Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology, Pune-37

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

Department of Computer Engineering

Lab Manual

Course Code	Course Name	Teaching Scheme(Hrs./ Week)	Credits
CS3215	**Web Technology**	**2**	**1**

Course Outcomes:

1. To summarize most commonly used HTML5 and CSS3 tags and attributes for website development.
2. To associate event handling with HTML5 forms and CSS3 using JavaScript as a front-end technology for website development.
3. To extend HTML5 and CSS3 and JavaScript front end technologies with PHP and MySQL as a server side and backend technologies for website development.
4. To simplify website development using REST API and Spring boot as server-side technologies.
5. To build single page applications using REACT as a reusable UI component technology as client-side technology.
6. To assemble REACT as a front-end technology and Node as a server-side technology to develop enterprise applications

Class: - TY Branch:- Computer Engineering

Year:- 2021-22 Prepared By:- M. L. Dhore and S.M. Jaybhaye

Required H/W and S/W: Linux, Open Source Software, P-IV 8GB RAM

Vision and Mission of Institute

VISION

"To be Globally Acclaimed Institute in Technical Education and Research for Holistic Socio-Economic Development"

MISSION

• To ensure that 100% students are employable and employed in Industry, Higher Studies, Become Entrepreneurs, Civil / Defense Services / Govt. Jobs and other areas like Sports and Theatre

• To Strengthen Academic Practices in terms of Curriculum, Pedagogy, Assessment and Faculty Competence

• Promote Research Culture among Students and Faculty through Projects and Consultancy

• To make students Socially Responsible Citizen

Vision and Mission of Computer Engineering Department

VISION

"To be a leader in the world of computing education practising creativity and innovation"

MISSION

• To ensure students' employability by developing aptitude, computing, soft, and entrepreneurial skills

• To enhance academic excellence through effective curriculum blended learning and comprehensive assessment with active participation of industry

• To cultivate research culture resulting in knowledge-base, quality publications, innovative products and patents

• To develop ethical consciousness among students for social and professional maturity to become responsible citizens

Contents

Experiment Number: 01

CO Mapping	PO Mapping
CO1 & CO2	P02 & PO3

Assignment Objective: To learn the process of installation and configuration of http server

Assignment Outcome: Able to install and use it as a web server in web development

Problem Statement: Installation and configuration and testingworking of XAMPP server for local host.

Installing XAMPP

Our XAMPP tutorial will take you through the installation process for the software package on Windows. If you're using Linux or Mac OS X, then the steps listed below for the installation process may differ.

Step 1: Download

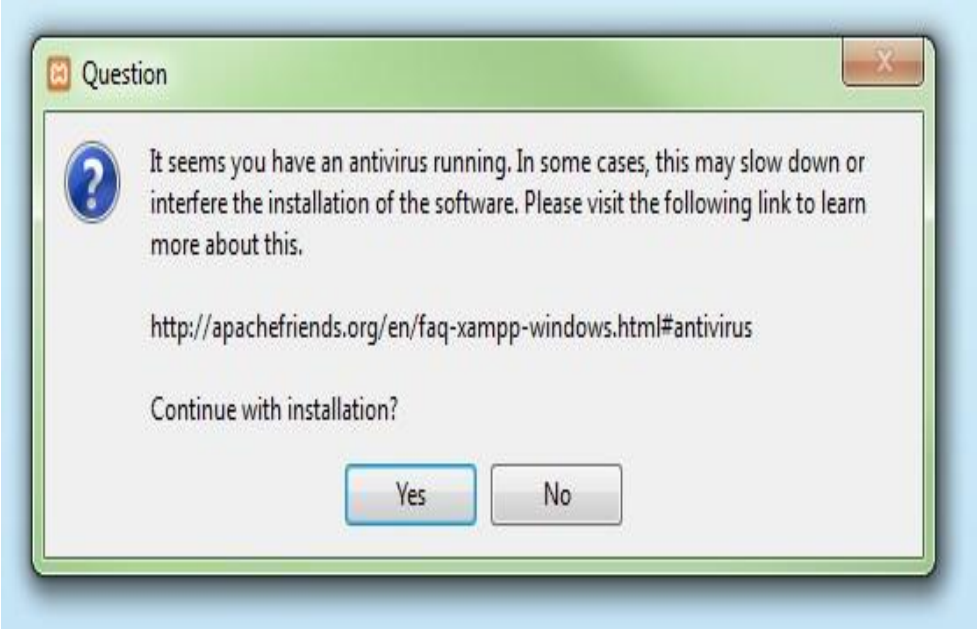
XAMPP is a release made available by the non-profit project Apache Friends. Versions with PHP 5.5, 5.6, or 7 are available for download on the Apache Friends website.

Step 2: Run .exe file

Once the software bundle has been downloaded, you can start the installation by double clicking on the file with the ending .exe.

Step 3: Deactivate any antivirus software

Since an active antivirus program can negatively affect the installation process, it's recommended to temporarily pause any antivirus software until all XAMPP components have successfully been installed.



installing XAMPP, it is advisable to disable the anti-virus program temporarily

Step 4: Deactivate UAC

User Account Control (UAC) can interfere with the XAMPP installation because it limits writing access to the C: drive, so we recommend you deactivate this too for the duration of the installation process. To find out how to turn off your UAC, head to the Microsoft Windows

support pages.

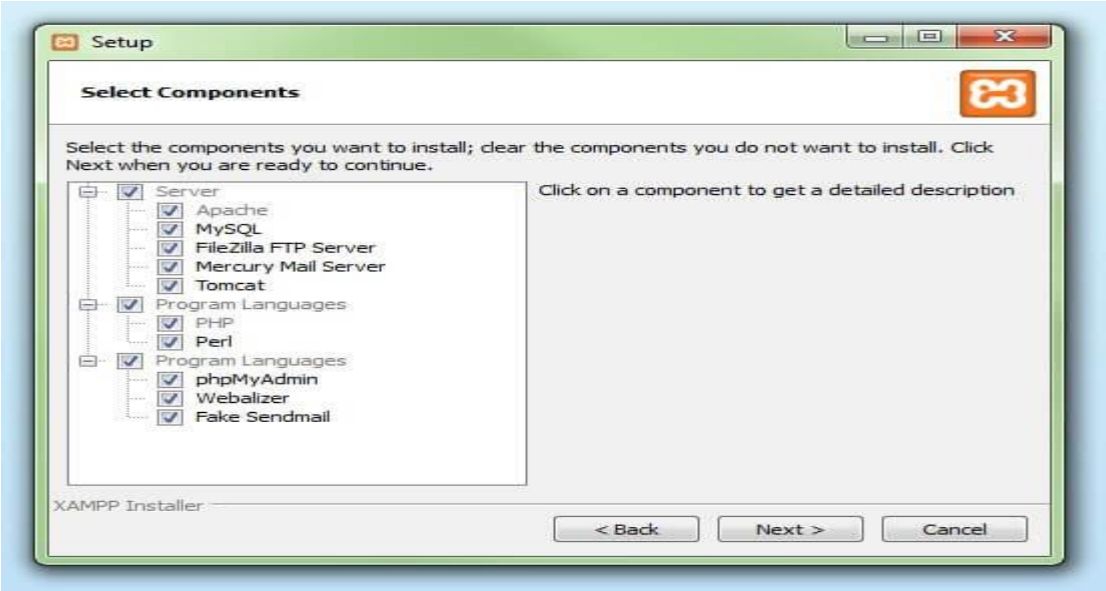
account control can affect the installation of XAMPP

Step 5: Start the setup wizard

After you've opened the .exe file (after deactivating your antivirus program(s) and taken note of the User Account Control, the start screen of the XAMPP setup wizard should appear automatically. Click on 'Next' to configure the installation settings.can start the setup on the start-up screen

Step 6: Choose software components

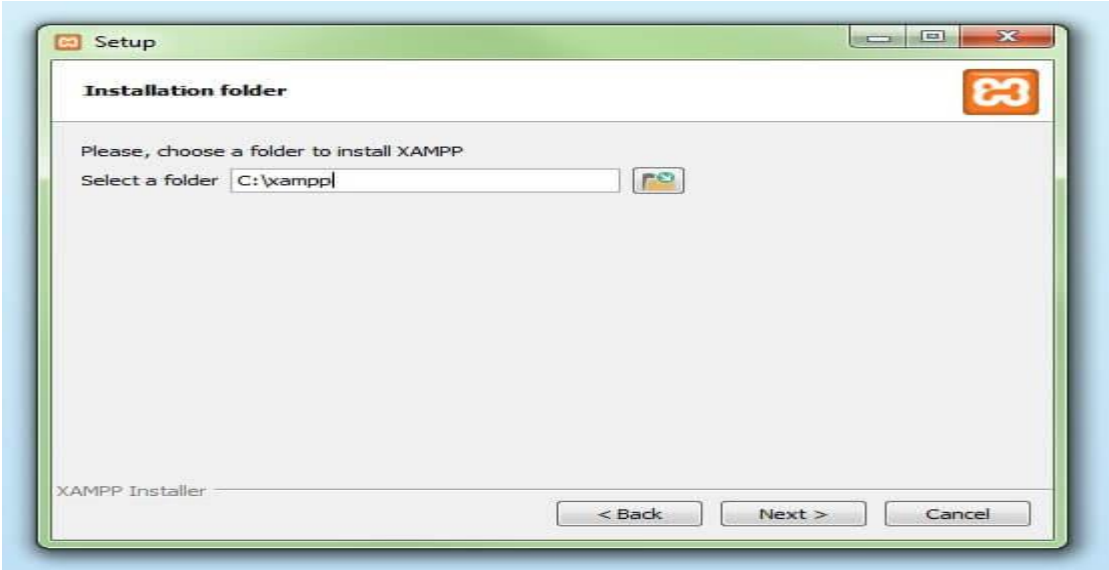
Under 'Select Components', you have the option to exclude individual components of the XAMPP software bundle from the installation. But for a full local test server, we recommend you install using the standard setup and all available components. After making your choice, click 'Next'.



dialog window entitled 'select components', you can choose the software components before installation

Step 7: Choose the installation directory

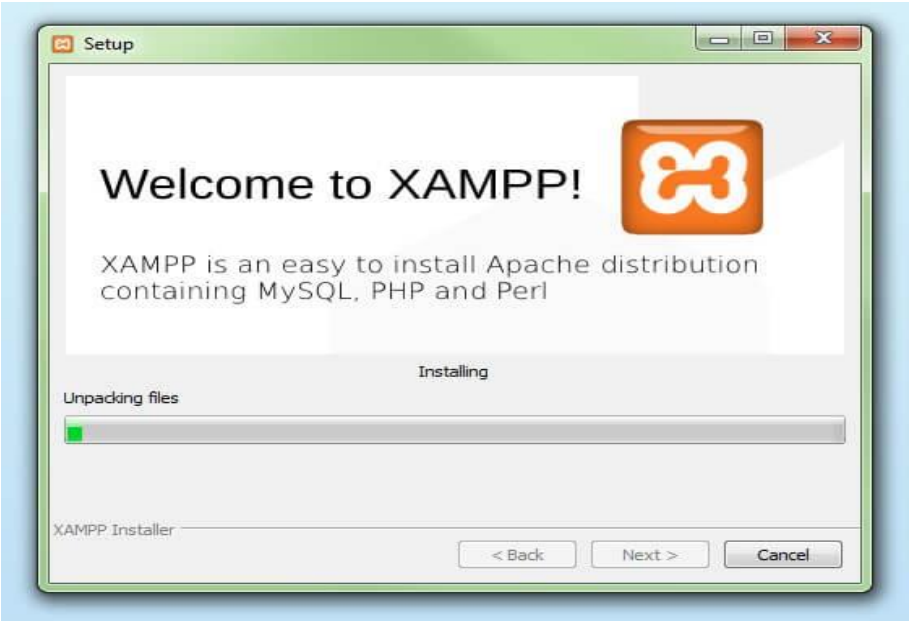
In this next step, you have the chance to choose where you'd like the XAMPP software packet to be installed. If you opt for the standard setup, then a folder with the name XAMPP will be created under C:\ for you. After you've chosen a location, click 'Next'.



For the next step, you need to select the directory where XAMPP should be installed

Step 8: Start the installation process

Once all the preferences have been decided, click to start the installation. The setup wizard will unpack and install the selected components and save them to the designated directory. This process can take several minutes in total. You can follow the progress of this installation by keeping an eye on the green loading bar in the middle of the screen.



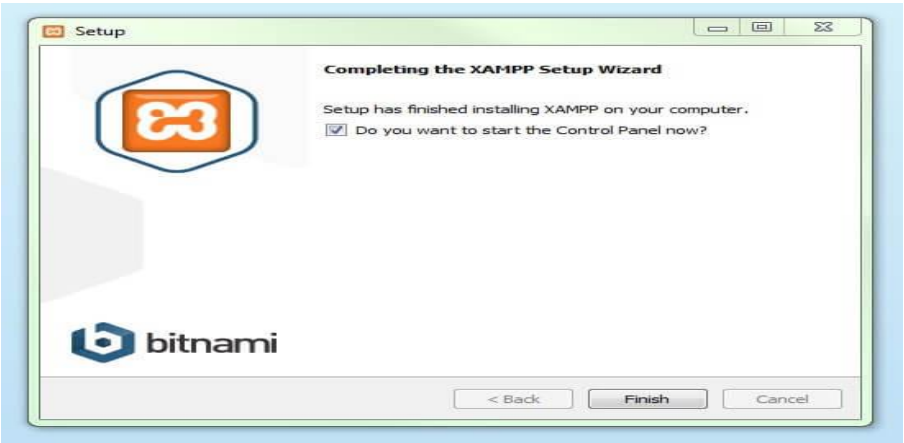
default settings, the selected software components are unpacked and installed in the target folder

Step 9: Windows Firewall blocking

Your Firewall may interrupt the installation process to block some components of the XAMPP. Use the corresponding check box to enable communication between the Apache server and your private network or work network. Remember that making your XAMPP server available for public networks isn't recommended.

Step 10: Complete installation

Once all the components are unpacked and installed, you can close the setup wizard by clicking on 'Finish'. Click to tick the corresponding check box and open the XAMPP Control Panel once the installation process is finished.

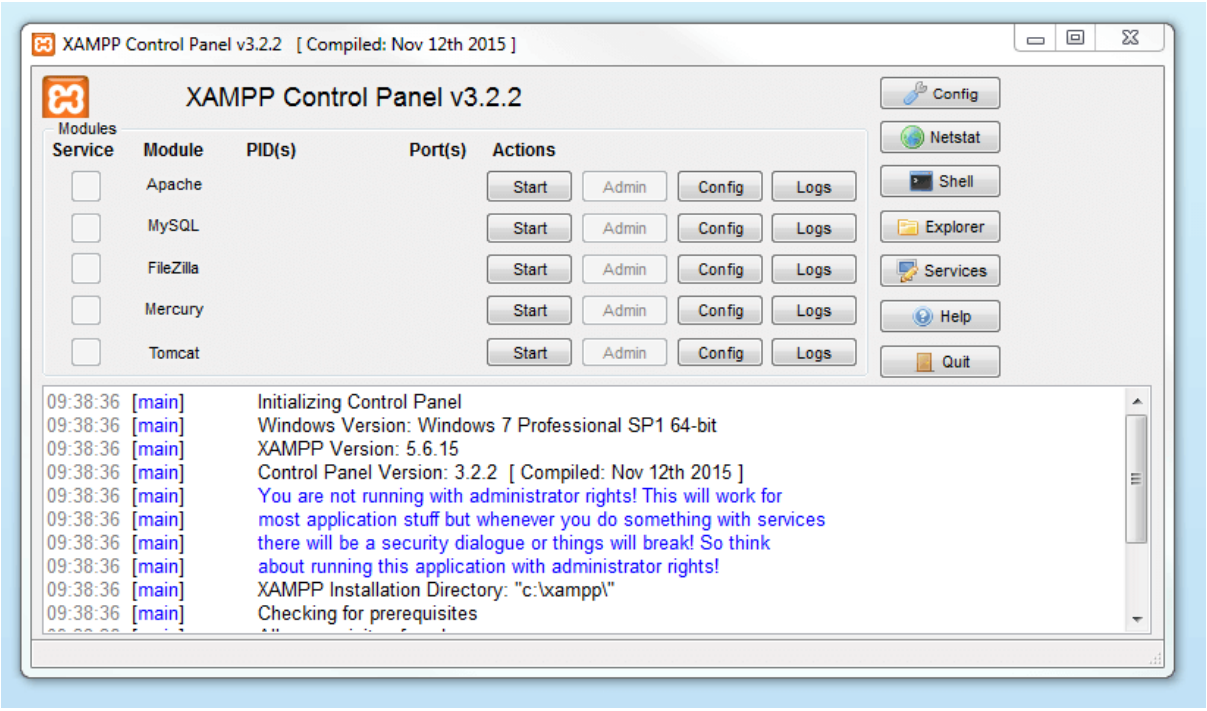


By clicking on 'finish', the XAMPP Setup Wizard is completed.

The XAMPP Control Panel

Controls for the individual components of your test server can be reached through the XAMPP Control Panel. **The clear user interface** logs all actions and allows you to start or stop individual modules with a single. The XAMPP Control Panel also offers you various other buttons, including:

- **Config:** allows you to configure the XAMPP as well as the individual components
- **Netstat:** shows all running processes on the local computer
- **Shell:** opens a UNIX shell
- **Explorer:** opens the XAMPP folder in Windows Explorer
- **Services:** shows all services currently running in the background
- **Help:** offers links to user forums
- **Quit:** closes the XAMPP Control Panel



In the Control Panel, you can start and stop individual modules

---

1) VS Code:-

   1] Download the Visual Studio Code installer for Windows.

   2] Once it is downloaded, run the installer (VSCodeUserSetup-{version}.exe). Thiswill only take a minute.

3] By default, VS Code is installed under
C:\users\{username}\AppData\Local\Programs\Microsoft VS Code.

Alternatively, you can also download a Zip archive, extract it and run Code from there.

Note: .NET Framework 4.5.2 or higher is required for VS Code. If you are using Windows 7, make sure you have at least .NET Framework 4.5.2 installed. You can check your version of .NET Framework using this command, reg query "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\NET Framework Setup\NDP\v4\full" /v version from a command prompt.

Tip: Setup will add Visual Studio Code to your %PATH%, so from the console you can type 'code .' to open VS Code on that folder. You will need to restart your console after the installation for the change to the %PATH% environmental variable to take effect.

Experiment Number: 02

CO Mapping	PO Mapping
CO1 & CO2	P02 & PO3

Assignment Objective: To learn the broad categories of HTML5 and CSS3 elements

Assignment Outcome: Able to make use of HTML and CSS tags to build simple websites

Problem Statement: Develop a basic web page using the HTML and CSS tags

Description:

What is HTML?

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content

A Simple HTML Document

```
<!DOCTYPE html>
<html>

<head>
<title>Page Title</title>
</head>

<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>

</html>
```

- The <!DOCTYPE html> declaration defines that this document is an HTML5 document
- The <html> element is the root element of an HTML page
- The <head> element contains meta information about the HTML page
- The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The <h1> element defines a large heading
- The <p> element defines a paragraph

```
<html>
<head>
<title>Page title</title>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
</body>
</html>
```

What is CSS?

CSS -
CSS is the language we use to style an HTML document. CSS describes how HTML elements should be displayed. CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

CSS Example

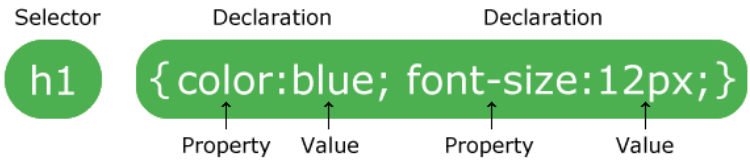
```
<style>
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
</style>
```

CSS Saves a Lot of Work!

The style definitions are normally saved in external .css files.

CSS Syntax

A CSS rule-set consists of a selector and a declaration block:

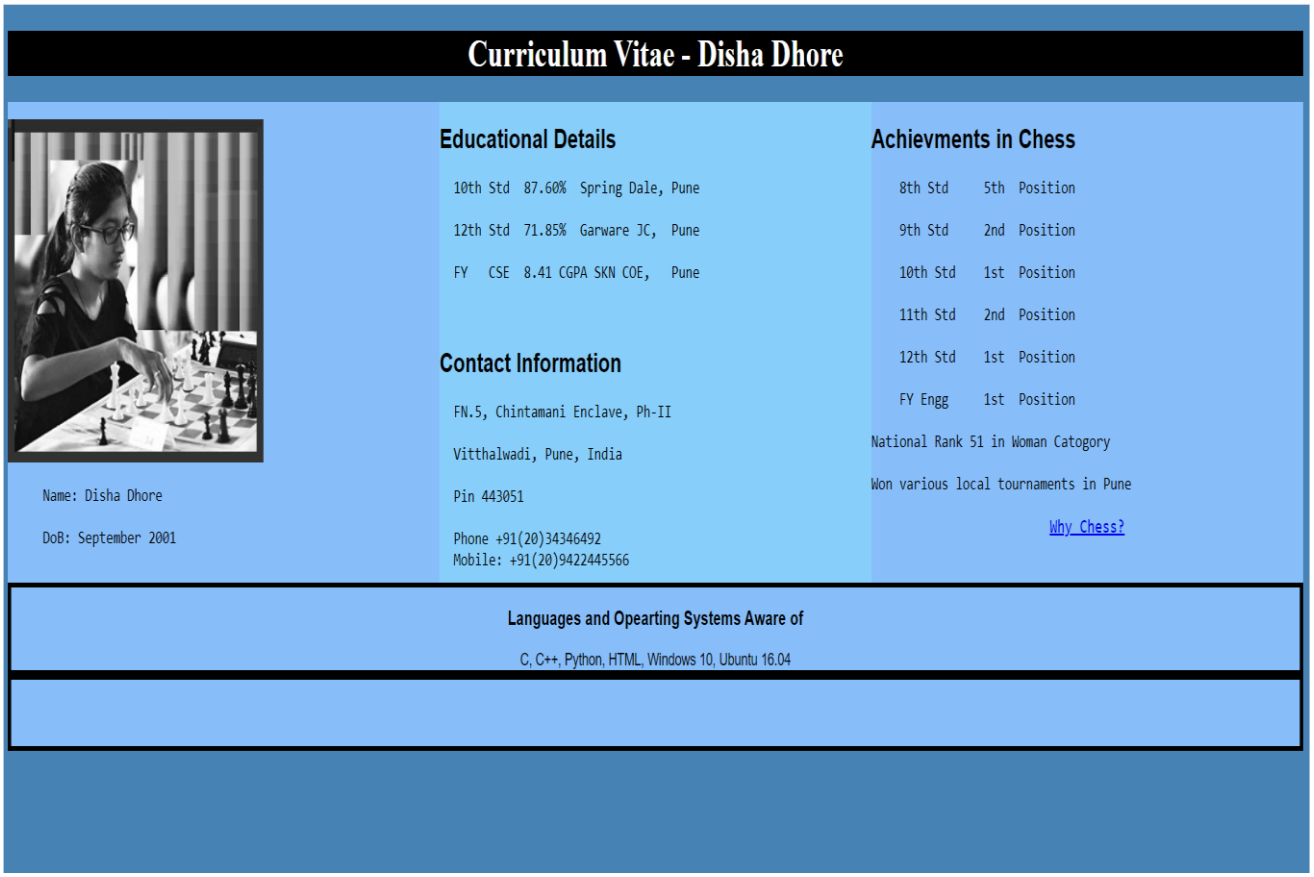


Example

In this example all <p> elements will be center-aligned, with a red text color:

```
p {
  color: red;
  text-align: center;
}
```

Sample Output



Experiment Number: 03

Title: Perform the validation for website developed in lab assignment number 2 using JavaScript.

CO Mapping	PO Mapping
CO1, CO2 and CO3	P02,PO3,PO4,PO5

Assignment Objective: To learn the event handling using JavaScript

Assignment Outcome: Able to make use of it to provide validation to commercial websites

Description:

JavaScript is the programming language of HTML and the Web.

Is JAVA and JavaScript same?

JavaScript and Java are completely different languages, both in concept and design.

JavaScript was invented by **Brendan Eich** in 1995, and became an ECMA standard in 1997. ECMA-262 is the official name of the standard. ECMAScript is the official name of the language.

Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle) and released in 1995 as a core component of Sun Microsystems' Java platform.

Why Study JavaScript?

JavaScript is one of the 3 languages all web developers must learn:

1. HTML to define the content of web pages
2. CSS to specify the layout of web pages
3. JavaScript to program the behavior of web pages

What can JavaScript Do? - **Event Handling**

Event handlers can be used to handle, and verify, user input, user actions, and browser actions:

- Things that should be done every time a page loads
- Content that should be verified when a user inputs data
- Action that should be performed when a user clicks a button
- Things that should be done when a page to be closed
- And more ...

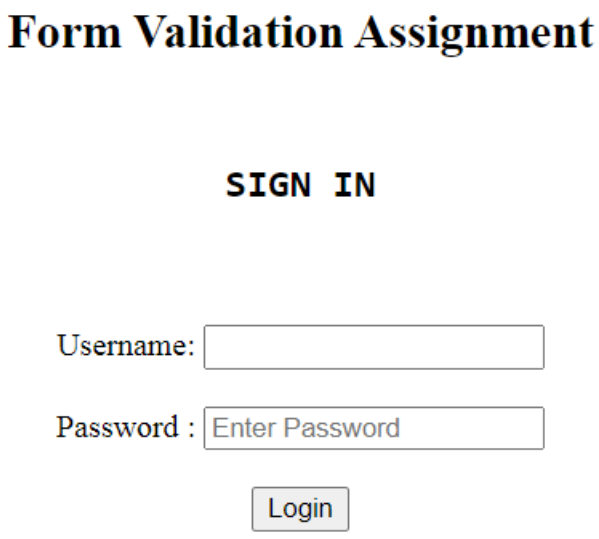
Many different methods can be used to let JavaScript work with events:

- HTML event attributes can call JavaScript functions
- HTML event attributes can execute JavaScript code directly
- You can assign your own event handler functions to HTML elements
- You can prevent events from being sent or being handled
- And more ...

Sample Supporting Code for Validation

```
</html>
<body>
<div class=vf>
<h2>Form Validation Assignment</h2>
<pre>
<h2>SIGN IN</h2>
</pre>
<form name="loginform" onsubmit="return validateForm()" action="index.htm" method="post">
 <label for="username">Username:</label>
 <input type="text" placeholder="Enter Username" id="username" name="username" required>
 <br><br>
 <label for="pwd">Password :</label>
 <input type="password"  placeholder="Enter Password" id="pwd" name="pwd" required>
 <br><br>
 <input type="submit" value="Login">
 </form>

</div>
</body>
```

Sample Output



Experiment Number: 04

CO Mapping	PO Mapping
CO1, CO2 and CO3	P02,PO3,PO4,PO5

Assignment Objective: To learn how to make responsive websites and have a video at background

Assignment Outcome: Able to build responsive websites as well to add video at background

Problem Statement: Perform the validation for website develop in lab assignment number 3 using JavaScript, make it responsive and play the video at background

Description

Following is the logic and syntax for making page responsive using media query

 # Meta Configuration for responsiveness

```
<meta name="viewport" content="width=device-width,initial-scale=1,maximum-scale=1,user-scalable=no">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<meta name="HandheldFriendly" content="true">
```

 # Media Query Configuration css file

```
/* Media Queries: Tablet Landscape */
@media screen and (max-width: 1060px) {
  #primary { width:67%; }
  #secondary { width:30%; margin-left:3%;}
}
/* Media Queries: Tabled Portrait */
@media screen and (max-width: 768px) {
  #primary { width:100%; }
  #secondary { width:100%; margin:0; border:none; }
```

 # Video Container Configuration

```
img{ max-width: 100%; height: auto; }
.video-container {
        position: relative;
        padding-bottom: 56.25%;
        padding-top: 30px;
        height: 0;
        overflow: hidden;
}
.video-container iframe,
.video-container object,
.video-container embed {
        position: absolute;
        top: 0;
        left: 0;
        width: 100%;
        height: 100%;
}
```

```
@media (min-width: 640px) { body {font-size:1rem;} }
 @media (min-width:960px) { body {font-size:1.2rem;} }
 @media (min-width:1100px) { body {font-size:1.5rem;} }
```

#Sample Code for video tag

```
<header class="v-header container">
   <div class="fullscreen-video-wrap">
   <video src="durga.mp4" autoplay="true" loop="true" muted></video>
   </div>
   <div class="header-overlay"></div>
   <div class="header-content">
   <h2>Curriculum Vitae</h2>
   <p>Disha Dhore </p>
   <p>Pune, India </p>
    <a class="btn">For More Details Scroll Down </a>
   </div>
 </header>
```

 # Configuration for Multi column Layouts for responsive website

How TO - Three Column Layout

```
<div class="row">
 <div class="column"></div>
 <div class="column"></div>
 <div class="column"></div>
</div>
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
 box-sizing: border-box;
}
/* Create three equal columns that floats next to each other */
.column {
 float: left;
 width: 33.33%;
 padding: 10px;
 height: 300px; /* Should be removed. Only for demonstration */
}
/* Clear floats after the columns */
.row:after {
 content: "";
 display: table;
clear : both;
}
```

 # Applying responsiveness to the embedded frames

Also let videos or slideshows scale properly on any device.

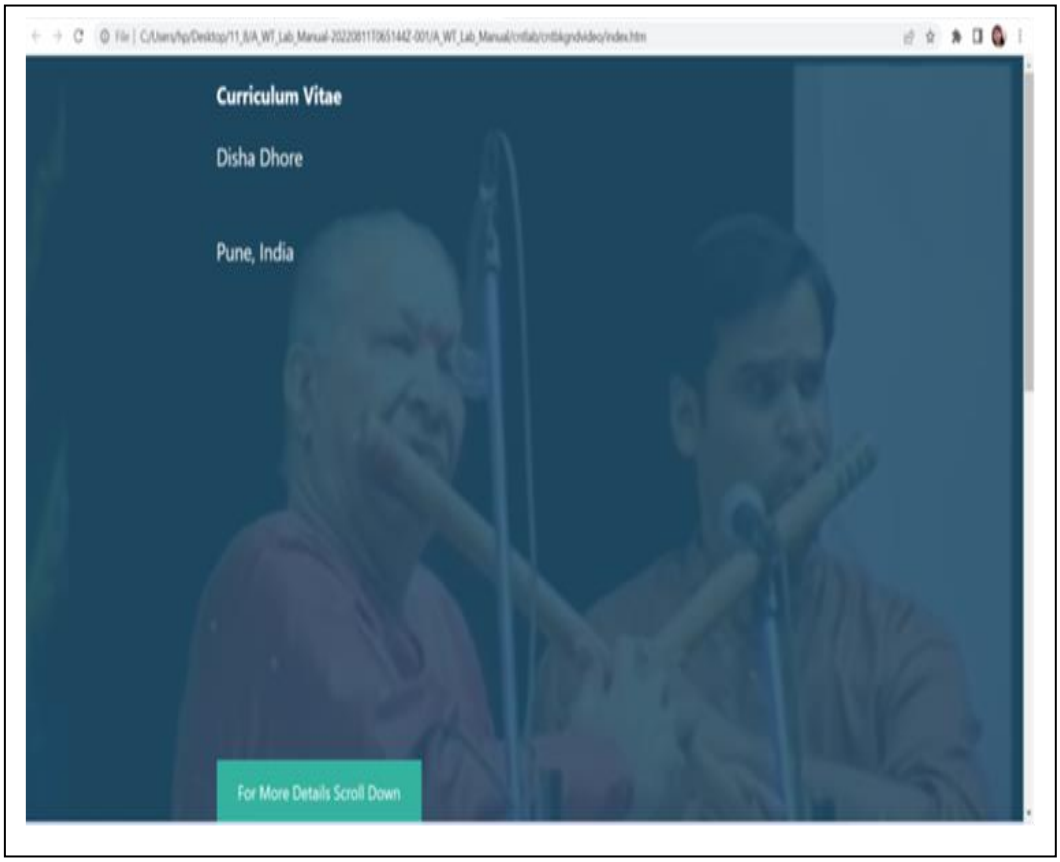
Classes can be applied directly to <iframe>, <embed>, <video>, and <object> elements.

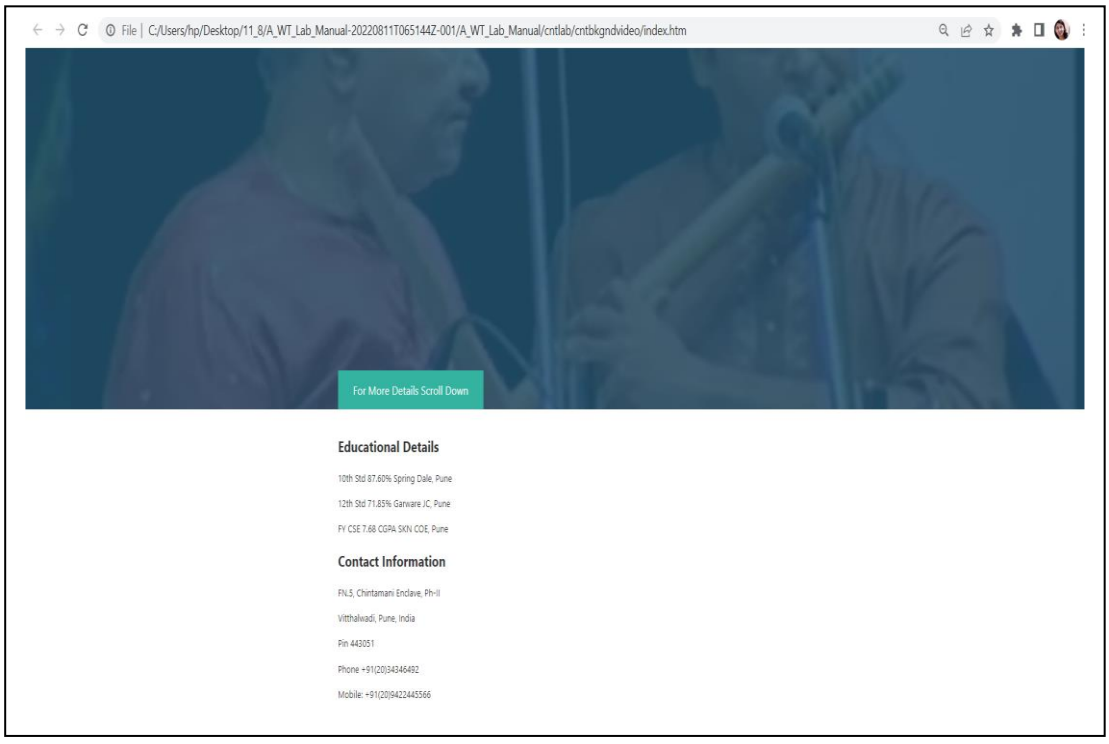
The following example creates a responsive video by adding an .embed-responsive-item class to an <iframe> tag (the video will then scale nicely to the parent element). The containing defines the aspect ratio of the video:

Example

```
<div class="embed-responsive embed-responsive-16by9">
 <iframe class="embed-responsive-item" src="..."></iframe>
</div>
```

Sample Output for Responsive Website


Curriculum Vitae
Disha Dhore
Pune, India
For More Details Scroll Down


File | C:/Users/hp/Desktop/11_8/A_WT_Lab_Manual-20220811T065144Z-001/A_WT_Lab_Manual/cntlab/cntbkgndvideo/index.htm
For More Details Scroll Down
Educational Details
10th Std 87.60% Spring Dale, Pune
12th Std 71.85% Garware JC, Pune
FY CSE 7.68 CGPA SKN COE, Pune
Contact Information
FN-5, Chintamani Enclave, Ph-II
Vitthalwadi, Pune, India
Pin 443051
Phone +91(20)34346492
Mobile: +91(20)9422445566

Experiment Number: 5

CO Mapping	PO Mapping
CO1, CO2 and CO3	P02,PO3,PO4,PO5

Assignment Objective: To learn how to make use of JavaScript to implement data structures for interactive websites

Assignment Outcome: Able use of JavaScript to implement data structures for interactive websites

Problem Statement: Develop a website using JavaScript program to implement array operations such as reverse the elements of a given array / sorting methods/ searching methods.

Description:
Following are the built in functions for sorting an element of an array. But for this assignment you are no permitted to use in built in functions.

Sorting an Array

The sort() method sorts an array alphabetically:

Example

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.sort();       // Sorts the elements of fruits
```

Reversing an Array

The reverse() method reverses the elements in an array.

You can use it to sort an array in descending order:

Example

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.sort();       // First sort the elements of fruits
fruits.reverse();    // Then reverse the order of the element
```

Numeric Sort

By default, the sort() function sorts values as **strings**.

However, if numbers are sorted as strings, "25" is bigger than "100", because "2" is bigger than "1".

Example

```
var points                                  =                   [40, 100, 1, 5, 25, 10];
points.sort(function(a, b){return a - b});
```

Example

```
var points                                  =                   [40, 100, 1, 5, 25, 10];
points.sort(function(a, b){return b - a});
```

Array.map()

Array.filter()

Array.reduce()

Array.reduceRight()

Array.every()

Array.some()

Array.indexOf()

Array.lastIndexOf()

Array.find()

Array.findIndex()

Sample Output:

Experiment Number: 06

CO Mapping	PO Mapping
CO1, CO2 and CO3	P02,PO3,PO4,PO5

Assignment Objective: To learn how to make use of multi-layouts, tabs, pills to build websites

Assignment Outcome: Able to build websites using of multi-layouts, tabs, pills to build websites

Problem Statement: . Develop a website using toggle able or dynamic tabs or pills with bootstrap and JQuery

Description:

Bootstrap Tabs and Pills

Menus
Most web pages have some kind of a menu.

In HTML, a menu is often defined in an unordered list <ul> (and styled afterwards), like this:

```
<ul>
<li><a href="#">Home</a></li>
<li><a href="#">Menu 1</a></li>
<li><a href="#">Menu 2</a></li>
<li><a href="#">Menu 3</a></li>
</ul>
```

If you want to create a horizontal menu of the list above, add the .list-inline class to <ul>:

```
<ul class="list-inline">
```

abs
Home
Menu 1
Menu 2
Menu 3

Tabs are created with <ul class="nav nav-tabs">:

Tip: Also mark the current page with <li class="active">.

The following example creates navigation tabs:

Example

```
<ul class="nav nav-tabs">
<li class="active"><a href="#">Home</a></li>
<li><a href="#">Menu 1</a></li>
```

```
<li><a href="#">Menu 2</a></li>
<li><a href="#">Menu 3</a></li>
</ul>
```

Tabs With Dropdown Menu

Home
Menu 1
Menu 2
Menu 3
Tabs can also hold dropdown menus.

The following example adds a dropdown menu to "Menu 1":

Example

```
<ul class="nav nav-tabs">
<li class="active"><a href="#">Home</a></li>
<li class="dropdown">
<a class="dropdown-toggle" data-toggle="dropdown" href="#">Menu 1
<span class="caret"></span></a>
<ul class="dropdown-menu">
<li><a href="#">Submenu 1-1</a></li>
<li><a href="#">Submenu 1-2</a></li>
<li><a href="#">Submenu 1-3</a></li>
</ul>
</li>
<li><a href="#">Menu 2</a></li>
<li><a href="#">Menu 3</a></li>
</ul>
```

Pills

Home
Menu 1
Menu 2
Menu 3
Pills are created with <ul class="nav nav-pills">. Also mark the current page with <li class="active">:

Example

```
<ul class="nav nav-pills">
<li class="active"><a href="#">Home</a></li>
<li><a href="#">Menu 1</a></li>
<li><a href="#">Menu 2</a></li>
<li><a href="#">Menu 3</a></li>
</ul>
```

Sample Output:

Experiment Number: 08

CO Mapping	PO Mapping
CO1, CO2 and CO3	P02,PO3,PO4,PO5

Assignment Objective: To learn how to build three tier websites using scripts at front and server side as well

Assignment Outcome: Able to build three tier websites using scripts at frond and server side

Problem Statement:. Create three MySQL database tables and write PHP scripts to read, insert & delete data through web interface.

Description:

What is a PHP File?
PHP files can contain text, HTML, CSS, JavaScript, and PHP code
PHP code is executed on the server, and the result is returned to the browser as plain HTML
PHP files have extension ".php"

What Can PHP Do?
PHP can generate dynamic page content
PHP can create, open, read, write, delete, and close files on the server
PHP can collect form data
PHP can send and receive cookies
PHP can add, delete, and modify data in your database
PHP can be used to control user-access
PHP can encrypt data
With PHP you are not limited to output HTML. You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.

Why PHP?
PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
PHP is compatible with almost all servers used today (Apache, IIS, etc.)
PHP supports a wide range of databases
PHP is free. Download it from the official PHP resource: www.php.net
PHP is easy to learn and runs efficiently on the server side

PHP Case Sensitivity

In PHP, keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are not case-sensitive.

PHP Global Variables - Superglobals

Some predefined variables in PHP are "superglobals", which means that they are always accessible, regardless of scope - and you can access them from any function, class or file without having to do anything special.

he PHP superglobal variables are:

- $GLOBALS
- $_SERVER
- $_REQUEST
- $_POST
- $_GET
- $_FILES
- $_ENV
- $_COOKIE
- $_SESSION

$GLOBALS - $GLOBALS is a PHP super global variable which is used to access global variables from anywhere in the PHP script (also from within functions or methods).

PHP stores all global variables in an array called $GLOBALS[*index*]. The *index* holds the name of the variable.

```
<?php
$x = 75;
$y = 25;
function addition() {
  $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}
 addition();
echo $z;
?>
```

PHP $_SERVER

$_SERVER is a PHP super global variable which holds information about headers, paths, and script locations.

The example below shows how to use some of the elements in $_SERVER:

Example

```
<?php
echo $_SERVER['PHP_SELF'];              o/p- /demo/demo_global_server.php
echo "<br>";
echo $_SERVER['SERVER_NAME'];    - 35.194.26.41
echo "<br>";
echo $_SERVER['HTTP_HOST'];      -35.194.26.41
echo "<br>";
echo $_SERVER['HTTP_REFERER'];  -https://tryphp.w3schools.com/showphp.php? echo "<br>";
echo $_SERVER['HTTP_USER_AGENT']; - AppleWebKit/537.36 (KHTML, like Gecko)
```

```
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];  -/demo/demo_global_server.php
?>
```

What is HTTP?

The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers.

HTTP works as a request-response protocol between a client and server.

Example: A client (browser) sends an HTTP request to the server; then the server returns a response to the client. The response contains status information about the request and may also contain the requested content.

HTTP Methods

- **GET**
- **POST**
- **PUT**
- **HEAD**
- **DELETE**
- **PATCH**
- **OPTIONS**

The two most common HTTP methods are: GET and POST.

The GET Method

GET is used to request data from a specified resource.

GET is one of the most common HTTP methods.

Note that the query string (name/value pairs) is sent in the URL of a GET request:

/test/demo_form.php?name1=value1&name2=value2

Some other notes on GET requests:

- GET requests can be cached
- GET requests remain in the browser history
- GET requests can be bookmarked
- GET requests should never be used when dealing with sensitive data
- GET requests have length restrictions
- GET requests are only used to request data (not modify)

The POST Method

POST is used to send data to a server to create/update a resource.

The data sent to the server with POST is stored in the request body of the HTTP request:

```
POST /test/demo_form.php HTTP/1.1
Host: w3schools.com
name1=value1&name2=value2
```

POST is one of the most common HTTP methods.

Some other notes on POST requests:

- POST requests are never cached
- POST requests do not remain in the browser history
- POST requests cannot be bookmarked
- POST requests have no restrictions on data length

PHP $_REQUEST

PHP $_REQUEST is a PHP super Global variable which is used to collect data after submitting an HTML form.

```
<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  // collect value of input field
  $name = $_REQUEST['fname'];
  if (empty($name)) {
    echo "Name is empty";
  } else {
    echo $name;
  }
}
?>
```

PHP $_POST

$_POST is an array of variables passed to the current script via the HTTP POST method.

PHP $_POST is a PHP super global variable which is used to collect form data after submitting an HTML form with method="post". $_POST is also widely used to pass variables.

```
<html>

<body>

<form method="post" action="collect.php">
```

```
 Name: <input type="text" name="fname">

 Middle Name: <input type="text" name="mname">

 Last Name: <input type="text" name="lname">

<input type="submit">

</form>

</body>
</html>
```

$_GET is an array of variables passed to the current script via the URL parameters.

PHP Form Handling

PHP - A Simple HTML Form

```
<form action="welcome.php" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>
```

When the user fills out the form above and clicks the submit button, the form data is sent for processing to a PHP file named "welcome.php". The form data is sent with the HTTP POST method.

```
<?php
if( $_POST["name"] || $_POST["email"] ) {
if (preg_match("/[^A-Za-z'-]/",$_POST['name'] )) {
die ("invalid name and name should be alpha");
}
echo "Welcome ". $_POST['name']."<br />";
echo "You mail id is". $_POST['email']. "<br />";
exit();
}
?>
```

PHP Form Handling – GET Method

```
<html>
<body>
<form action="welcome1.php" method="get">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>
</body>
</html>
<?php
if( $_GET["name"] || $_GET["email"] ) {
if (preg_match("/[^A-Za-z'-]/",$_GET['name'] )) {
```

```
die ("invalid name and name should be alpha");
}
echo "Welcome ". $_GET['name']."<br />";
echo "You mail id is". $_GET['email']. "<br />";
exit();
}
?>
```

Developers prefer POST for sending form data.

For Security

1. Strip unnecessary characters (extra space, tab, newline) from the user input data (with the PHP trim() function)
2. Remove backslashes (\) from the user input data (with the PHP stripslashes() function)
3. Use htmlspecialchars() function converts special characters to HTML entities. This means that it will replace HTML characters like < and > with < and >. This prevents attackers from exploiting the code by injecting HTML or Javascript code (Cross-site Scripting attacks) in forms.

```
<?php
// define variables and set to empty values
$name = $email = $gender = $comment = $website = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  $name = test_input($_POST["name"]);
  $email = test_input($_POST["email"]);
  $website = test_input($_POST["website"]);
  $comment = test_input($_POST["comment"]);
  $gender = test_input($_POST["gender"]);
}
function test_input($data) {
  $data = trim($data);
  $data = stripslashes($data);
  $data = htmlspecialchars($data);
  return $data;
}
?>
```

MySQL Database

What is MySQL?

- MySQL is a database system used on the web
- MySQL is a database system that runs on a server
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, and easy to use
- MySQL uses standard SQL
- MySQL compiles on a number of platforms
- MySQL is free to download and use
- MySQL is developed, distributed, and supported by Oracle Corporation

- MySQL is named after co-founder Monty Widenius's daughter: My

PHP + MySQL Database System

- PHP combined with MySQL are cross-platform - **you can develop in Windows and serve on a Unix platform**

PHP Connect to MySQL

PHP 5 and later can work with a MySQL database using:

- MySQL extension (deprecated in 2012)
- MySQLi extension (the "i" stands for improved)
- PDO (PHP Data Objects)

Should I Use MySQLi or PDO?

- If you need a short answer, it would be "**Whatever you like**".
- **PDO will work on 12 different database systems**, whereas MySQLi will only work with MySQL databases.
- So, if you have to switch your project to use another database, PDO makes the process easy. You only have to change the connection string and a few queries. With MySQLi, you will need to rewrite the entire code - queries included.
- Both are **object-oriented, but MySQLi also offers a procedural API.**
- Both support Prepared Statements - Prepared Statements protect from SQL injection, and are very important for web application security.

Example (MySQLi Procedural)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
// Create connection
$conn = mysqli_connect($servername, $username, $password);
// Check connection
if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";
?>
```

Example (MySQLi Object-Oriented)

```
<?php
$servername = "localhost";
```

```
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);

// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

Example (PDO)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

try {
  $conn = new PDO("mysql:host=$servername;dbname=myDB", $username, $password);
  // set the PDO error mode to exception
  $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
  echo "Connected successfully";
} catch(PDOException $e) {
  echo "Connection failed: " . $e->getMessage();
}
?>
```

Close the Connection

MySQLi Procedural:**mysqli_close($conn);**

MySQLi Object-Oriented:**$conn->close();**

PDO:**$conn = null;**

Create a MySQL Database Using MySQLi (OOP)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
// Create database
$sql = "CREATE DATABASE myDB";
```

```
if ($conn->query($sql) === TRUE) {
  echo "Database created successfully";
} else {
  echo "Error creating database: " . $conn->error;
}

?>
```

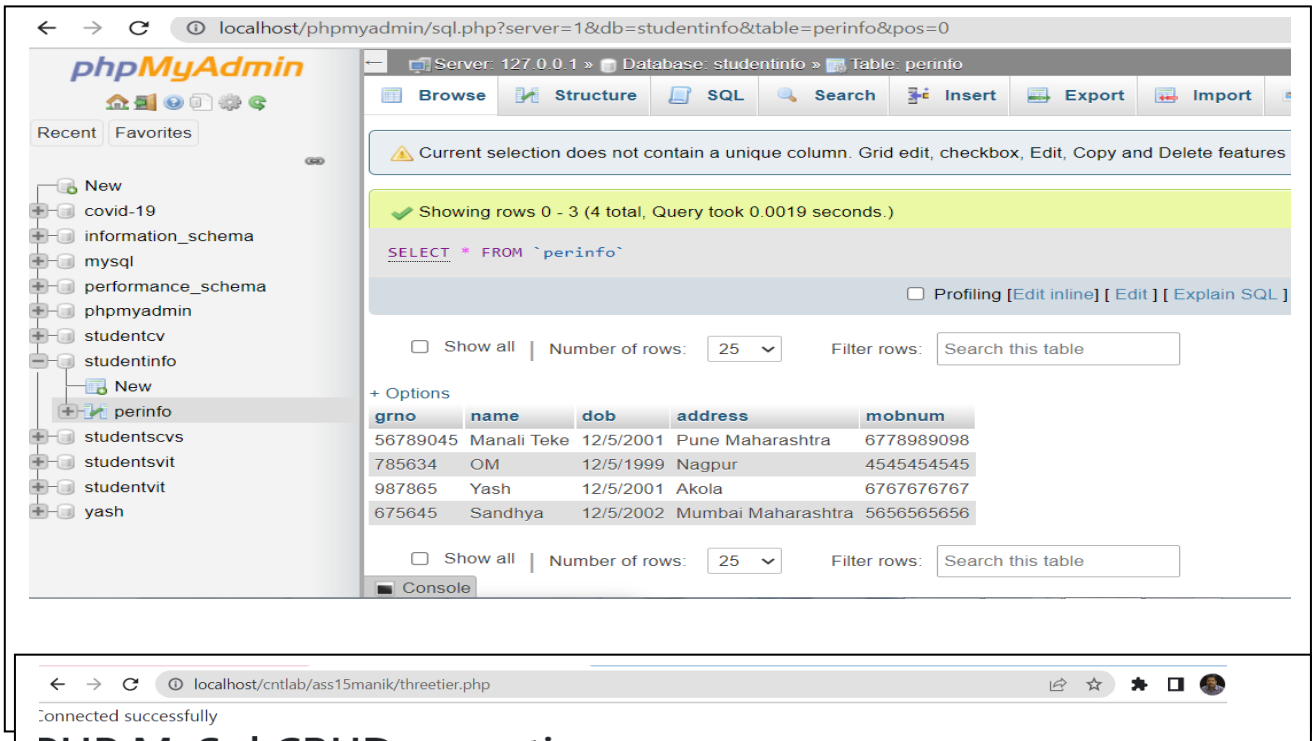
Create a MySQL Database Using MySQLi (Procedural)

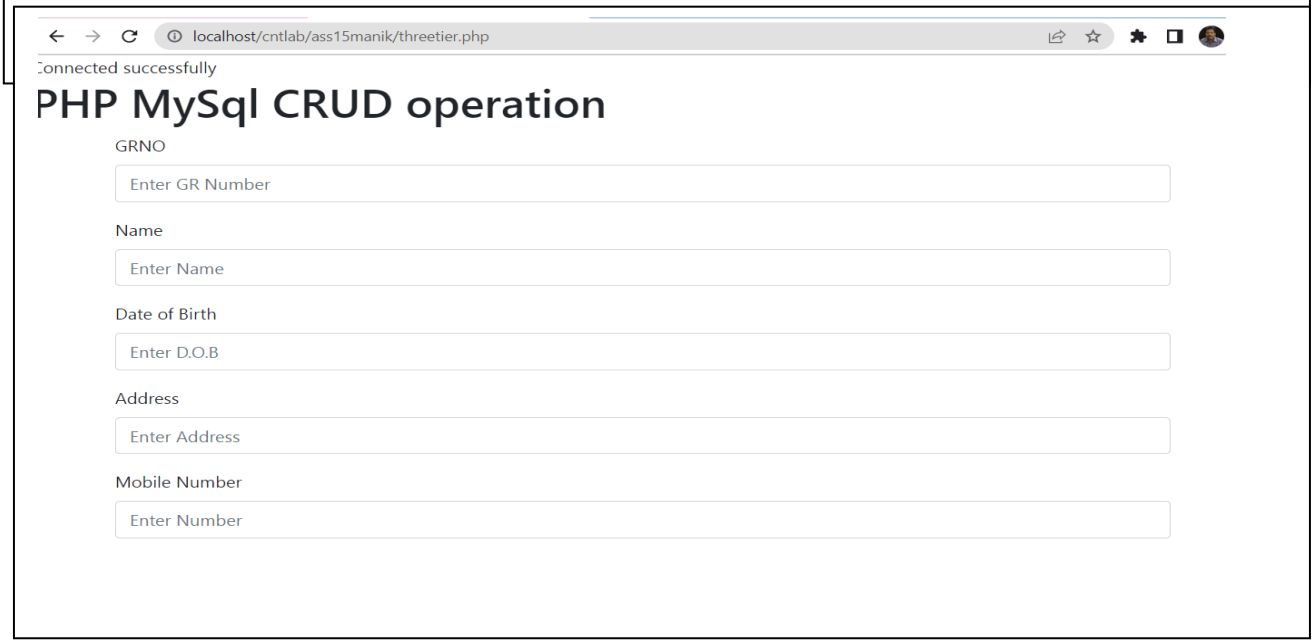
```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
// Create connection
$conn = mysqli_connect($servername, $username, $password);
// Check connection
if (!$conn) {
  die("Connection failed: " . mysqli_connect_error());
}
// Create database
$sql = "CREATE DATABASE myDB";
if (mysqli_query($conn, $sql)) {
  echo "Database created successfully";
} else {
  echo "Error creating database: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

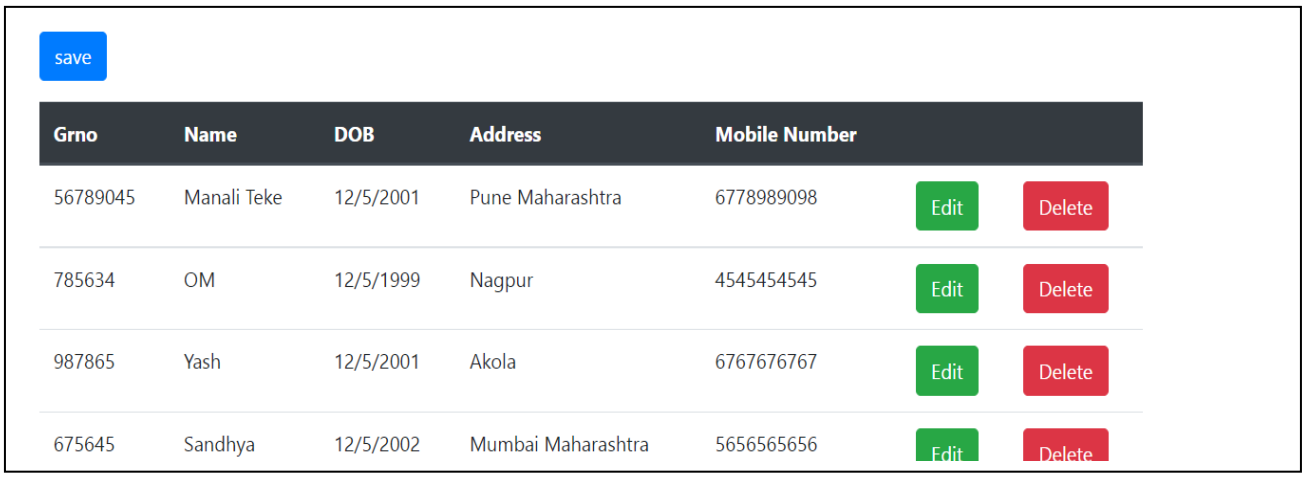
Create a MySQL Database Using PDO

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

try {
  $conn = new PDO("mysql:host=$servername", $username, $password);
  // set the PDO error mode to exception
  $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
  $sql = "CREATE DATABASE myDBPDO";
  // use exec() because no results are returned
  $conn->exec($sql);
  echo "Database created successfully<br>";
} catch(PDOException $e) {
  echo $sql . "<br>" . $e->getMessage();
}

$conn = null;
?>
```

Sample Outputs:







Experiment Number: 09

CO Mapping	PO Mapping
CO1, CO2 and CO3	P02,PO3,PO4,PO5

Assignment Objective: To learn how to build three tier websites using scripts at front and server side as well

Assignment Outcome: Able to build three tier websites using scripts at frond and server side

Problem Statement: Write a program to calculate Electricity bill in PHP

(a) You need to write a PHP program to calculate electricity bill using if-else conditions.

(b) Conditions are :

For first 50 units – Rs. 3.50/unit

For next 100 units – Rs. 4.00/unit

For next 100 units – Rs. 5.20/unit

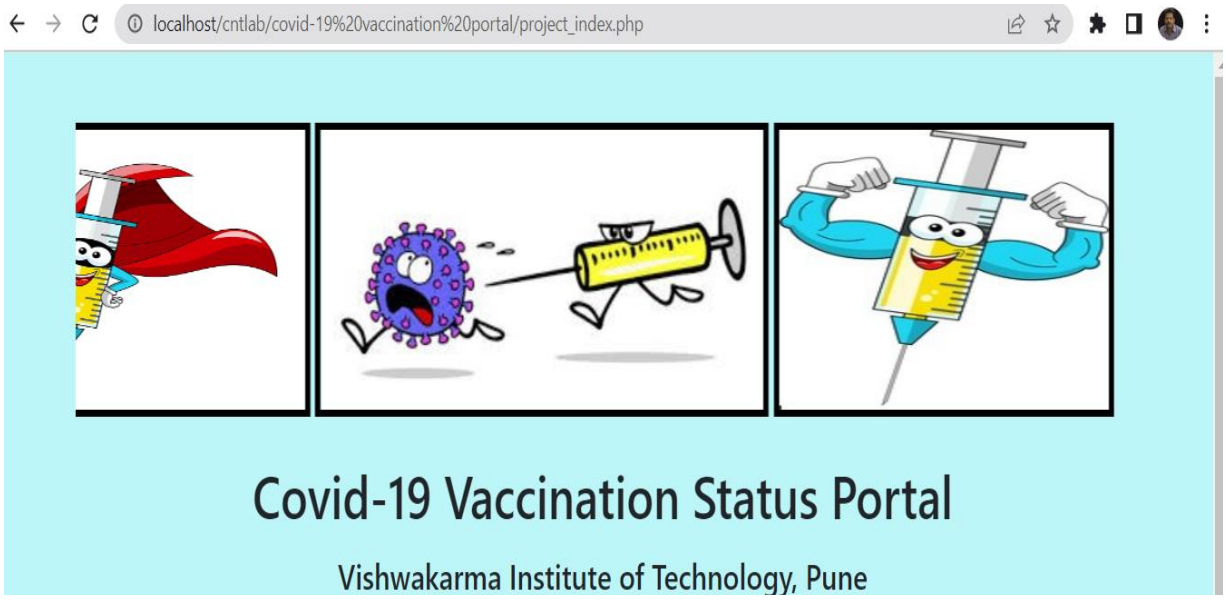
For units above 250 – Rs. 6.50/unit

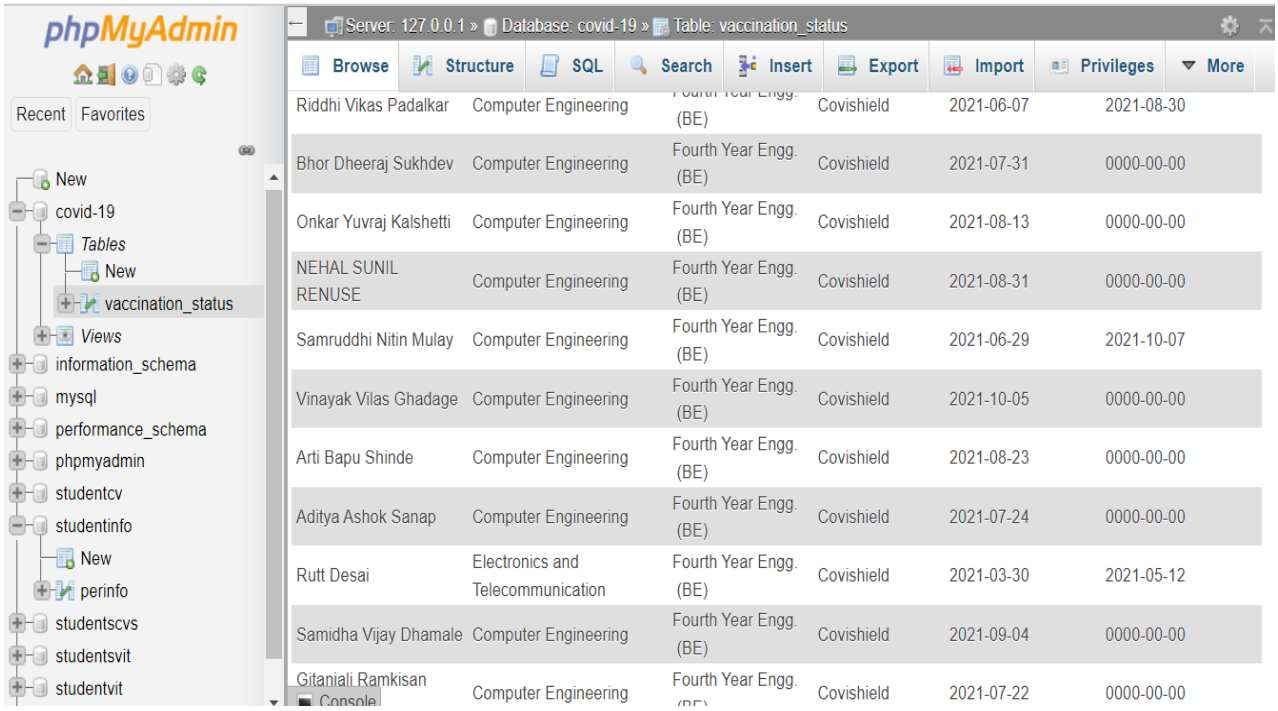
Or

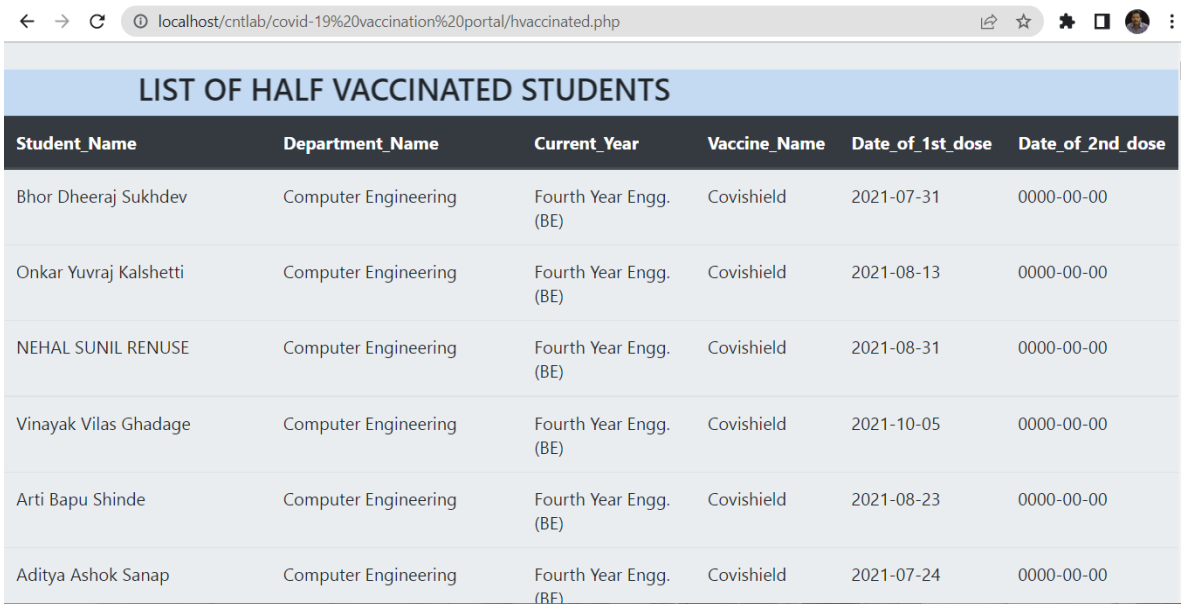
Develop a COVID-19 Vaccination Status Portal.

Collect the Database from College Office

Sample Outputs:




phpMyAdmin
Recent
Favorites
New
covid-19
Tables
New
vaccination_status
Views
information_schema
mysql
performance_schema
phpmyadmin
studentcv
studentinfo
New
perinfo
studentscvs
studentsvit
studentvit
Server: 127.0.0.1 » Database: covid-19 » Table: vaccination_status
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
More
Riddhi Vikas Padalkar Computer Engineering (BE) Covishield 2021-06-07 2021-08-30
Bhor Dheeraj Sukhdev Computer Engineering Fourth Year Engg. (BE) Covishield 2021-07-31 0000-00-00
Onkar Yuvraj Kalshetti Computer Engineering Fourth Year Engg. (BE) Covishield 2021-08-13 0000-00-00
NEHAL SUNIL RENUSE Computer Engineering Fourth Year Engg. (BE) Covishield 2021-08-31 0000-00-00
Samruddhi Nitin Mulay Computer Engineering Fourth Year Engg. (BE) Covishield 2021-06-29 2021-10-07
Vinayak Vilas Ghadage Computer Engineering Fourth Year Engg. (BE) Covishield 2021-10-05 0000-00-00
Arti Bapu Shinde Computer Engineering Fourth Year Engg. (BE) Covishield 2021-08-23 0000-00-00
Aditya Ashok Sanap Computer Engineering Fourth Year Engg. (BE) Covishield 2021-07-24 0000-00-00
Rutt Desai Electronics and Telecommunication Fourth Year Engg. (BE) Covishield 2021-03-30 2021-05-12
Samidha Vijay Dhamale Computer Engineering Fourth Year Engg. (BE) Covishield 2021-09-04 0000-00-00
Gitanjali Ramkisan Computer Engineering Fourth Year Engg. Covishield 2021-07-22 0000-00-00
Console


localhost/cntlab/covid-19%20vaccination%20portal/hvaccinated.php
LIST OF HALF VACCINATED STUDENTS
Student_Name Department_Name Current_Year Vaccine_Name Date_of_1st_dose Date_of_2nd_dose
Bhor Dheeraj Sukhdev Computer Engineering Fourth Year Engg. (BE) Covishield 2021-07-31 0000-00-00
Onkar Yuvraj Kalshetti Computer Engineering Fourth Year Engg. (BE) Covishield 2021-08-13 0000-00-00
NEHAL SUNIL RENUSE Computer Engineering Fourth Year Engg. (BE) Covishield 2021-08-31 0000-00-00
Vinayak Vilas Ghadage Computer Engineering Fourth Year Engg. (BE) Covishield 2021-10-05 0000-00-00
Arti Bapu Shinde Computer Engineering Fourth Year Engg. (BE) Covishield 2021-08-23 0000-00-00
Aditya Ashok Sanap Computer Engineering Fourth Year Engg. (BE) Covishield 2021-07-24 0000-00-00

Experiment Number: 10/11/12/13

Experiment	CO Mapping	PO Mapping
10	CO1, CO2,CO4 and CO5	P02,PO3,PO5,PO6
11,12	CO1, CO4 and CO5	P02,PO5,PO11
13	CO1, CO2,CO4 and CO5	P02,PO5,PO6,PO11, PSO3

Assignment Objective: To learn how to build scalable enterprise three tier websites using REACT at front end ,REST API, Node JS and Spring BOOT at Server Side and MySQL/Oracle at backend.

Assignment Outcome: Able to build three tier websites using react at front, REST API, Node JS and Spring BOOT at Server Side and MySQL/Oracle at backend.

Problem Statement: Design and implement a website using REST API and Spring Boot.

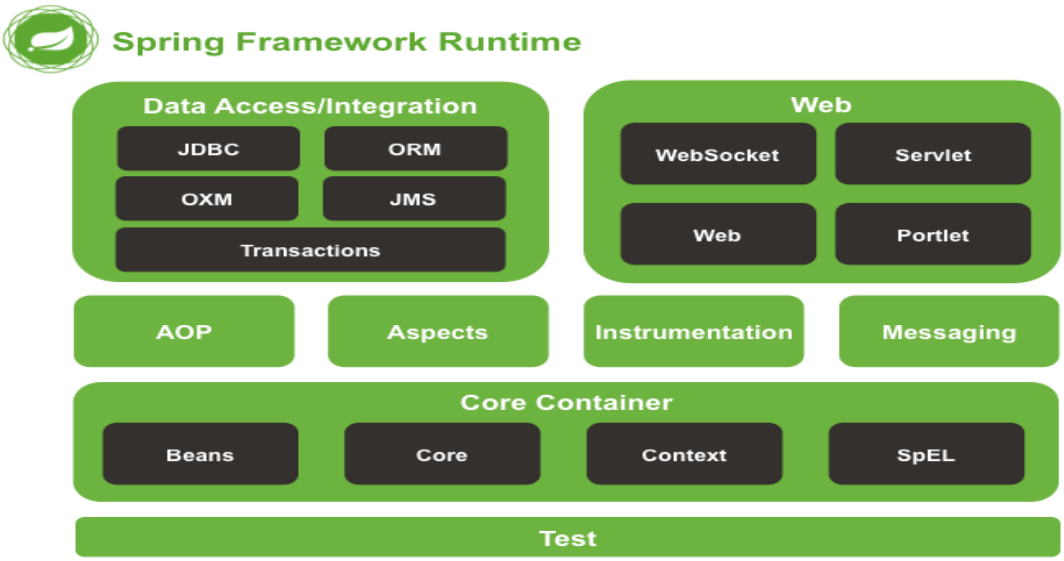
Problem Statement: Design and implement a website using REST API, Spring Boot and MySQL/Oracle

Problem Statement: Design and implement a website using REACT, Spring Boot and MySQL/Oracle

Problem Statement: Design and implement a **ENTERPRISE** website using REACT, Node JS and MySQL/Oracle

Description:

Spring allows you:
--Make a Java method execute in a database transaction without dealing with transaction APIs.
--Make a local Java method a remote procedure without having to deal with remote APIs.
--Make a local Java method a management operation without having to deal with JMX APIs.
--Make a local Java method a message handler without having to deal with JMS APIs.
-- The Spring Framework Inversion of Control (IoC) component provides a formalized means of composing disparate components into a fully working application ready for use.



Core Container: consists of the spring-core, spring-beans, spring-ontext, spring-context-support, and spring-expression (Spring Expression Language) modules.

The spring-core and spring-beans modules provide the fundamental parts of the framework, including the IoC and Dependency Injection features.

The BeanFactory is a sophisticated implementation of the factory pattern.

Context module inherits its features from the Beans module and adds support for internationalization, event propagation, resource loading, a Servlet container,. Java EE features such as EJB, JMX, and basic remoting, Application Context interface, integrating common third-party libraries into a Spring application context

Spring-expression module provides a powerful Expression Language for querying and manipulating an object graph at runtime. It is an extension of the unified expression language (unified EL) as specified in the JSP 2.1 specification. The language supports setting and getting property values, property assignment, method invocation, accessing the content of arrays, collections and indexers, logical and arithmetic operators, named variables, and retrieval of objects by name from Spring's IoC container. It also supports list projection and selection as well as common list aggregations.

AOP Alliance-compliant aspect-oriented programming implementation allowing you to define, for example, method interceptors and point cuts to cleanly decouple code that implements functionality that should be separated.

Aspects module provides integration with AspectJ.

Instrument module provides class instrumentation support and class loader implementations to be used in certain application servers.

Messaging module with key abstractions from the Spring Integration project such as Message, Message Channel, Message Handler, and others to serve as a foundation for messaging-based applications. The module also includes a set of annotations for mapping messages to methods, similar to the Spring MVC annotation based programming model.

Data Access/Integration Layer: consists of the JDBC, ORM, OXM, JMS, and Transaction modules.
Spring-JDBC module provides a JDBC-abstraction layer that removes the need to do tedious JDBC coding and parsing of database-vendor specific error codes.
Spring-TX module supports programmatic and declarative transaction management for classes that implement special interfaces and for all your POJOs (Plain Old Java Objects).
Spring-ORM module provides integration layers for popular object-relational mapping APIs, including JPA and Hibernate.
Spring-OXM module provides an abstraction layer that supports Object/XML mapping implementations such as JAXB, Castor, JiBX and XStream.
Spring-jms module (Java Messaging Service) contains features for producing and consuming messages.

Web Layer: consists of the spring-web, spring-webmvc and spring-websocket modules.
Spring-web module provides basic web-oriented integration features such as multipart file upload functionality and the initialization of the IoC container using Servlet listeners and a web-oriented application context. It also contains an HTTP client and the web-related parts of Spring's remoting support.
Spring-webmvc module (also known as the Web-Servlet module) contains Spring's model-view-controller (MVC) and REST Web Services implementation for web applications. Spring's MVC framework provides a clean separation between domain model code and web forms and integrates with all of the other features of the Spring Framework.

The spring-test module supports the unit testing and integration testing of Spring components with JUnit or TestNG. It provides consistent loading of Spring ApplicationContexts and caching of those contexts. It also provides mock objects that you can use to test your code in isolation.

What is Spring Boot
--Spring Boot is an open source Java-based framework
-- is built on top of the conventional spring framework, so, it provides all the features of spring and is yet easier to use than spring
-- Spring Boot is a micro service-based framework and making a production-ready application in very less time.
-- In Spring Boot everything is auto-configured. We just need to use proper configuration for utilizing a particular functionality.
-- Spring Boot is very useful if we want to develop REST API
--developed by Pivotal Team

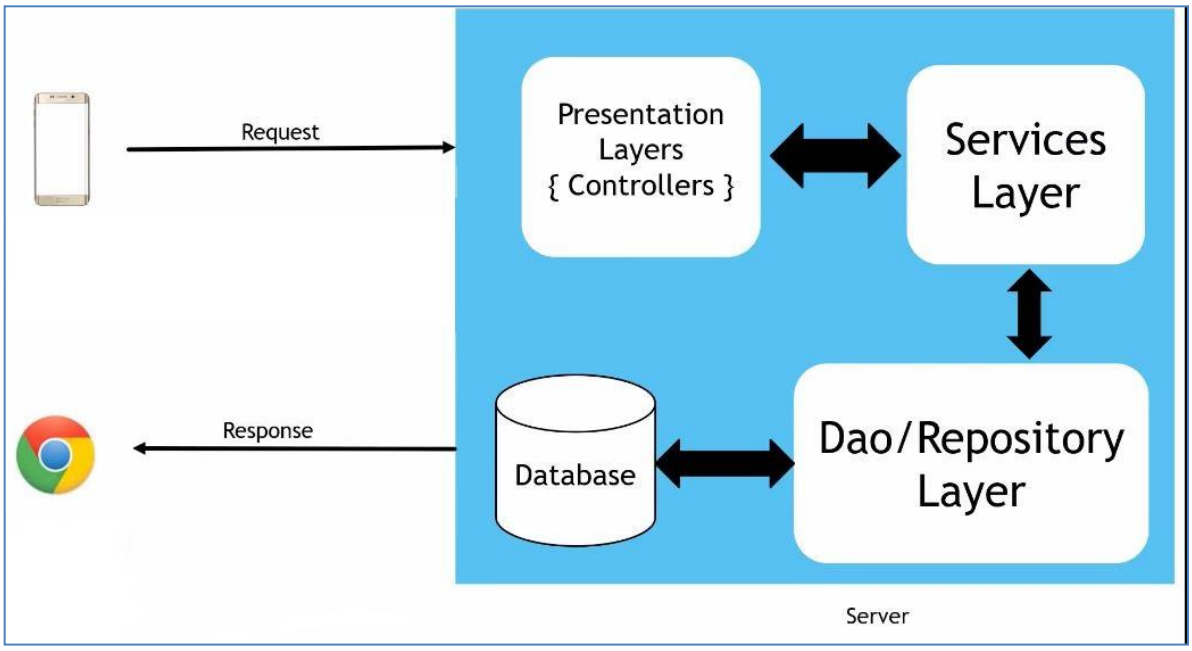
<table>
<tr><th>Client</th><td rowspan="2">----->
Request

<-----
Reply</td><th colspan="3">Server</th><td rowspan="2"><-----
Request

<-----
Reply</td><th>Database</th></tr>
<tr><td rowspan="2">Browser - Chrome, Mozilla,
Mobile Phone,
Laptop, Desktop

Postman</td><td>Presentation layer
Controller uses</td><td>Service Layer</td><td>DAO Layer</td><td>Repository layer</td></tr>
<tr><td>To accept Request
(What client wants)</td><td>Business Logic
(Classess, Methods)</td><td>Database Connectivity</td><td>Database

Oracle, MySQL</td></tr>
</table>



Request ---> Controller --> Controller uses Services from Service layer -->DAO---> Database

---

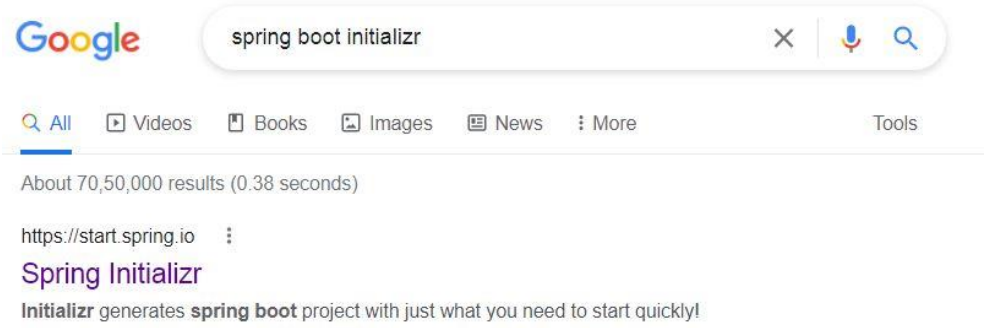
From hands on perspective...how to send or fire a request.....
Use the software Tool Postman - To fire request
What is POSTMAN Tool?
Postman is an interactive and automatic tool for verifying the APIs of your project. Postman is a Google Chrome app for interacting with HTTP APIs. It presents you with a friendly GUI for constructing requests and reading responses.

Request Method	Response
GET	Get one or multiple records from database
POST	Add one or more records in database
PUT	Update one or more records in database
DELETE	Delete one or more records from database

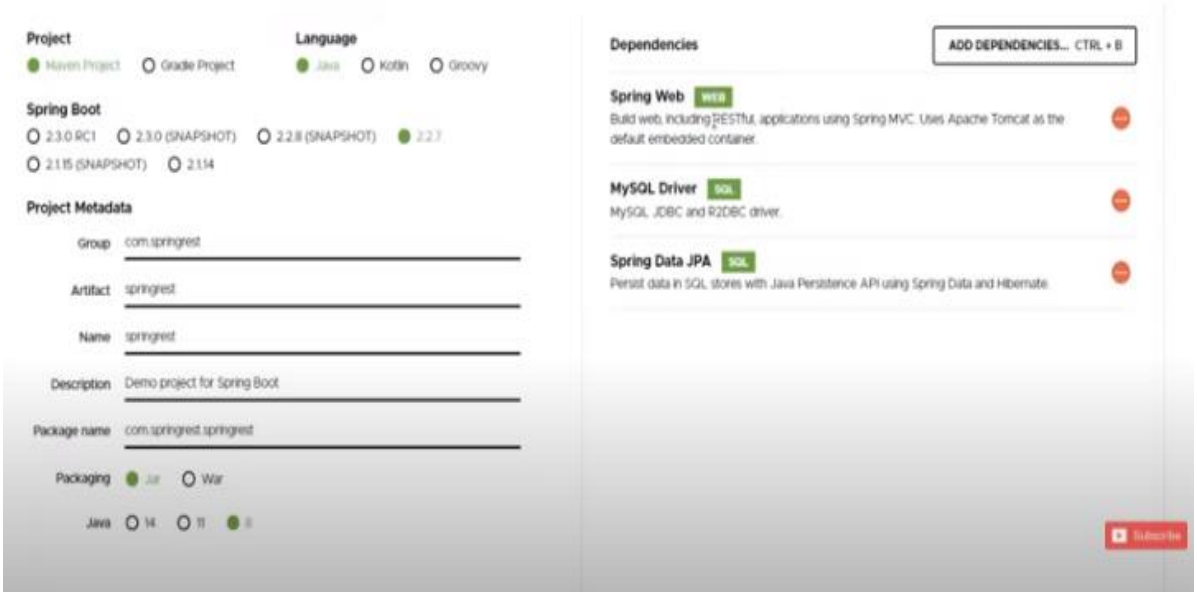
Moving towards writing API for our CV project

Request Method	API URLs	Response
GET	/biodata	Get all biodatas from database
GET	/ biodata /grno	Get only one biodata from database
POST	/ biodata	Add new biodata to database
PUT	/ biodata	Update the biodata in database
DELETE	/ biodata/grno	Delete a biodata from a database

How to start coding?
How to start spring boot project?
Type "Spring boot Initializer" in google search: You will get



Next - Click the first link : https://start.spring.io/



Select: Mavan Java and spring boot Version 2.2.7
Project Metadata: comp.springrest
Artifact: springrest ----------This is your project name
Packaging ------ jar

Add Dependencies: Spring Web.....MySQL Driver.....Spring Data JPA

Click on: Generate .....it will generate jar file in download folder

---

Extract it
Open it in ID
Import - as a maven project
Tick on ------ pom.xml
It will update

Click on your project
springrest
-- src/main/java ---- consists of all java Classess
within it
com.springrest.springrest

within it

SpringrestApplication.java ----this is our main spring boot application.

------- right click and run as app

--

src/main/resources **----** application properties --- configuration

pom.xml ---- heart file : it consists of all dependencies'- **web, jpa and my-sql**-----

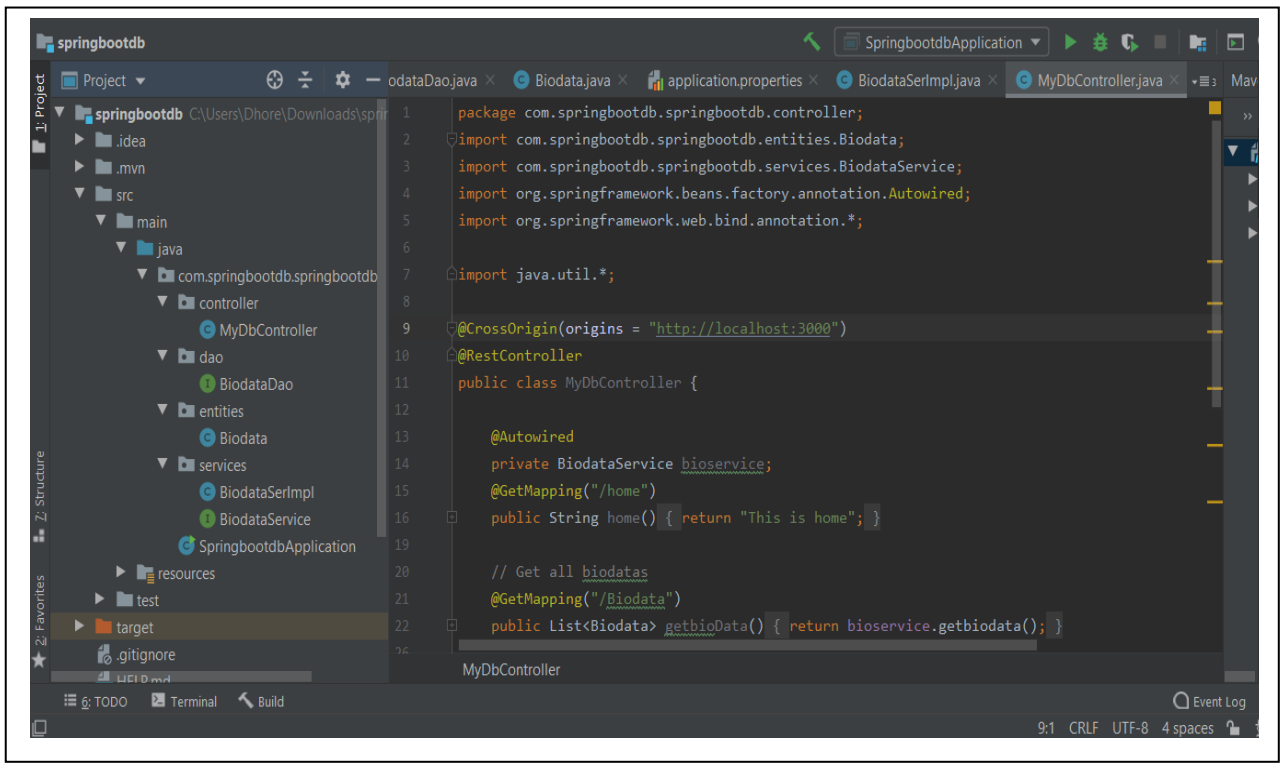
use build path -----to add and remove dependencies

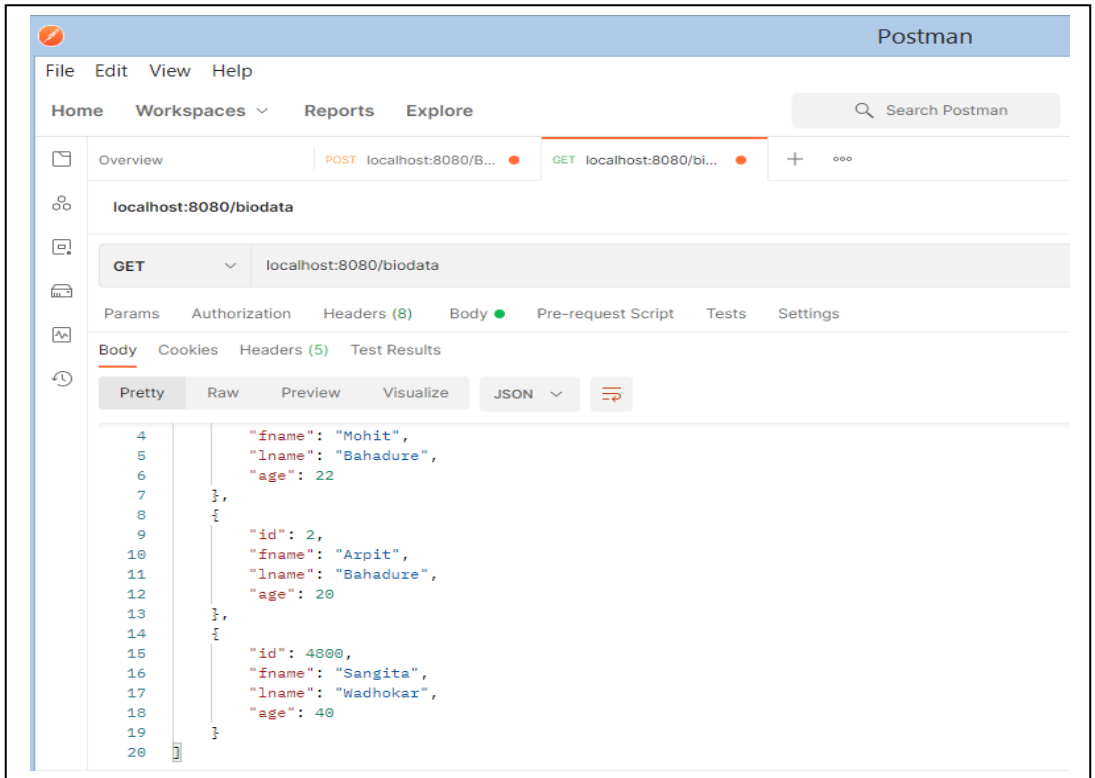
java version--updation --- project-maven---update

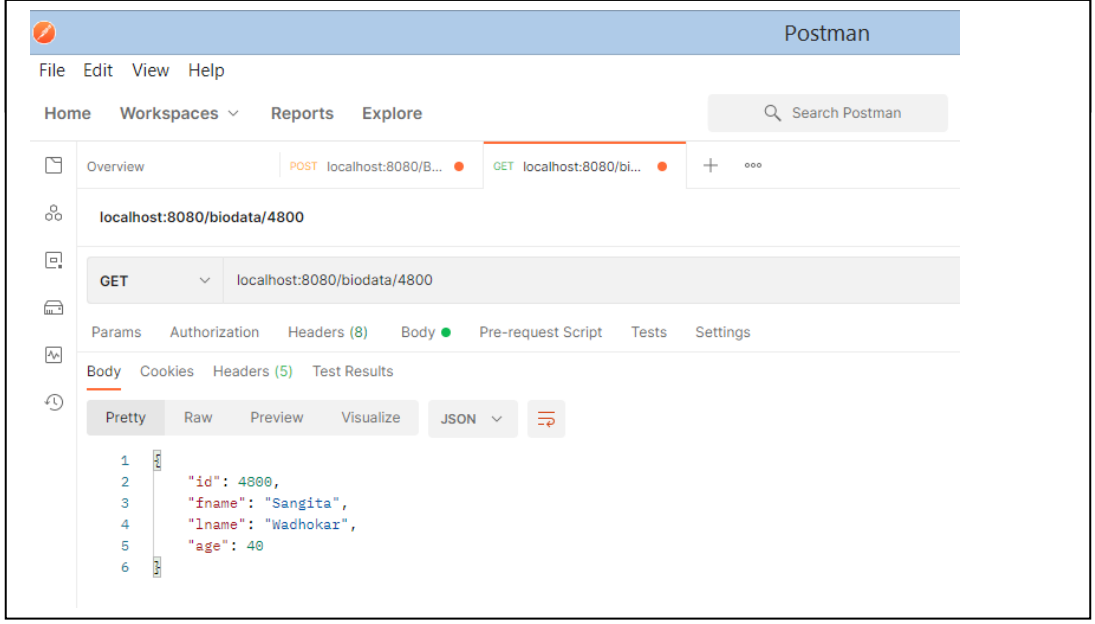
Built-in server in Maven / gradle -- Tomcat - 8080

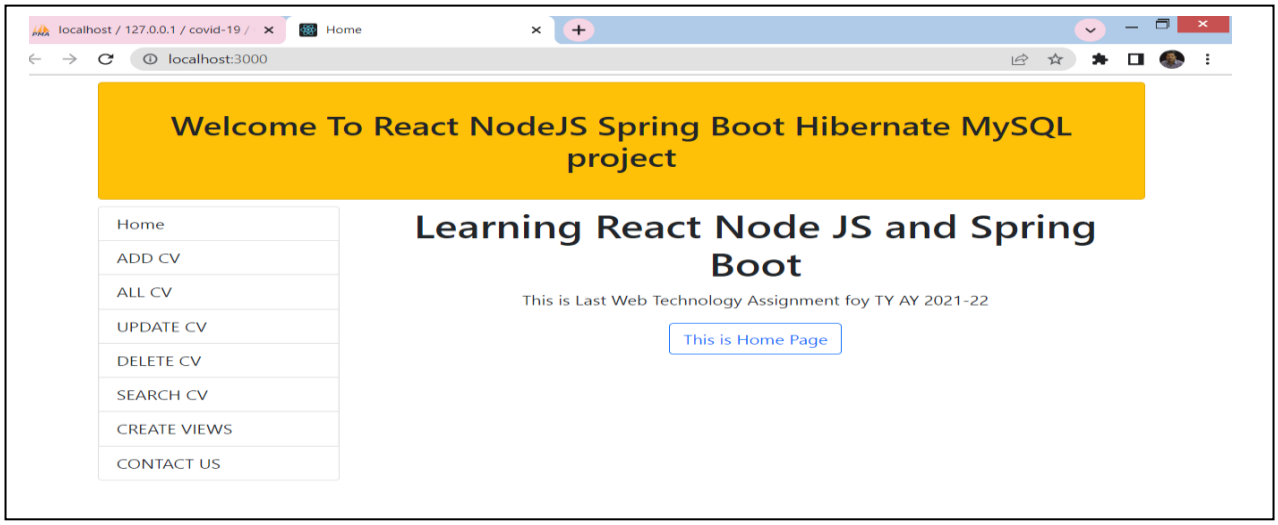
Postman: GET localhost 8080

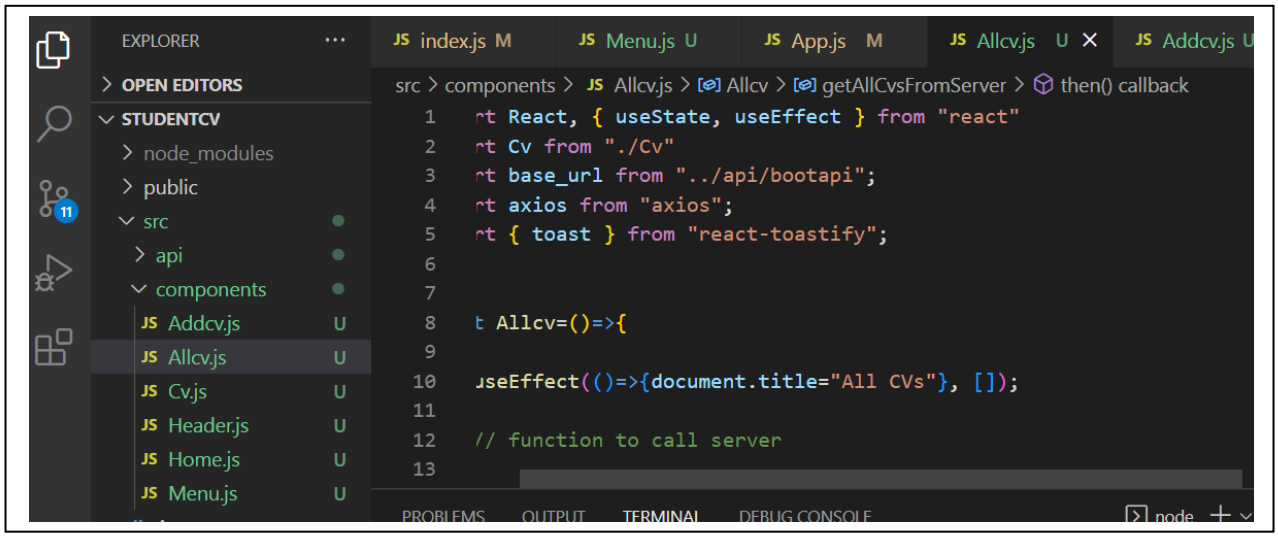
Sample Output:

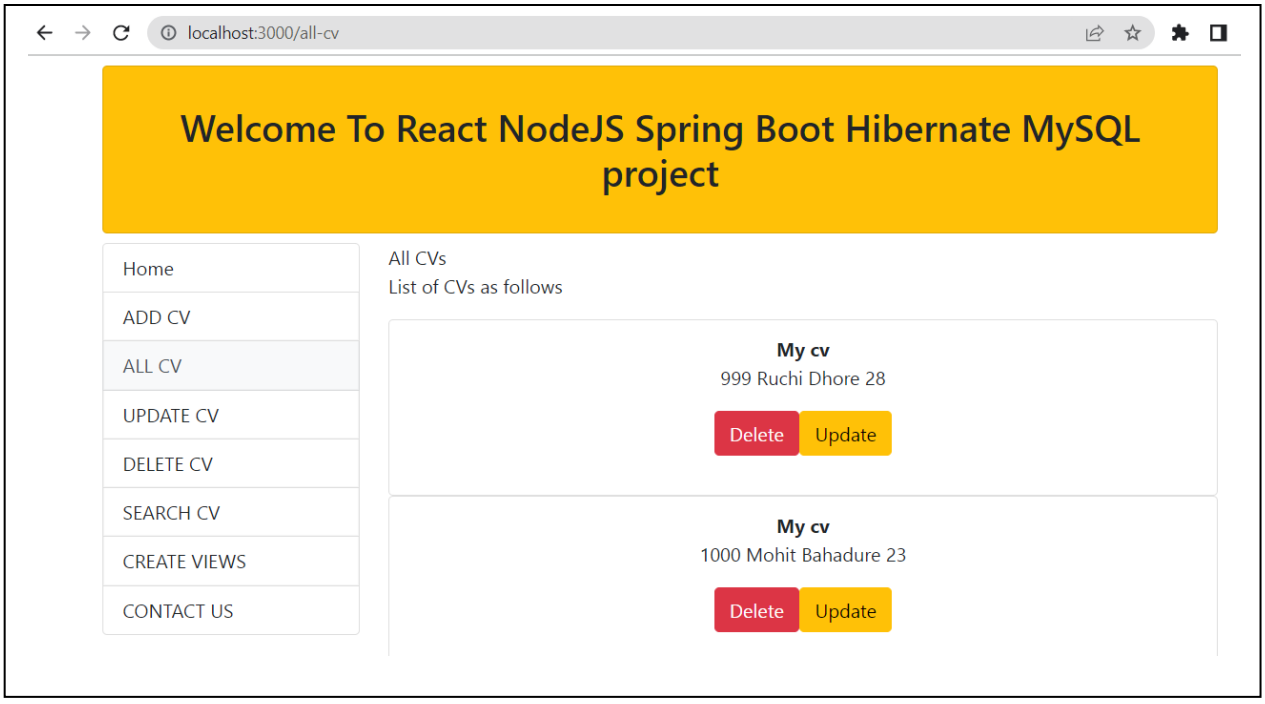



Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Overview
POST localhost:8080/B...
GET localhost:8080/bi...
localhost:8080/biodata
GET
localhost:8080/biodata
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Body Cookies Headers (5) Test Results
Pretty Raw Preview Visualize JSON
4 "fname": "Mohit",
5 "lname": "Bahadure",
6 "age": 22
7 },
8 {
9 "id": 2,
10 "fname": "Arpit",
11 "lname": "Bahadure",
12 "age": 20
13 },
14 {
15 "id": 4800,
16 "fname": "Sangita",
17 "lname": "Wadhokar",
18 "age": 40
19 }
20]


Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Overview
POST localhost:8080/B...
GET localhost:8080/bi...
localhost:8080/biodata/4800
GET
localhost:8080/biodata/4800
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Body Cookies Headers (5) Test Results
Pretty Raw Preview Visualize JSON
1 {
2 "id": 4800,
3 "fname": "Sangita",
4 "lname": "Wadhokar",
5 "age": 40
6 }


localhost / 127.0.0.1 / covid-19 /
Home
localhost:3000
Welcome To React NodeJS Spring Boot Hibernate MySQL project
Home
ADD CV
ALL CV
UPDATE CV
DELETE CV
SEARCH CV
CREATE VIEWS
CONTACT US
Learning React Node JS and Spring Boot
This is Last Web Technology Assignment foy TY AY 2021-22
This is Home Page


EXPLORER
OPEN EDITORS
STUDENTCV
node_modules
public
src
api
components
Addcv.js
Allcv.js
Cv.js
Header.js
Home.js
Menu.js
index.js M
Menu.js U
App.js M
Allcv.js U
Addcv.js U
src > components > Allcv.js > Allcv > getAllCvsFromServer > then() callback
rt React, { useState, useEffect } from "react"
rt Cv from "./Cv"
rt base_url from "../api/bootapi";
rt axios from "axios";
rt { toast } from "react-toastify";
t Allcv=()=>{
seEffect(()=>{document.title="All CVs"}, []);
// function to call server
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
node


localhost:3000/all-cv
Welcome To React NodeJS Spring Boot Hibernate MySQL project
Home
ADD CV
ALL CV
UPDATE CV
DELETE CV
SEARCH CV
CREATE VIEWS
CONTACT US
All CVs
List of CVs as follows
My cv
999 Ruchi Dhore 28
Delete
Update
My cv
1000 Mohit Bahadure 23
Delete
Update