninstall](#) 

This extension is enabled globally.

File Edit Selection View Go Run Terminal

SOURCE CONTROL

Install Git, a popular source control system, to track code changes and collaborate with others. Learn more in our [Git guides](#).

[Download Git for Windows](#)

After installing, please [reload](#) (or [troubleshoot](#)). Additional source control providers can be installed from the [Marketplace](#).

► HTML Headings

`<h1>` Main Heading `</h1>`

`<h2>` Sub Heading `</h2>`

`<h3>` Sub Heading of Sub Heading `</h3>`

`<h4>` ... `</h4>`

`<h5>` ... `</h5>`

`<h6>` ... `</h6>`

► HTML Formatting

`Bold`

Bold

`<i>Italic</i>`

Italic

`<u>Underlined</u>`

Underlined

Subscripts: `f₀ + f₁`

Subscripts: $f_0 + f_1$

Superscripts: `x² + y²`

Superscripts: $x^2 + y^2$

`<strike>Strike Through</strike>`

~~Strike Through~~

`<i>Bold Italic</i>`

Bold Italic

► HTML Links

- Creates a hyperlink to a Uniform Resource Locator

`clickable text`

- Creates a hyperlink to an email address

`clickable text`

- Creates a target location within a document

``

- Creates a link to that target location

`clickable text`

▶ HTML Links

```
<!DOCTYPE html>
<html>
  <body>
    <p>
      <a href="default.asp">HTML Tutorial</a>
      This is a link to a page on this website.
    </p>
    <p>
      <a href="http://www.w3.org/">W3C</a>
      This is a link to a website on the World Wide Web.
    </p>
  </body>
</html>
```

► HTML Images

```

```

► Ordered List

```
<!DOCTYPE html>
<html>
  <body>
    <h4>An Ordered List:</h4>
    <ol type="1">
      <li>Coffee</li>
      <li>Tea</li>
      <li>Milk</li>
    </ol>
  </body>
</html>
```




Unordered List

```
<!DOCTYPE html>
<html>
  <body>
    <h4>An Unordered List:</h4>
    <ul type="disc">
      <li>Coffee</li>
      <li>Tea</li>
      <li>Milk</li>
    </ul>
  </body>
</html>
```

Tables

```
<table border="1">
  <tr>
    <th colspan="3">Common Heading</th>
  </tr>
  <tr>
    <th>Heading 1</th>
    <th>Heading 2</th>
    <th>Heading 3</th>
  </tr>
  <tr>
    <td>Row 1 Col 1</td>
    <td>Row 1 Col 2</td>
    <td rowspan="2">Row 1 & Row 2 Col 3</td>
  </tr>
  <tr>
    <td>Row 2 Col 1</td>
    <td>Row 2 Col 2</td>
  </tr>
</table>
```

Common Heading		
Heading 1	Heading 2	Heading 3
Row 1 Col 1	Row 1 Col 2	Row 1 & Row 2 Col 3
Row 2 Col 1	Row 2 Col 2	

▶ Getting Input

An HTML form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more.

`<Input>`

Type of input is specified with type attribute

Text Fields

Used when we want to type numbers or text.

```
<input type="text" name="first name" />
```

▶ Getting Input

Radio Buttons

Used when we want user to select one of a limited number of choices

☒ Male
☐ Female

Check Boxes

Used when we want user to select one or more options of a limited number of choices

☒ I have a bike
☒ I have a car

Submit Button

When user clicks on submit button, the content of the form is send to the server

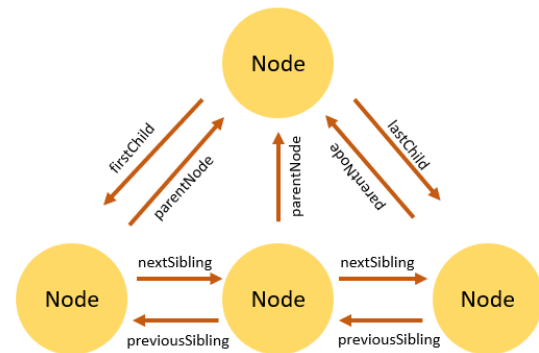
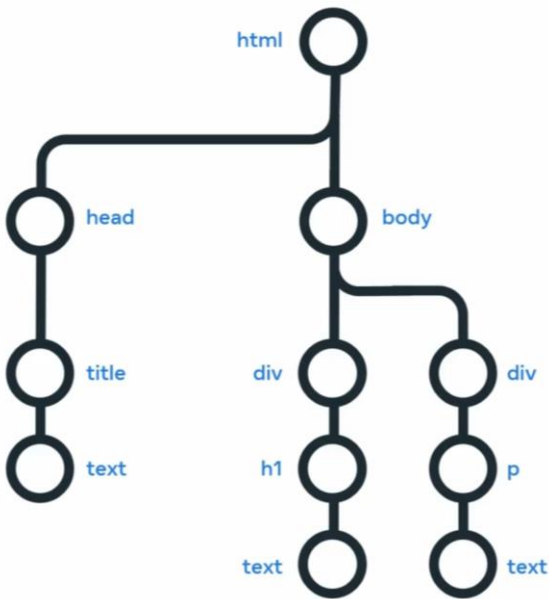
HTML Forms

```
<form action="">
  Name <input type= "text" name="username"/>
    <br />
  <input type="radio" name="gender" value="male" />Male
  <input type="radio" name="gender" value="female" />Female
  <br />
  <input type="checkbox" name="vehicle" value="Bike" />I have a bike
  <br />
  <select name="bikes">
    <option value="honda">Honda</option>
    <option value="hero">Hero</option>
  </select>    <br />
  <textarea rows="10" cols="30">Enter more information here...</textarea>
  <br />
  <input type="button" value="Save" />
</form>
```



Document Object Model (DOM)

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Website</title>
  </head>
  <body>
    <div>
      <h1>This is the heading</h1>
    </div>
    <div>
      <p>This is the paragraph</p>
    </div>
  </body>
</html>
```



► XML

- ▶ XML documents must start with an XML declaration like
`<?xml version="1.0"?>`
- ▶ XML elements are case sensitive
- ▶ All elements must be nested in a single root element.
- ▶ Every element must be fully enclosed.
`<Product><ID></ID></Product>` is valid, but
`<Product><ID></Product></ID>` isn't
- ▶ Attributes contain values that are associated with an element
`<element attribute="value"></element>`





JSON

- ▶ Data is represented as field name (in double quotes), followed by a colon, followed by a value
- ▶ Value can be a string, a number, an object, an array, a boolean or null
- ▶ Data is separated by commas
- ▶ Curly braces hold objects
- ▶ Square brackets hold arrays

