



Mahavir Education Trust's  
**SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE**  
Chembur, Mumbai - 400 088  
**UG Program in Information Technology**

<b>Experiment No: 1</b>					
<b>Date of Performance:</b>	10/02/2022				
<b>Date of Submission:</b>	21/02/2022				
<b>Program formation/ Execution/ ethical practices (07)</b>	<b>Documentation (02)</b>	<b>Timely Submission (03)</b>	<b>Viva Answer (03)</b>	<b>Experiment Marks (15)</b>	<b>Teacher Signature with date</b>

# Experiment 1

**Aim:** Study Web Analytics using open source tool Matomo.

**Lab Outcome Number:** 3.ITL602 .1

**Lab Outcome:** Understand open source tools for web analytics and semantic web apps development and deployment.

## Theory:

### Web analytics:

Web analytics is the process of analyzing the behavior of visitors to a website. This involves tracking, reviewing and reporting data to measure web activity, including the use of a website and its components, such as webpages, images and videos.

Data collected through web analytics may include traffic sources, referring sites, page views, paths taken and conversion rates. The compiled data often forms a part of customer relationship management analytics (CRM analytics) to facilitate and streamline better business decisions.

Web analytics enables a business to retain customers, attract more visitors and increase the dollar volume each customer spends.

### Matomo:

Matomo Analytics, formerly Piwik and not to be confused with Piwik Pro (a fork of the original tool now totally independent and maintained by a New Zealand team) is a website statistics measurement tool made in France (Cocorico!). The first version of the Piwik software dates back to 2007, created by Matthieu Aubry, still the team leader today (thanks to him for this great tool). It only took the name Matomo in 2018.

The software is open source. You can contribute to its development. You can also download the full source code to install it on your own server without paying a royalty. The code of the tool is in PHP, Javascript and uses a SQL database (MySQL, PostGre is not usable for the moment). It is multi-platform. The use of PHP makes its installation easier, this language being the most used at the moment and guarantees to find a compatible server very easily.

Matomo is currently used by more than a million websites, translated into 54 languages and enjoys a large community. It is a particularly interesting solution for a professional website such as a company's showcase site. It can be limited or complex to use as soon as you want to integrate it with other tools, especially those of the Google ecosystem (Google Ads, Google DataStudio, ...). You can also use Matomo to track your application statistics (views, clicks, ...).

## Output:

- 1) Install Wamp or xampp server
- 2) In your project folder inside the Wamp or xampp directory add the file index.php



```
index.php - Notepad
File Edit Format View Help
<!DOCTYPE html>

        <html>
        <head>
            <title>My Website</title>
        </head>
        <body>

            <?php
                echo "My first Website!";
            ?>
        </body>
    </html>
```

- 3) Visit localhost and create a database

## Databases

 Create database 

- 4) From <https://matomo.org/download/> download matomo
- 5) Copy extracted **matomo** folder under project directory and proceed by entering the details in localhost/matomos
- 6) Copy the JavaScript tracking code to the **index.php** file inside the **<head>** tag as shown below:

```

index.php - Notepad
File Edit Format View Help
<!DOCTYPE html>

        <html>
            <head>
                <title>My Website</title>

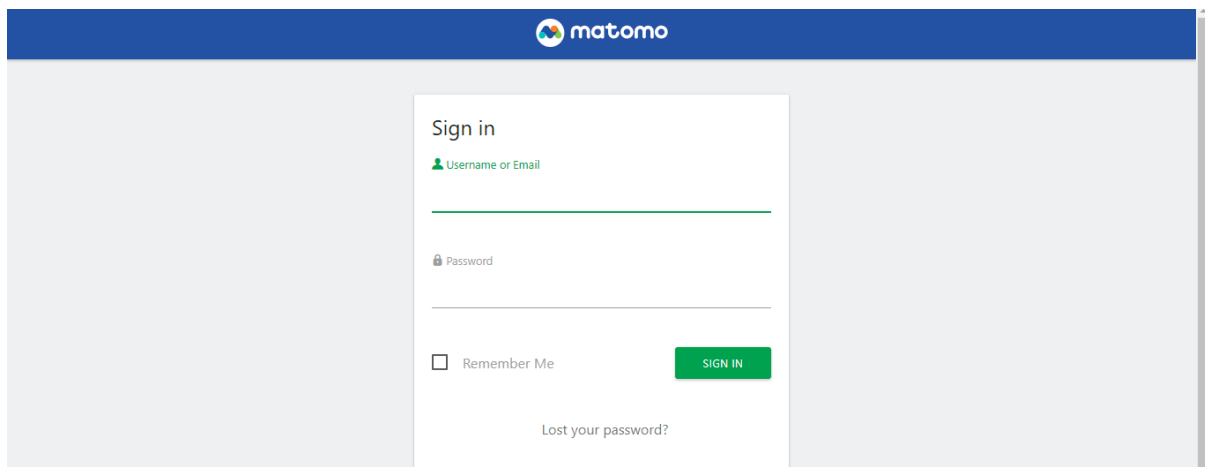
<!-- Matomo -->
<script>
    var _paq = window._paq = window._paq || [];
    /* tracker methods like "setCustomDimension" should be called before "trackPageView" */
    _paq.push(['trackPageView']);
    _paq.push(['enableLinkTracking']);
    (function() {
        var u="//localhost/matomo/";
        _paq.push(['setTrackerUrl', u+'matomo.php']);
        _paq.push(['setSiteId', '1']);
        var d=document, g=d.createElement('script'), s=d.getElementsByTagName('script')[0];
        g.async=true; g.src=u+'matomo.js'; s.parentNode.insertBefore(g,s);
    })();
</script>
<!-- End Matomo Code -->
|
        </head>
        <body>

            <?php
                echo "My first Website!";
            ?>
        </body>

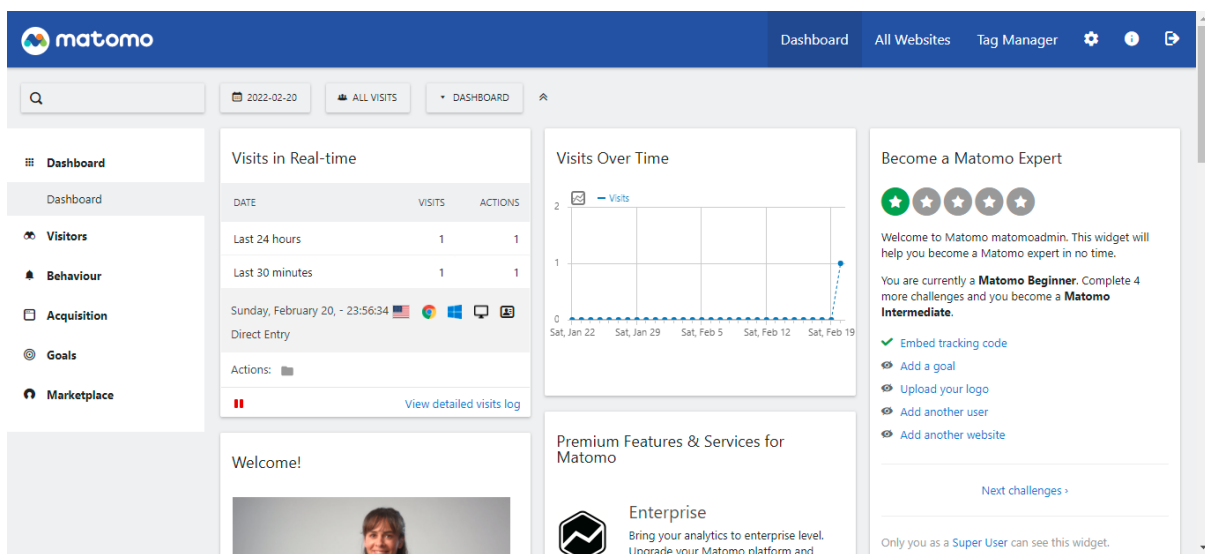
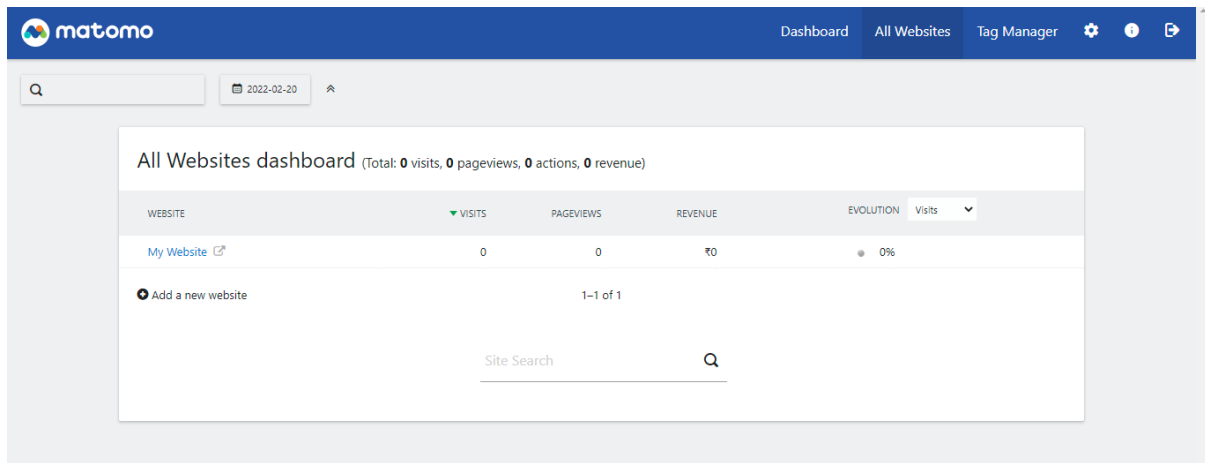
    </html>

```

- 7) Click **NEXT**, Click **CONTINUE TO MATOMO**, Use Super user login name and password for login. Check **All Website** and **Dashboard** links



The image shows the Matomo Sign in page. At the top, there is a blue header with the Matomo logo and name. Below the header, the page has a light gray background. In the center, there is a white box with the title "Sign in". Inside this box, there are two input fields: the first is labeled "Username or Email" with a green user icon, and the second is labeled "Password" with a green lock icon. Below these fields, there is a checkbox labeled "Remember Me" and a green button labeled "SIGN IN". At the bottom of the white box, there is a link that says "Lost your password?".



**Conclusion:** Installed and studied the Web Analytics tool Matomo and tried to analyze the traffic on the website.



Mahavir Education Trust's  
**SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE**  
Chembur, Mumbai - 400 088  
**UG Program in Information Technology**

Experiment No: 2					
Date of Performance:	21/02/2022				
Date of Submission:	28/02/2022				
Program formation/ Execution/ ethical practices (07)	Documentation (02)	Timely Submission (03)	Viva Answer (03)	Experiment Marks (15)	Teacher Signature with date

## Experiment 2

**Aim:** Design a web page including a Calculator using TypeScript

**Lab Outcome Number:** 3.ITL602 .2

**Lab Outcome:** Understand the basic concepts of TypeScript for designing web applications

### Theory:

JavaScript was introduced as a language for the client side. The development of Node.js has marked JavaScript as an emerging server-side technology too. However, as JavaScript code grows, it tends to get messier, making it difficult to maintain and reuse the code. Moreover, its failure to embrace the features of Object Orientation, strong type checking and compile-time error checks prevents JavaScript from succeeding at the enterprise level as a full-fledged server-side technology. **TypeScript** was presented to bridge this gap.

### TypeScript:

By definition, “TypeScript is JavaScript for application-scale development.” TypeScript is a strongly typed, object oriented, compiled language. It was designed by **Anders Hejlsberg** (designer of C#) at Microsoft. TypeScript is both a language and a set of tools. TypeScript is a typed superset of JavaScript compiled to JavaScript. In other words, TypeScript is JavaScript plus some additional features.

### Features of TypeScript:

- **TypeScript is just JavaScript.** TypeScript starts with JavaScript and ends with JavaScript. Typescript adopts the basic building blocks of your program from JavaScript. Hence, you only need to know JavaScript to use TypeScript. All TypeScript code is converted into its JavaScript equivalent for the purpose of execution.
- **TypeScript supports other JS libraries.** Compiled TypeScript can be consumed from any JavaScript code. TypeScript-generated JavaScript can reuse all of the existing JavaScript frameworks, tools, and libraries.
- **JavaScript is TypeScript.** This means that any valid **.js** file can be renamed to **.ts** and compiled with other TypeScript files.
- **TypeScript is portable.** TypeScript is portable across browsers, devices, and operating systems. It can run on any environment that JavaScript runs on. Unlike its counterparts, TypeScript doesn't need a dedicated VM or a specific runtime environment to execute.



TypeScript adopts its basic language features from the ECMAScript5 specification, i.e., the official specification for JavaScript. TypeScript language features like Modules and class-based orientation are in line with the EcmaScript 6 specification. Additionally, TypeScript also embraces features like generics and type annotations that aren't a part of the EcmaScript6 specification.

### Why Use TypeScript?

TypeScript is superior to its other counterparts like CoffeeScript and Dart programming languages in a way that TypeScript is extended JavaScript. In contrast, languages like Dart, CoffeeScript are new languages in themselves and require language-specific execution environment.

The benefits of TypeScript include –

- **Compilation** – JavaScript is an interpreted language. Hence, it needs to be run to test that it is valid. It means you write all the codes just to find no output, in case there is an error. Hence, you have to spend hours trying to find bugs in the code. The TypeScript transpiler provides the error-checking feature. TypeScript will compile the code and generate compilation errors, if it finds some sort of syntax errors. This helps to highlight errors before the script is run.
- **Strong Static Typing** – JavaScript is not strongly typed. TypeScript comes with an optional static typing and type inference system through the TLS (TypeScript Language Service). The type of a variable, declared with no type, may be inferred by the TLS based on its value.
- TypeScript **supports type definitions** for existing JavaScript libraries. TypeScript Definition file (with **.d.ts** extension) provides definition for external JavaScript libraries. Hence, TypeScript code can contain these libraries.
- TypeScript **supports Object Oriented Programming** concepts like classes, interfaces, inheritance, etc.

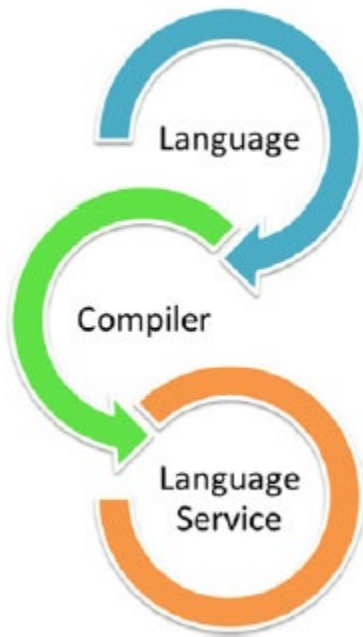
### Components of TypeScript

At its heart, TypeScript has the following three components –

- **Language** – It comprises of the syntax, keywords, and type annotations.
- **The TypeScript Compiler** – The TypeScript compiler (tsc) converts the instructions written in TypeScript to its JavaScript equivalent.



- **The TypeScript Language Service** – The "Language Service" exposes an additional layer around the core compiler pipeline that are editor-like applications. The language service supports the common set of a typical editor operations like statement completions, signature help, code formatting and outlining, colorization, etc.



## Declaration Files

When a TypeScript script gets compiled, there is an option to generate a **declaration file** (with the extension **.d.ts**) that functions as an interface to the components in the compiled JavaScript. The concept of declaration files is analogous to the concept of header files found in C/C++. The declaration files (files with **.d.ts** extension) provide intellisense for types, function calls, and variable support for JavaScript libraries like jQuery, MooTools, etc.

## Code:

### calc.html

```
<!DOCTYPE html>
<head>
<title>Simple Calculator</title>
</head>
```

```

<body>
<h1>Simple Calculator</h1>
<input placeholder="Number1" type="text" id="num1">
<select id="operator">
<option value="+">+</option>
<option value="-">-</option>
<option value="*">*</option>
<option value="/">/</option>
</select>
<input placeholder="Number2" type="text" id="num2">
<input type="button" value="Calculate" onclick="calculate()">
<p id="result"></p>
<script src="calculator.js"></script>
</body>
</html>

```

## calculator.ts

```

function calculate(){
    const num1 = document.getElementById("num1") as HTMLInputElement;
    const num2 = document.getElementById("num2") as HTMLInputElement;
    const operator = document.getElementById("operator") as HTMLSelectElement;
    const op: string = operator.options[operator.selectedIndex].text;
    let result: number;
    if(op === "+"){
        result = parseFloat(num1.value) + parseFloat(num2.value);
    } else if(op === "-"){
        result = parseFloat(num1.value) - parseFloat(num2.value);
    } else if(op === "*"){
        result = parseFloat(num1.value) * parseFloat(num2.value);
    } else if(op === "/"){

```

```
result = parseFloat(num1.value) / parseFloat(num2.value);  
}  
document.getElementById("result").innerHTML = result.toString();  
}
```

### Output:

```
C:\01-Sarim\Sem 6\Web lab\program>tsc calculator.ts  
C:\01-Sarim\Sem 6\Web lab\program>_
```

## Simple Calculator

\*

120

**Conclusion** – Designed a simple calculator using TypeScript which performs the basic arithmetic operations.



Mahavir Education Trust's  
**SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE**  
Chembur, Mumbai - 400 088  
**UG Program in Information Technology**

Experiment No: 3					
Date of Performance:	28/02/2022				
Date of Submission:	07/03/2022				
Program formation/ Execution/ ethical practices (07)	Documentation (02)	Timely Submission (03)	Viva Answer (03)	Experiment Marks (15)	Teacher Signature with date

## Experiment 3

**Aim:** Demonstrate Multilevel Inheritance example using TypeScript.

**Lab Outcome Number:** 3.ITL602 .2

**Lab Outcome:** Understand the basic concepts of TypeScript for designing web applications.

### Theory:

#### TypeScript Inheritance:

Inheritance is an aspect of OOPs languages, which provides the ability of a program to create a new class from an existing class. It is a mechanism which acquires the properties and behaviors of a class from another class. The class whose members are inherited is called the base class, and the class that inherits those members is called the derived/child/subclass. In child class, we can override or modify the behaviors of its parent class.

Before ES6, JavaScript uses functions and prototype-based inheritance, but TypeScript supports the class-based inheritance which comes from ES6 version. The TypeScript uses class inheritance through the extends keyword. TypeScript supports only single inheritance and multilevel inheritance. It doesn't support multiple and hybrid inheritance.

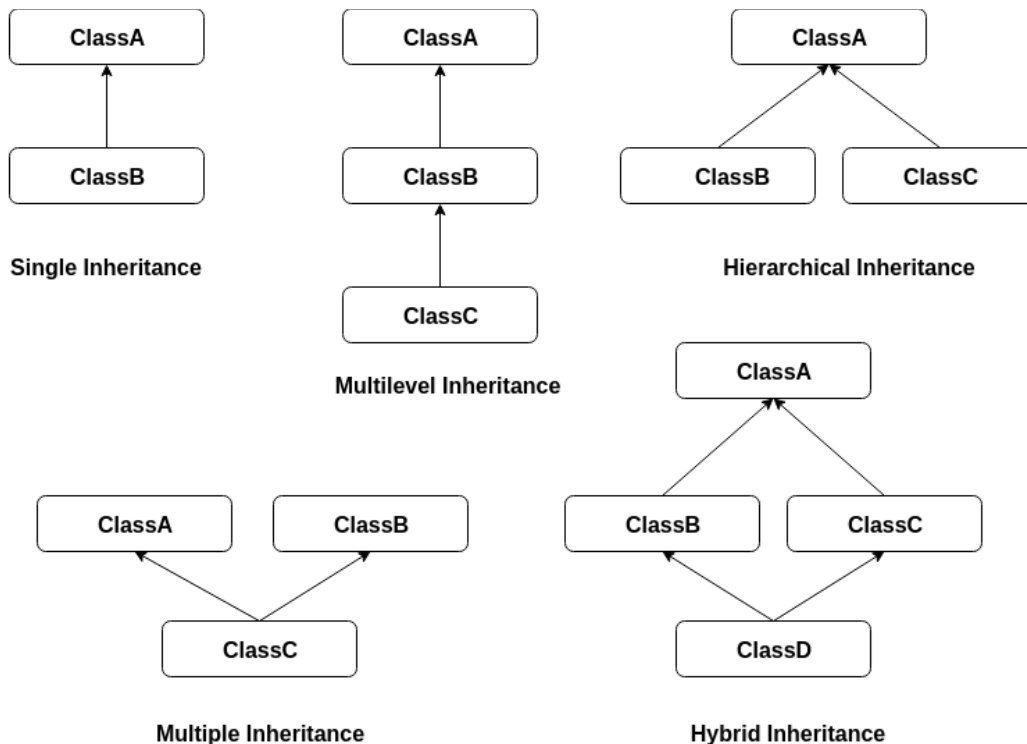
#### Why use inheritance?

- We can use it for Method Overriding (so runtime polymorphism can be achieved).
- We can use it for Code Reusability.

#### Types of Inheritance

We can classify the inheritance into the five types. These are:

- Single Inheritance
- Multilevel Inheritance
- Multiple Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance



## Single Inheritance

Single inheritance can inherit properties and behavior from at most one parent class. It allows a derived/subclass to inherit the properties and behavior of a base class that enable the code reusability as well as we can add new features to the existing code. The single inheritance makes the code less repetitive.

## Multilevel Inheritance

When a derived class is derived from another derived class, then this type of inheritance is known as multilevel inheritance. Thus, a multilevel inheritance has more than one parent class. It is similar to the relation between Grandfather, Father, and Child.

## Multiple Inheritance

When an object or class inherits the characteristics and features from more than one parent class, then this type of inheritance is known as multiple inheritance. Thus, a multiple inheritance acquires the properties from more than one parent class. TypeScript does not support multiple inheritance.

## Hierarchical Inheritance

When more than one subclass is inherited from a single base class, then this type of inheritance is known as hierarchical inheritance. Here, all features which are common in sub-classes are included in the base class. TypeScript does not support hierarchical inheritance.

## Hybrid Inheritance

When a class inherits the characteristics and features from more than one form of inheritance, then this type of inheritance is known as **Hybrid inheritance**. In other words, it is a combination of multilevel and multiple inheritance. We can implement it by combining more than one type of inheritance. TypeScript does not support hybrid inheritance.

TypeScript supports only single and multilevel inheritance. It doesn't support multiple, hierarchical, and hybrid inheritance.

### Code:

#### Inheritance.ts

```
// Parent Class
class Animal {
    // Method of Parent class
    voice1():void {
        console.log("I am the Animal(parent) class");
    }
}
// Child class extending the parent class
class Lion extends Animal {
    // Method of Child class
    voice2():void {
        console.log("I am Lion(child class of Animal)");
        console.log("I roar!!!");
    }
}
// Child class extending the parent class(Lion)
class BabyLion extends Lion {
    // Method of Child class
    voice3():void {
        console.log("I am Baby Lion(child class of Lion)");
        console.log("Hello !!!");
    }
}
```

```
}  
console.log()  
console.log("Demo to show multilevel inheritance in Typescript");  
console.log()  
// Creating objects of child class  
let myobject1 = new Lion();  
myobject1.voice1(); // Calling the methods present in this class  
myobject1.voice2();  
console.log()  
// Creating objects of child class  
let myobject2 = new BabyLion();  
// Calling the methods present in this class(all methods have been included in this class)  
myobject2.voice1();  
myobject2.voice2();  
myobject2.voice3();  
console.log()
```

### Output:

```
C:\01-Sarim\Sem 6\Web lab\program>tsc inheritance.ts  
C:\01-Sarim\Sem 6\Web lab\program>node inheritance  
Demo to show multilevel inheritance in Typescript  
  
I am the Animal(parent) class  
I am Lion(child class of Animal)  
I roar!!  
  
I am the Animal(parent) class  
I am Lion(child class of Animal)  
I roar!!  
I am Baby Lion(child class of Lion)  
Hello !!!  
  
C:\01-Sarim\Sem 6\Web lab\program>
```

**Conclusion:** Learnt about Inheritance and demonstrated multilevel inheritance example using TypeScript.





Mahavir Education Trust's  
**SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE**  
Chembur, Mumbai - 400 088  
**UG Program in Information Technology**

<b>Experiment No: 4</b>					
<b>Date of Performance:</b>	07/03/2022				
<b>Date of Submission:</b>	14/03/2022				
<b>Program formation/ Execution/ ethical practices (07)</b>	<b>Documentation (02)</b>	<b>Timely Submission (03)</b>	<b>Viva Answer (03)</b>	<b>Experiment Marks (15)</b>	<b>Teacher Signature with date</b>

## Experiment 4

**Aim:** Develop single page web application using AngularJS Framework.

**Lab Outcome No:** 3.ITL602 .3

**Lab Outcome:** Implement Single Page Applications using AngularJS Framework.

**Objective:** Develop single page web application using AngularJS Framework and apply ng-controller, ng-model, ng-bind, ng-init, expressions and filters.

**Theory:**

### What is Directive in AngularJS?

A Directive in AngularJS is a command that gives HTML new functionality. When Angular go through the HTML code, it will first find the directives in the page and then parse the HTML page accordingly. This directive is used to bind our data model to our view.

### How to Create Directives in AngularJS

As we defined in the introduction, AngularJS directives is a way to extend the functionality of HTML.

#### Different directives defined in AngularJS.

Below is the list of the AngularJS directives along with examples provided to explain each one of them.

#### 1. ng-app directive

This is used to initialize an AngularJS application. When this directive is place in an HTML page, it basically tells Angular that this HTML page is an angular.js application.

```
<div ng-app = "">  
...  
</div>
```

#### 2. ng-init directive

The ng-init directive defines **initial values** for an AngularJS application.

It is used to assign values to the variables. Normally, you will not use ng-init. You will use a controller or module instead.

```
<div ng-app = "" ng-init = "countries = [{language:'language',name:'United States'},  
  {language:'Hindi',name:'India'}, {language:'French',name:'France'}]">
```

```
...  
</div>
```

### 3. ng-model directive

The ng-model directive defines the model/variable to be used in AngularJS Application. In the following example, we define a model named *name*.

```
<div ng-app = "">  
...  
<p>Enter your Name: <input type = "text" ng-model = "name"></p>  
</div>
```

### 4. ng-repeat directive

The ng-repeat directive repeats HTML elements for each item in a collection. In the following example, we iterate over the array of countries.

```
<divng-app="">  
...  
<p>List of Countries with locale:</p>  
  
<ol>  
<li>  
  <divng-repeat="country in countries">  
    {{ 'Country: ' + country.name + ', Language: ' + country.language }}  
  </li>  
</ol>  
</div>
```

## AngularJS Filters

AngularJS provides filters to transform data:

- **currency** Format a number to a currency format.
- **date** Format a date to a specified format.
- **filter** Select a subset of items from an array.
- **json** Format an object to a JSON string.
- **limitTo** Limits an array/string, into a specified number of elements/characters.
- **lowercase** Format a string to lower case.
- **number** Format a number to a string.
- **orderBy** Orders an array by an expression.
- **uppercase** Format a string to upper case.

## Expressions

Expressions are used to bind application data to HTML. Expressions are written inside double curly braces such as in {{ expression }}. Expressions behave similar to ngbind directives. AngularJS expressions are pure JavaScript expressions and output the data where they are used.

## Using numbers

<p>Expense on Books : {{cost \* quantity}} Rs</p>

## Using Strings

<p>Hello {{student.firstname + " " + student.lastname}}!</p>

## Using Object

<p>Roll No: {{student.rollno}}</p>

## Controllers

AngularJS application mainly relies on controllers to control the flow of data in the application. A controller is defined using *ng-controller* directive. A controller is a JavaScript object that contains attributes/properties, and functions. Each controller accepts \$scope as a parameter, which refers to the application/module that the controller needs to handle.

```
<div ng-app = "" ng-controller = "studentController">
```

```
...
</div>
```

### Example:

```
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
    $scope.firstName = "John";
    $scope.lastName = "Doe";
});
```

## Setup AngularJS Development Environment

We need the following tools to setup a development environment for AngularJS:

1. AngularJS Library
2. Editor/IDE
3. Browser
4. Web server

### AngularJS Library

To download AngularJS library, go to [angularjs.org](http://angularjs.org) -> click download button, which will open the following popup.

## Download AngularJS

Branch   ?

Build    ?

CDN  ?

Bower  ?

npm

Extras [Browse additional modules](#)

[Previous Versions](#)

Select the required version from the popup and click on download button in the popup.

CDN: You can include AngularJS library from CDN url:-

<https://ajax.googleapis.com/ajax/libs/angularjs/1.3.16/angular.min.js>

Editor

AngularJS is eventually HTML and JavaScript code. So you can install any good editor/IDE as per your choice.

The following editors are recommended:

- Sublime Text
- Aptana Studio 3
- Ultra Edit
- Eclipse
- Visual Studio

**Web server**

Use any web server such as IIS, apache etc., locally for development purpose.

**Browser**

You can install any browser of your choice as AngularJS supports cross-browser compatibility. However, it is recommended to use [Google Chrome](https://www.google.com/chrome/) while developing an application.

### **Local environment setup:**

<https://angular.io/cli>

### **Installing Angular CLI**

Install the CLI using the npm package manager:

```
npm install -g @angular/cli
```

To create, build, and serve a new, basic Angular project on a development server, go to the parent directory of your new workspace use the following commands:

```
ng new my-first-project
```

```
cd my-first-project
```

```
ng serve
```

In your browser, open <http://localhost:4200/> to see the new application run. When you use the `ng serve` command to build an application and serve it locally, the server automatically rebuilds the application and reloads the page when you change any of the source files.

When you run `ng new my-first-project` a new folder, named `my-first-project`, will be created in the current working directory. Since you want to be able to create files inside that folder, make sure you have sufficient rights in the current working directory before running the command.

If the current working directory is not the right place for your project, you can change to a more appropriate directory by running `cd <path-to-other-directory>` first.

### **Example:**

```
<!DOCTYPEhtml>
<html>
<scriptsrc="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.js"></script>
<body>
<divng-app="myApp"ng-controller="myCtrl">
  <h1>{{message}}</h1>
  Country Name: <inputtype="text"ng-model="name"><br>
  <png-bind="name"></p>
</div>
```

```

<script>

constapp = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
    $scope.message= "Hello World!";

});
</script>
</body>
</html>

```

### Output:



### Example using all directives:

```

<html>
<head>
<title>Angular JS Filters</title>
<script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js">
</script>
</head>
<body>
<h2>AngularJS Single web page Application</h2>
<div ng-app = "mainApp" ng-controller = "studentController">
<table border = "0">
<tr>
<td>Enter first name:</td>
<td><input type = "text" ng-model = "student.firstName"></td>
</tr>

```

```

<tr>
<td>Enter last name: </td>
<td><input type = "text" ng-model = "student.lastName"></td>
</tr>

<tr>
<td>Enter fees: </td>
<td><input type = "text" ng-model = "student.fees"></td>
</tr>
<tr>
<td>Enter subject: </td>
<td><input type = "text" ng-model = "subjectName"></td>
</tr>
</table>
<br/>
<table border = "0">
<tr>
<td>Name in Upper Case: </td><td>{{student.fullName() | uppercase}}</td>
</tr>
<tr>
<td>Name in Lower Case: </td><td>{{student.fullName() | lowercase}}</td>
</tr>
<tr>
<td>fees: </td><td>{{student.fees | currency}}
</td>
</tr>
<tr>
<td>Subject:</td>
<td>
<ul>
<li ng-repeat="subject in student.subjects | filter:subjectName |orderBy:'marks'">
{{ subject.name + ', marks:' + subject.marks }}
</li>

```



</ul>

</td>

</tr>

</table>

</div>

<script>

```
var mainApp = angular.module("mainApp", []);
mainApp.controller('studentController', function($scope) {
  $scope.student = {
    firstName: "Rohit",
    lastName: "Jagtap",
    fees:500,
    subjects:[
      {name:'Artificial Intelligence',marks:70},
      {name:'Ethical Hacking',marks:80},
      {name:'Math',marks:65}
    ],
    fullName: function() {
      var studentObject;
      studentObject = $scope.student;
      return studentObject.firstName + " " + studentObject.lastName;
    }
  };
});
```

</script></body></html>

## Output:

**AngularJS Single web page Application**

Enter first name:

Enter last name:

Enter fees:

Enter subject:

Name in Upper Case:

ROHIT JAGTAP

Name in Lower Case:

rohit jagtap

fees:

\$500.00

Subject:

- Math, marks:65
- Artificial Intelligence, marks:70
- Ethical Hacking, marks:80

## Conclusion:

Developed single page web application using AngularJS Framework and applied ng-controller, ng-model, ng-bind, ng-init, expressions and filters.



Mahavir Education Trust's  
**SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE**  
Chembur, Mumbai - 400 088  
**UG Program in Information Technology**

<b>Experiment No: 5</b>					
<b>Date of Performance:</b>	14/03/2022				
<b>Date of Submission:</b>	21/03/2022				
<b>Program formation/ Execution/ ethical practices (07)</b>	<b>Documentation (02)</b>	<b>Timely Submission (03)</b>	<b>Viva Answer (03)</b>	<b>Experiment Marks (15)</b>	<b>Teacher Signature with date</b>

## Experiment 5

**Aim:** Implement a single page web application using AngularJS Framework including Services, Events, Validations and Built-in Helper Functions.

**Lab Outcome No:** 3.ITL602 .3

**Lab Outcome:** Implement Single Page Applications using AngularJS Framework.

**Objective:** Develop single page web application using AngularJS Framework including Services, Events, Validations and Built-in Helper Functions.

### Theory:

Built-in helper Functions

- angular.copy
- angular.equals
- angular.forEach
- angular.fromJson
- angular.identity
- angular.isArray
- angular.isDate
- angular.isDefined and angular.isUndefined
- angular.isElement
- angular.isFunction
- angular.isNumber
- angular.isObject
- angular.isString
- angular.merge
- angular.noop
- angular.toJson
- angular.copy

### Example

The angular.copy() function takes an object, array or a value and creates a deep copy of it.

angular.copy()

#### Code:

##### Objects:

```
let obj = {name: "vespa", occupation: "princess"};
let cpy = angular.copy(obj);
cpy.name = "yogurt"
// obj = {name: "vespa", occupation: "princess"}
// cpy = {name: "yogurt", occupation: "princess"}
```

### Arrays:

```
var w = [a, [b, [c, [d]]]];
var q = angular.copy(w);
// q = [a, [b, [c, [d]]]]
```

At the above example `angular.equals(w, q)` will evaluate to true because `.equals` tests equality by value. however `w === q` will evaluate to false because strict comparison between objects and arrays is done by reference.

`angular.equals`

### Example

The `angular.equals` function compares and determines if 2 objects or values are equal, `angular.equals` performs a deep comparison and returns true if and only if at least 1 of the following conditions is met.

`angular.equals(value1, value2)`

If the objects or values pass the `===` comparison

If both objects or values are of the same type, and all of their properties are also equal by using `angular.equals`

Both values are equal to NaN

Both values represent the same regular expression's result.

This function is helpful when you need to deep compare objects or arrays by their values or results rather than just references.

### Examples

```
angular.equals(1, 1) // true
angular.equals(1, 2) // false
angular.equals({}, {}) // true, note that {}==={} is false
angular.equals({a: 1}, {a: 1}) // true
angular.equals({a: 1}, {a: 2}) // false
angular.equals(NaN, NaN) // true
```

AngularJS has its own HTML events directives.

#### ● AngularJS Events

You can add AngularJS event listeners to your HTML elements by using one or more of these directives:

```
ng-blur
ng-change
ng-click
ng-copy
```

ng-cut  
ng-dblclick  
ng-focus  
ng-keydown  
ng-keypress  
ng-keyup  
ng-mousedown  
ng-mouseenter  
ng-mouseleave  
ng-mousemove  
ng-mouseover  
ng-mouseup  
ng-paste

The event directives allows us to run AngularJS functions at certain user events.

An AngularJS event will not overwrite an HTML event, both events will be executed.

---

- **Mouse Events**

Mouse events occur when the cursor moves over an element, in this order:

ng-mouseover  
ng-mouseenter  
ng-mousemove  
ng-mouseleave

Or when a mouse button is clicked on an element, in this order:

ng-mousedown  
ng-mouseup  
ng-click

You can add mouse events on any HTML element.

- **On-click:**

The ng-click Directive

The ng-click directive defines AngularJS code that will be executed when the element is being clicked.

```
<!DOCTYPE html>
```

```
<html>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
```

```

<body>

<div ng-app="myApp" ng-controller="myCtrl">

<button ng-click="count = count + 1">Click Me!</button>

<p>{{ count }}</p>

</div>

<script>

var app = angular.module('myApp', []);

app.controller('myCtrl', function($scope) {

$scope.count = 0;

});

</script>

</body>

</html>

```

## ● What is a Service?

In AngularJS, a service is a function, or object, that is available for, and limited to, your AngularJS application.

AngularJS has about 30 built-in services. One of them is the \$location service.

The \$location service has methods which return information about the location of the current web page:

## Example

Use the \$location service in a controller:

```

var app = angular.module('myApp', []);
app.controller('customersCtrl', function($scope, $location) {
    $scope.myUrl = $location.absUrl();
});

<!DOCTYPE html>

<html>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

<body>

<div ng-app="myApp" ng-controller="myCtrl">

<p>The url of this page is:</p>

<h3>{{ myUrl }}</h3>

```

```
</div>

<p>This example uses the built-in $location service to get the absolute url of the page.</p>

<script>

var app = angular.module('myApp', []);

app.controller('myCtrl', function($scope, $location) {

$scope.myUrl = $location.absUrl();

});

</script>

</body>

</html>
```

### Output:

The url of this page is:

**[https://www.w3schools.com/angular/tryit.asp?filename=try\\_ng\\_services](https://www.w3schools.com/angular/tryit.asp?filename=try_ng_services)**

This example uses the built-in \$location service to get the absolute url of the page.

## Program 3

### AngularJS Form Validation

AngularJS offers client-side form validation.

AngularJS monitors the state of the form and input fields (input, textarea, select), and lets you notify the user about the current state.

AngularJS also holds information about whether they have been touched, or modified, or not.

You can use standard HTML5 attributes to validate input, or you can make your own validation functions.

Client-side validation cannot alone secure user input. Server side validation is also necessary.

### Required

Use the HTML5 attribute required to specify that the input field must be filled out:



## E-mail

Use the HTML5 type email to specify that the value must be an e-mail:

### code:

```
<!DOCTYPE html>

<html>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

<body>

<h2>Validation Example</h2>

<form ng-app="myApp" ng-controller="validateCtrl"
name="myForm" novalidate>

<p>Username:<br>

<input type="text" name="user" ng-model="user" required>

<span style="color:red" ng-show="myForm.user.$dirty && myForm.user.$invalid">

<span ng-show="myForm.user.$error.required">Username is required.</span>

</span>

</p>


<p>Email:<br>

<input type="email" name="email" ng-model="email" required>

<span style="color:red" ng-show="myForm.email.$dirty && myForm.email.$invalid">

<span ng-show="myForm.email.$error.required">Email is required.</span>

<span ng-show="myForm.email.$error.email">Invalid email address.</span>

</span>

</p>

<p>

<input type="submit"

ng-disabled="myForm.user.$dirty && myForm.user.$invalid ||

myForm.email.$dirty && myForm.email.$invalid">

</p>

</form>

</script>
```

```
var app = angular.module('myApp', []);
app.controller('validateCtrl', function($scope) {
  $scope.user = 'John Doe';
  $scope.email = 'john.doe@gmail.com';
});
</script>
</body>
</html>
```

## Output:

### Validation Example

Username:

Email:

### Validation Example

Username:  
 Username is required.

Email:

### Validation Example

Username:

Email:  
 Invalid email address.

### Validation Example

Username:

Email:  
 Email is required.

**Conclusion:** Developed a single page web application using AngularJS Framework including Services, Events, Validations and Built-in Helper Functions.



**Mahavir Education Trust's**  
**SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE**  
Chembur, Mumbai - 400 088  
**UG Program in Information Technology**

<b>Experiment No: 6</b>					
<b>Date of Performance:</b>	21/03/2022				
<b>Date of Submission:</b>	28/03/2022				
<b>Program formation/ Execution/ ethical practices (07)</b>	<b>Documentation (02)</b>	<b>Timely Submission (03)</b>	<b>Viva Answer (03)</b>	<b>Experiment Marks (15)</b>	<b>Teacher Signature with date</b>

## Experiment 6

**Aim:** Implement a JavaScript program to perform AJAX based search.

**Lab Outcome Number:** 3.ITL602 .4

**Lab Outcome:** Develop Rich Internet Applications using AJAX.

### Theory:

#### Ajax:

AJAX is an acronym for Asynchronous JavaScript and XML. It is a group of inter-related technologies like JavaScript, DOM, XML, HTML/XHTML, CSS, XMLHttpRequest etc. AJAX allows you to send and receive data asynchronously without reloading the web page. So it is fast.

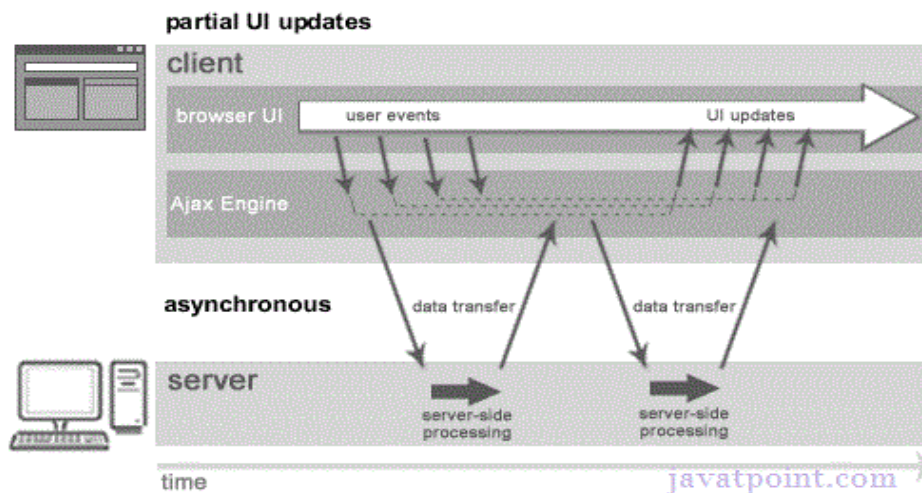
AJAX allows you to send only important information to the server not the entire page. So only valuable data from the client side is routed to the server side. It makes your application interactive and faster.

#### Where it is used?

There are too many web applications running on the web that are using ajax technology like gmail, facebook, twitter, google map, youtube etc.

#### Asynchronous (AJAX Web-Application Model)

An asynchronous request doesn't block the client i.e. browser is responsive. At that time, user can perform another operation also. In such case, Javascript engine of the browser is not blocked.



As you can see in the above image, full page is not refreshed at request time and user gets response from the ajax engine.

## AJAX Technologies

As describe earlier, ajax is not a technology but group of inter-related technologies. Ajax technologies include:

- **HTML /XHTML and CSS**
- **DOM**
- **XML or JSON**
- **XMLHttpRequest**
- **JavaScriptHTML/XHTML and CSS**

Live search has many benefits compared to traditional searching:

- Results are shown as you type
- Results narrow as you continue typing
- If results become too narrow, remove characters to see a broader result

### Code:

#### Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exp 6</title>
</script>
  function showHint(str) {
    if (str.length == 0) {
      document.getElementById("txtHint").innerHTML = "";
      return;
    } else {
      var xmlhttp = new XMLHttpRequest();
      xmlhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
          document.getElementById("txtHint").innerHTML = this.responseText;
        }
      };
      xmlhttp.open("GET", "states.php?q=" + str, true);
```

```

        xmlhttp.send();
    }
}
</script>
</head>
<body>
    <h3>Enter the names of Indian states to get suggestions:</h3>
    <form>
        <input type="text" onkeyup="showHint(this.value)">
    </form>
    <p><span id="txtHint"></span></p>
</body>
</html>

```

## states.php

```

<?php

$a[] = "Andhra Pradesh";
$a[] = "Arunachal Pradesh";
$a[] = "Assam";
$a[] = "Bihar";
$a[] = "Chhattisgarh";
$a[] = "Goa";
$a[] = "Gujarat";
$a[] = "Haryana";
$a[] = "Himachal Pradesh";
$a[] = "Jharkhand";
$a[] = "Karnataka";
$a[] = "Kerala";
$a[] = "Madhya Pradesh";
$a[] = "Maharashtra";
$a[] = "Manipur";
$a[] = "Meghalaya";
$a[] = "Mizoram";
$a[] = "Nagaland";
$a[] = "Odisha";
$a[] = "Punjab";
$a[] = "Rajasthan";
$a[] = "Sikkim";
$a[] = "Tamil Nadu";
$a[] = "Telangana";
$a[] = "Tripura";
$a[] = "Uttar Pradesh";

```

```

$a[] = "Uttarakhand";
$a[] = "West Bengal";

// get the q parameter from URL
$q = $_REQUEST["q"];

$hint = "";

// lookup all hints from array if $q is different from ""
if ($q !== "") {
    $q = strtolower($q);
    $len=strlen($q);
    foreach($a as $name) {
        if (stristr($q, substr($name, 0, $len))) {
            if ($hint === "") {
                $hint = $name;
            } else {
                $hint .= ", $name";
            }
        }
    }
}

// Output "no suggestion" if no hint was found or output correct values
echo $hint === "" ? "no suggestion" : $hint;
?>

```

## Output:

**Enter the names of Indian states to get suggestions:**

Andhra Pradesh, Arunachal Pradesh, Assam

**Enter the names of Indian states to get suggestions:**

no suggestion

## Conclusion:

Implemented a JavaScript program to perform AJAX based search.





Mahavir Education Trust's  
**SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE**  
Chembur, Mumbai - 400 088  
**UG Program in Information Technology**

<b>Experiment No: 7</b>					
<b>Date of Performance:</b>	28/03/2022				
<b>Date of Submission:</b>	28/03/2022				
<b>Program formation/ Execution/ ethical practices (07)</b>	<b>Documentation (02)</b>	<b>Timely Submission (03)</b>	<b>Viva Answer (03)</b>	<b>Experiment Marks (15)</b>	<b>Teacher Signature with date</b>

## Experiment 7

**Aim:** Perform real time form validation (before submit) using AJAX.

**Lab Outcome Number:** 3.ITL602 .4

**Lab Outcome:** Develop Rich Internet Applications using AJAX.

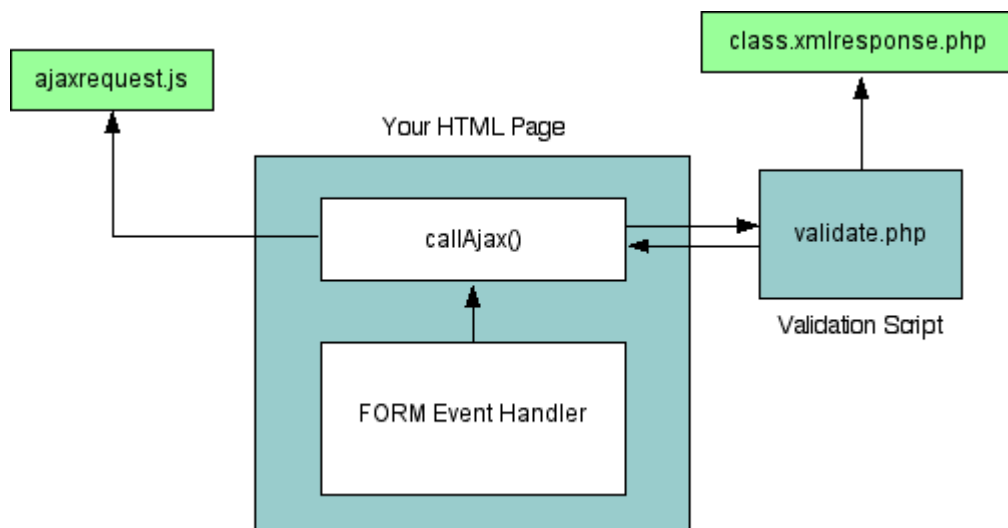
**Objective:** Perform real time form validation (before submit) using AJAX

### Theory:

Form Validation Using Ajax

AJAX (Asynchronous JavaScript and XML) is the art of exchanging data with a server, and updating parts of a web page – without reloading the whole page. Therefore, Ajax makes web page quick responsive.

A few people have commented that this is all a bit confusing when it comes to implementing it on your own website. Here is a graphic (below) showing how the different files fit together and what goes where:



You need to create two files. The first is an HTML file containing the FORM which needs to be validated, and some JavaScript which can be called from different fields in the form using event handlers (e.g. 'onclick', 'onchange', ...).

The JavaScript code then passes data to the second file which is your validation script. In this case it's a PHP file, but another server-side scripting language would work just as well. The validation script returns XML data which is then processed and applied to the form to provide feedback to the user.

The other two files (represented in the graphic in green) can be downloaded or copied from this site and included from your files as indicated.

## Realtime validation using Ajax

### Program:

#### Index.php

```
<!DOCTYPE html>
<html>
<head>
<link href="style.css" rel="stylesheet" type="text/css">
<script src="script.js"></script>
</head>
<body>
<div id="mainform">
<div class="innerdiv">
<!-- Required Div Starts Here -->
<h2>Form Validation Using AJAX</h2>
<form action="#" id="myForm" method='post' name="myForm">
<h3>Fill Your Information!</h3>
<table>
<tr>
<td>Username</td>
<td><input id='username1' name='username' onblur="validate('username', this.value)"
type='text'></td>
<td>
<div id='username'></div>
</td>
</tr>
<tr>
<td>Password</td>
<td><input id='password1' name='password' onblur="validate('password', this.value)"
type='password'></td>
<td>
<div id='password'></div>
</td>
</tr>
<tr>
<td>Email</td>
<td><input id='email1' name='email' onblur="validate('email', this.value)"
type='text'></td>
<td>
<div id='email'></div>
</td>
</tr>
```

```

<tr>
<td>website</td>
<td><input id='website1' name='website' onblur="validate('website', this.value)"
type='text'></td>
<td>
<div id='website'></div>
</td>
</tr>
</table>
<input onclick="checkForm()" type='button' value='Submit'>
</form>
</div>
</body>
</html>

```

### Script.js

```

function checkForm() {
// Fetching values from all input fields and storing them in variables.
var name = document.getElementById("username1").value;
var password = document.getElementById("password1").value;
var email = document.getElementById("email1").value;
var website = document.getElementById("website1").value;
//Check input Fields Should not be blanks.
if (name == " || password == " || email == " || website == ") {
alert("Fill All Fields");
} else {
//Notifying error fields
var username1 = document.getElementById("username");
var password1 = document.getElementById("password");
var email1 = document.getElementById("email");
var website1 = document.getElementById("website");
//Check All Values/Informations Filled by User are Valid Or Not.If All Fields Are
invalid Then Generate alert.
if (username1.innerHTML == 'Must be 3+ letters' || password1.innerHTML ==
'Password too short' || email1.innerHTML == 'Invalid email' || website1.innerHTML
== 'Invalid website') {
alert("Fill Valid Information");
} else {
//Submit Form When All values are valid.
document.getElementById("myForm").submit();
} } }
// AJAX code to check input field values when onblur event triggered.

```

```

function validate(field, query) {
var xmlhttp;
if (window.XMLHttpRequest) { // for IE7+, Firefox, Chrome, Opera, Safari
xmlhttp = new XMLHttpRequest();
} else { // for IE6, IE5
xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
}
xmlhttp.onreadystatechange = function() {
if (xmlhttp.readyState != 4 && xmlhttp.status == 200) {
document.getElementById(field).innerHTML = "Validating..";
} else if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
document.getElementById(field).innerHTML = xmlhttp.responseText;
} else {
document.getElementById(field).innerHTML = "Error Occurred. <a
href='index.php'>Reload Or Try Again</a> the page.";
}
}
xmlhttp.open("GET", "validation.php?field=" + field + "&query=" + query, false);
xmlhttp.send();
}

```

### **validation.php**

```

<?php
$value = $_GET['query'];
$formfield = $_GET['field'];
// Check Valid or Invalid user name when user enters user name in username input
field.
if ($formfield == "username") {
if (strlen($value) < 4) {
echo "Must be 3+ letters";
} else {
echo "<span>Valid</span>";
}
}
// Check Valid or Invalid password when user enters password in password input field.
if ($formfield == "password") {
if (strlen($value) < 6) {
echo "Password too short";
} else {
echo "<span>Strong</span>";
}
}
}

```

```

// Check Valid or Invalid email when user enters email in email input field.
if ($formfield == "email") {
if (!preg_match("^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.a-
z){2,3})$", $value)) {
echo "Invalid email";
} else {
echo "<span>Valid</span>";
}
}
// Check Valid or Invalid website address when user enters website address in website
input field.
if ($formfield == "website") {
if (!preg_match("/^b(?:(:https?|ftp):\\V|www\\.)[-a-z0-9+&@#\\/%?=_~_!:,.;]*[-a-z0-
9+&@#\\/%?=_~_]/i", $value)) {
echo "Invalid website";
} else {
echo "<span>Valid</span>";
} }
?>

```

## style.css

```

@import "http://fonts.googleapis.com/css?family=Fauna+One|Muli";
#mainform{
width:960px;
margin:20px auto;
padding-top:20px;
font-family:'Fauna One',serif
}
#mainform h2{
width:100%;
float:left;
text-align:center;
margin-top:35px
}
.innerdiv{
width:65%;
float:left
}
form{
background-color:#fff;
color:#123456;
box-shadow:0 1px 1px 1px gray;

```

```
width:500px;
margin:50px 250px 0 50px;
float:left;
height:400px;
padding:10px
}
h3{
margin-top:0;
color:#fff;
background-color:#0B87AA;
text-align:center;
width:100%;
height:50px;
padding-top:30px
}
input{
width:250px;
height:30px;
margin-top:10px;
border-radius:3px;
padding:2px;
box-shadow:0 1px 1px 0 #a9a9a9;
margin:10px
}
input[type=button]{
background-color:#0B87AA;
border:1px solid #fff;
font-family:'Fauna One',serif;
font-weight:700;
font-size:18px;
color:#fff;
width:50%;
margin-left:105px;
margin-top:30px
}
span{
color:green
}
#myForm div{
color:red;
font-size:14px
}
```

## Output:

### Form Validation Using AJAX

Fill Your Information!

Username

veena

Valid

Password

.....

Password too short

Email

abc@gmail

Invalid email

website

abc.com

Invalid website

Submit

## With proper validations:

### Form Validation Using AJAX

Fill Your Information!

Username

veena

Valid

Password

.....

Strong

Email

abc@gmail.com

Valid

website

www.abc.com

Valid

Submit

## Conclusion:

Used Ajax for real time form validation before submitting the form.





Mahavir Education Trust's  
**SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE**  
Chembur, Mumbai - 400 088  
**UG Program in Information Technology**

<b>Experiment No: 8</b>					
<b>Date of Performance:</b>	04/04/2022				
<b>Date of Submission:</b>	04/04/2022				
<b>Program formation/ Execution/ ethical practices (07)</b>	<b>Documentation (02)</b>	<b>Timely Submission (03)</b>	<b>Viva Answer (03)</b>	<b>Experiment Marks (15)</b>	<b>Teacher Signature with date</b>

## Experiment 8

**Aim:** Build a RESTful API using MongoDB.


**Lab Outcome Number:** 3.ITL602 .5

**Lab Outcome:** Create REST Web services using MongoDB.

**Objective:** Study and install MongoDB and Create REST Web services using MongoDB.

### Theory:

Initial Steps:

- Install MongoDB Community Server
- Open MongoDB Compass App
- In the “New Connection” screen click the **Connect** button.
- Click on the **MONGOSH** () link in the extreme left bottom corner.

Type following commands in the MONGOSH terminal

Show existing databases	Display current database	Change/create database	Show collections
show dbs show databases	db	use mydb	show collections

Prerequisite: Install extension “Thunder Client” in Visual Studio Code.

Steps to run the project:

### Code:

#### Index.js

```
const express = require('express');
const mongoose = require('mongoose');

// set up our express app
const app = express();
app.use(express.json());

// connect to mongodb
mongoose.connect('mongodb://0.0.0.0/mydb');

// initialize routes
app.use('/api', require('./routes/api'));
// home page
```

```
app.get("/", (req, res) => res.send("<h1>Build a RESTful API using MongoDB</h1>"))

// listen for requests
const port = process.env.PORT || 3000;
app.listen(port, ()=>console.log(`listening on port ${port}`));
```

## Models/book.js

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;

// create book schema & model
const BookSchema = new Schema({
  title: {
    type: String,
    required: [true, 'Name field is required']
  },
  pages: {
    type: Number
  }
});

const Book = mongoose.model('book', BookSchema);
module.exports = Book;
```

## Routs/api.js

```
const express = require('express');
const router = express.Router();
const Book = require('../models/book');

// get a list of books from the database
router.get('/books',function(req,res,next){
  Book.find({}).then(function(books){
    res.send(books);
  }).catch(next);
});

// add a new book to database
router.post('/books',function(req,res,next){
  Book.create(req.body).then(function(book){
    res.send(book);
  }).catch(next);
});

// update a book in the database
router.put('/books/:name',function(req,res,next){
  Book.findOneAndUpdate({name: req.params.name},req.body).then(function(book){
    Book.findOne({name: req.params.name}).then(function(book){
      res.send(book);
    });
  });
});
```

```
// delete a book in the database
router.delete('/books/:name',function(req,res,next){
  Book.findOneAndDelete({name: req.params.name}).then(function(book){
    res.send(book);
  });
});

module.exports = router;
```

## package.json

```
{
  "name": "rest-api",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^4.17.3",
    "mongoose": "^6.2.10"
  }
}
```

## package-lock.json

- Open the above files in Visual Studio code
- Open the terminal and type command: [npm install](#)
- To run the code type command: [node index](#)

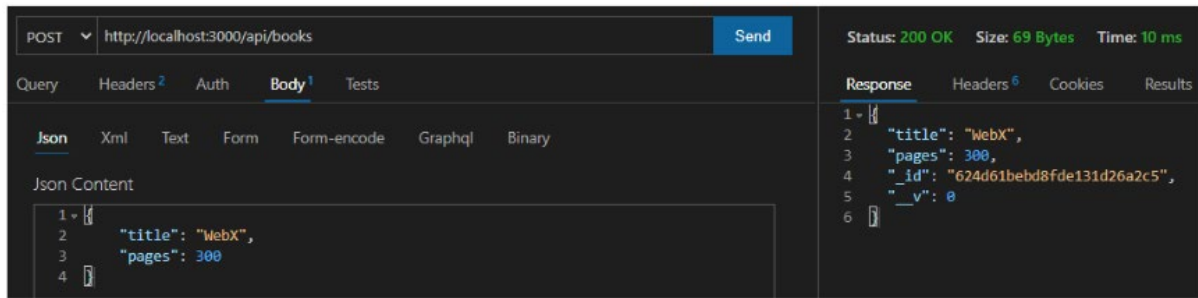
Open <http://localhost:3000/api/books> in the browser

To create a book:

- Open Thunder Client in VS Code
- Click New Request
- Select method “POST”
- Send request to “http://localhost:3000/api/books”
- In the **Body** -> **JSON** tab enter following data:

```
{
  "title": "WebX",
  "pages": 300
}
```

- Click the **Send** button and check the response sent

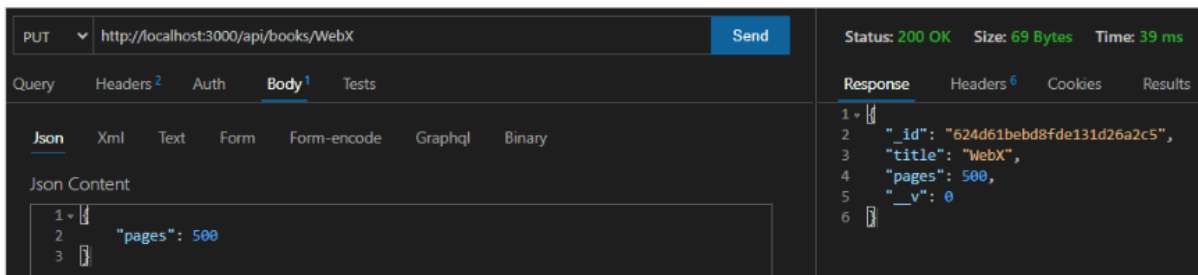


To update the book:

- Open Thunder Client in VS Code
- Click New Request
- Select method “PUT”
- Send request to “http://localhost:3000/api/books/WebX”
- In the **Body** -> **JSON** tab enter following data:

```
{
  "pages": 500
}
```

Click the **Send** button and check the response sent



To delete the book:

- Open Thunder Client in VS Code
- Click New Request
- Select method “DELETE”
- Send request to “http://localhost:3000/api/books/WebX”

Click the **Send** button

## Conclusion:

Thus, we have built a RESTful API using MongoDB



**Mahavir Education Trust's**  
**SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE**  
Chembur, Mumbai - 400 088  
**UG Program in Information Technology**

<b>Experiment No: 9</b>					
<b>Date of Performance:</b>	12/04/2022				
<b>Date of Submission:</b>	13/04/2022				
<b>Program formation/ Execution/ ethical practices (07)</b>	<b>Documentation (02)</b>	<b>Timely Submission (03)</b>	<b>Viva Answer (03)</b>	<b>Experiment Marks (15)</b>	<b>Teacher Signature with date</b>

## Experiment 9

**Aim:** Design Feedback Form using Flask.

**Lab Outcome Number:** 3.ITL602.5

**Lab Outcome:** Design web applications using Flask.

**Objective:** Study and install flask python and design web applications using Flask.

**Theory:**

### What is Flask?

Flask is an API of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web-Application. Flask is based on WSGI s(Web Server Gateway Interface) toolkit and Jinja2 template engine. Flask is called a Micro Web Framework because there is no need for third-party libraries and tools to add functionality to a website.

- Flask is a **lightweight** Web Server Gateway Interface Web Application (WSGI) framework.
- It gives the developer varieties of choice when developing web applications,
- It provides you with the necessary tools to build and deploy a web application.
- It does not enforce any dependencies or give a fixed project structure like other Python Web Frameworks like Django offers.

### Installation of flask

#### Prerequisite

Python 2.6 or higher is usually required for installation of Flask. Install virtualenv on Windows

1. Open the command line with administrator privileges.
2. Use pip to install *virtualenv* on Windows: `py -2 -m pip install virtualenv`

#### Step 2: Create an Environment

1. Make a separate directory for your project:

```
mkdir <project name>
```

2. Move into the directory:

```
cd <project name>
```

3. Within the directory, create the virtual environment for Flask. When you create the environment, a new folder appears in your project directory with the environment's name.

### Create an Environment in Windows

- **For Python 3:**

Create and name a virtual environment in Python 3 with:

```
py -3 -m venv <name of environment>
```

- **For Python 2:**

For Python 2, create the virtual environment with the *virtualenv* module:

```
py -2 -m virtualenv <name of environment>
```

List the folder structure using the **dir** command:

```
dir *<project name>*
```

### Step 3: Activate the Environment

Activate the virtual environment before installing Flask. The name of the activated environment shows up in the CLI after activation.

### Activate the Environment on Windows

For Windows, activate the virtual environment with:

```
<name of environment>\Scripts\activate
```

### Step 4: Install Flask

Install Flask within the activated environment using **pip**:

```
pip install Flask
```

Flask is installed automatically with all the dependencies.

### Step 5: Test the Development Environment

1. Create a simple Flask application to test the newly created development environment.

2. Make a file in the Flask project folder called **hello.py**.



3. Edit the file using a [text editor](#) and add the following code to make an application that prints "Hello world!":

### Code-

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello_world():
    return 'Hello world!'
```

4. Save the file and close.

5. Using the console, navigate to the project folder using the **cd** command.

6. Set the *FLASK\_APP* environment variable.

### For Windows:

```
setx FLASK_APP "hello.py"
```

7. Run the Flask application with:

```
flask run
```

The output prints out a confirmation message and the address.

8. Copy and paste the address into the browser to see the project running:

Flask web applications are easy to configure and run. It is one of the most popular web application frameworks for Python.

### Program Code:

#### main.py

```
#      Import some packages      #
from flask import Flask, render_template, request
from forms import ContactForm
import pandas as pd
#      Define flask app          #
app = Flask(__name__)
app.secret_key = 'dev fao football app'
#      Render Contact page      #
@app.route('/contactus', methods=["GET", "POST"])
```

```

def get_contact():
    form = ContactForm()
    if request.method == 'POST':
        name = request.form["name"]
        email = request.form["email"]
        subject = request.form["subject"]
        message = request.form["message"]
        res = pd.DataFrame({'name': name, 'email': email, 'subject': subject, 'message':
message}, index=[0])
        res.to_csv('./contactusMessage.csv')
        return render_template('contact.html', form=form)
    else:
        return render_template('contact.html', form=form)
#          Run app          #
if __name__ == '__main__':
    app.run(debug=True)

```

- **forms.py**

```

from flask_wtf import FlaskForm
from wtforms import TextField, BooleanField, TextAreaField, SubmitField
class ContactForm(FlaskForm):
    name = TextField("Name")
    email = TextField("Email")
    subject = TextField("Subject")
    message = TextAreaField("Message")
    submit = SubmitField("Send")

```

- **templates/contact.html**

```

<!-- here we add our contact forms -->
{% block content %}
<div class="contact">
    <h3>Contact us</h3>
    <form action="{{ url_for('get_contact') }}" method=post>
        <div>
            <!-- name forms -->
            <p>
                {{ form.name.label }} <br>
                {{ form.name(placeholder='Your name') }}
            </p>

```

```

        <!-- email forms -->
        <p>
            {{ form.email.label }} <br>
            {{ form.email(placeholder='Your e-mail ...') }}
        </p>
    <!-- Subject of your request -->
    <p>
        {{ form.subject.label }}<br>
        {{ form.subject(placeholder='Your subject ...') }}
    </p>
    <!-- Content of your request -->
    <p>
        {{ form.message.label }}<br>
        {{ form.message(placeholder='Write your request ...') }}
    </p>
    <!-- Submit button -->
    <p>
        {{ form.submit }}
    </p>
</div>
</form>
</div>
{% endblock %}
<!-- add the right css file -->
{% block styles %}
<link rel="stylesheet" href="{{ url_for('.static', filename='styles.css') }}">
{% endblock %}

```

- **static/styles.css**

```

.contact form {
    margin-top: 0px;
    padding: 0px 30px;
}
.contact h3 {
    color: #488ace;
    text-align: center;
    font-weight: 600;
    margin-top: 0px;
    margin-bottom: 0px;
}

```

```

.contact p {
    margin: 0 0px 10px;
    margin-left: 0px!important;
}
.contact label {
    display: none;
}
.contact input {
    width: 100%;
    border: none;
    border: 1px solid #dddddd78;
    padding: 10px;
    border-radius: 4px;
}
.contact textarea#message {
    width: 100%;
    height: 122px;
    border: 1px solid #dddddd78;
    border-radius: 4px;
    padding: 10px;
}
.contact input#submit {
    color: white;
    background: linear-gradient(0.25turn,#2a78ca, #71a1d4);
    float: right;
    border: none;
    border-radius: 4px;
    padding: 15px 23px;
    /* margin-right: 10px; */
    box-shadow: 0px 0px 4px 0px #00000087;
    margin-top: 15px;
}

```

- **requirements.txt**

```

flask
pandas
flask_wtf
flask==2.0.2
flask_login==0.5.0
flask_migrate==3.1.0
WTForms==3.0.1

```

```
flask_wtf==1.0.0
flask_sqlalchemy==2.5.1
sqlalchemy==1.4.29
email_validator==1.1.3
python-decouple==3.5
gunicorn==20.1.0
jinja2==3.0.3
flask-restx==0.5.1
```

## Execution of Program

### Commands

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.613]
(c) Microsoft Corporation. All rights reserved.

C:\Users\acer\OneDrive\Desktop\College\WL\Practical\Exp09>pip install flask
Requirement already satisfied: flask in c:\users\acer\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (2.1.1)
Requirement already satisfied: importlib-metadata>=3.6.0 in c:\users\acer\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from flask) (4.11.3)
Requirement already satisfied: Werkzeug>=2.0 in c:\users\acer\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from flask) (2.1.1)
Requirement already satisfied: itsdangerous>=2.0 in c:\users\acer\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from flask) (2.1.2)
Requirement already satisfied: click>=8.0 in c:\users\acer\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from flask) (8.1.2)
Requirement already satisfied: Jinja2>=3.0 in c:\users\acer\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from flask) (3.1.1)
Requirement already satisfied: colorama in c:\users\acer\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from click>=8.0->flask) (0.4.4)
Requirement already satisfied: zipp>=0.5 in c:\users\acer\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from importlib-metadata>=3.6.0->flask) (3.8.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\acer\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from Jinja2>=3.0->flask) (2.1.1)

C:\Users\acer\OneDrive\Desktop\College\WL\Practical\Exp09>
```

```
C:\Windows\System32\cmd.exe
C:\Users\acer\OneDrive\Desktop\College\WL\Practical\Exp09>pip install -r requirement.txt
Requirement already satisfied: flask in c:\users\acer\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from -r requirement.txt (line 1)) (2.1.1)
Requirement already satisfied: pandas in c:\users\acer\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from -r requirement.txt (line 2)) (1.4.2)
Collecting flask_wtf
  Downloading Flask_WTF-1.0.1-py3-none-any.whl (12 kB)
Collecting flask
  Downloading Flask-2.0.2-py3-none-any.whl (95 kB)
----- 95.2/95.2 KB 494.2 kB/s eta 0:00:00
Collecting flask_login==0.5.0
  Downloading Flask_Login-0.5.0-py2.py3-none-any.whl (16 kB)
Collecting flask_migrate==3.1.0
  Downloading Flask_Migrate-3.1.0-py3-none-any.whl (20 kB)
Collecting WTForms==3.0.1
  Downloading WTForms-3.0.1-py3-none-any.whl (136 kB)
----- 136.5/136.5 KB 384.8 kB/s eta 0:00:00
Collecting flask_wtf
  Downloading Flask_WTF-1.0.0-py3-none-any.whl (12 kB)
Collecting flask_sqlalchemy==2.5.1
  Downloading Flask_SQLAlchemy-2.5.1-py2.py3-none-any.whl (17 kB)
Collecting sqlalchemy==1.4.29
  Downloading SQLAlchemy-1.4.29-cp39-cp39-win_amd64.whl (1.5 MB)
----- 1.5/1.5 MB 664.7 kB/s eta 0:00:00
Collecting email_validator==1.1.3
  Downloading email_validator-1.1.3-py2.py3-none-any.whl (18 kB)
Collecting python-decouple==3.5
  Downloading python_decouple-3.5-py3-none-any.whl (9.6 kB)
Collecting gunicorn==20.1.0
  Downloading gunicorn-20.1.0-py3-none-any.whl (79 kB)
```

```
C:\Users\acer\OneDrive\Desktop\College\WL\Practical\Exp09>pip install pandas
Requirement already satisfied: pandas in c:\users\acer\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (1.4.2)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\acer\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from pandas) (2.8.2)
Requirement already satisfied: numpy>=1.18.5 in c:\users\acer\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from pandas) (1.22.3)
Requirement already satisfied: pytz>=2020.1 in c:\users\acer\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from pandas) (2022.1)
Requirement already satisfied: six>=1.5 in c:\users\acer\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\python39\site-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
```

```
cmd Command Prompt - python main.py

(sakec) C:\Users\Vishwajit\sakec>python forms.py

(sakec) C:\Users\Vishwajit\sakec>python main.py
* Serving Flask app 'main' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 112-707-585
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [20/Mar/2022 13:45:38] "GET / HTTP/1.1" 404 -
127.0.0.1 - - [20/Mar/2022 13:45:57] "GET /contact.html HTTP/1.1" 404 -
127.0.0.1 - - [20/Mar/2022 13:48:18] "GET /contactus HTTP/1.1" 200 -
127.0.0.1 - - [20/Mar/2022 13:48:18] "GET /static/styles.css HTTP/1.1" 200 -
127.0.0.1 - - [20/Mar/2022 13:48:18] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [20/Mar/2022 13:51:15] "POST /contactus HTTP/1.1" 200 -
127.0.0.1 - - [20/Mar/2022 13:51:15] "GET /static/styles.css HTTP/1.1" 304 -
127.0.0.1 - - [20/Mar/2022 13:51:17] "POST /contactus HTTP/1.1" 200 -
127.0.0.1 - - [20/Mar/2022 13:51:17] "GET /static/styles.css HTTP/1.1" 304 -
```

## Output

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000/contactus'. The page title is 'Contact us'. The form contains the following elements:

- A text input field labeled 'Your name'.
- A text input field labeled 'Your e-mail ...'.
- A text input field labeled 'Your subject ...'.
- A larger text area labeled 'Write your request ...'.
- A blue button labeled 'Send' at the bottom of the form.

**Conclusion:** Hence we have designed a Feedback Form using Flask. Flask web applications are easy to configure and run.



**Mahavir Education Trust's**  
**SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE**  
Chembur, Mumbai - 400 088  
**UG Program in Information Technology**

<b>Experiment No: 10</b>					
<b>Date of Performance:</b>	13/04/2022				
<b>Date of Submission:</b>	13/04/2022				
<b>Program formation/ Execution/ ethical practices (07)</b>	<b>Documentation (02)</b>	<b>Timely Submission (03)</b>	<b>Viva Answer (03)</b>	<b>Experiment Marks (15)</b>	<b>Teacher Signature with date</b>

## Experiment 10

**Aim:** Design Portfolio Website using Flask.

**Lab Outcome Number:** 3.ITL602 .6

**Lab Outcome:** Design web applications using Flask.

**Objective:** To design Portfolio Website using flask framework.

**Theory:**

### What is Flask?

Flask is an API of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web-Application. Flask is based on WSGI (Web Server Gateway Interface) toolkit and Jinja2 template engine. Flask is called a Micro Web Framework because there is no need for third-party libraries and tools to add functionality to a website.

- Flask is a **lightweight** Web Server Gateway Interface Web Application (WSGI) framework.
- It gives the developer varieties of choice when developing web applications,
- It provides you with the necessary tools to build and deploy a web application.
- It does not enforce any dependencies or give a fixed project structure like other Python Web Frameworks like Django offers.

### Step 1: Create an Environment in Windows

- For Python 3:

```
py -3 -m venv <name of environment>
```

- For Python 2:

```
py -2 -m virtualenv <name of environment>
```

### Step 2: Activate the Environment

Activate the Environment on Windows

```
<name of environment>\Scripts\activate
```

### Step 5: Make a file in the Flask project folder main.py



## Code:

- **main.py**

```
from flask import Flask, render_template, request, url_for, redirect
from email.mime.text import MIMEText
import smtplib
from email.message import EmailMessage
app = Flask(__name__)
@app.route("/")
def index():
    return render_template("index.html")
@app.route("/sendemail/", methods=['POST'])
def sendemail():
    if request.method == "POST":
        name = request.form['name']
        subject = request.form['Subject']
        email = request.form['_replyto']
        message = request.form['message']
        your_name = "John Doe"
        your_email = "johndoe@gmail.com"
        your_password = "pASSWORD@25243912524391"
        # Logging in to our email account
        server = smtplib.SMTP('smtp.gmail.com', 587)
        server.ehlo()
        server.starttls()
        server.login(your_email, your_password)
        # Sender's and Receiver's email address
        sender_email = "ayushkalla2050@gmail.com"
        receiver_email = "akalla@stevens.edu"
        msg = EmailMessage()
        msg.set_content("First Name : "+str(name)+"\nEmail : "+str(email)+"\nSubject : "+str(subject)+"\nMessage : "+str(message))
        msg['Subject'] = 'New Response on Personal Website'
        msg['From'] = sender_email
        msg['To'] = receiver_email
        # Send the message via our own SMTP server.
        try:
            # sending an email
            server.send_message(msg)
        except:
            pass
    return redirect('/');
if __name__ == "__main__":
    app.run(debug=True)
```

- **index.html**

```
<!DOCTYPE html>
<html lang="en-US">
<head>
<meta charset="UTF-8">
```

```

<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>John Doe</title>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.7.0/dist/css/bootstrap.min.css">
<link href="https://fonts.googleapis.com/css?family=Montserrat:400,700,200" rel="stylesheet">
<link href="https://maxcdn.bootstrapcdn.com/font-awesome/latest/static/css/font-awesome.min.css"
rel="stylesheet">
<link rel="stylesheet" type="text/css" href= "{{ url_for('static',filename='css/aos.css') }}">
<link rel="stylesheet" type="text/css" href= "{{ url_for('static',filename='css/bootstrap.min.css') }}">
<link rel="stylesheet" type="text/css" href= "{{ url_for('static',filename='css/main.css') }}">

<noscript>
<style type="text/css">
    [data-aos] {
        opacity: 1 !important;
        transform: translate(0) scale(1) !important;
    }
</style>
</noscript>
</head>
<body id="top">
<header>
<div class="profile-page sidebar-collapse">
<nav class="navbar navbar-expand-lg fixed-top navbar-transparent bg-primary" color-on-scroll="400">
<div class="container">

    <button class="navbar-toggler navbar-toggler" type="button" data-toggle="collapse" data-
target="#navigation" aria-controls="navigation" aria-expanded="false" aria-label="Toggle navigation"><span
class="navbar-toggler-bar bar1"></span><span class="navbar-toggler-bar bar2"></span><span class="navbar-
toggler-bar bar3"></span></button>
</div>
<div class="collapse navbar-collapse justify-content-end" id="navigation">
<ul class="navbar-nav">
<li class="nav-item"><a class="nav-link smooth-scroll" href="#about">About</a></li>
<li class="nav-item"><a class="nav-link smooth-scroll" href="#skill">Skills</a></li>
<li class="nav-item"><a class="nav-link smooth-scroll" href="#experience">Experience</a></li>
<li class="nav-item"><a class="nav-link smooth-scroll" href="#contact">Contact</a></li>
</ul>
</div>
</div>
</nav>
</div>
</header>
<div class="page-content">
<div>
<div class="profile-page">
<div class="wrapper">
<div class="page-header page-header-small filter-color="green">
<div class="page-header-image" data-parallax="true" style="background-image: url('static/images/cc-bg-
1.jpg')"></div>
<div class="container">
<div class="content-center">

    <div class="h2 title">John Doe</div>
    <p class="category text-white">Full Stack Developer</p><a class="btn btn-primary smooth-scroll mr-2"
href="#contact" data-aos="zoom-in" data-aos-anchor="data-aos-anchor">Hire Me</a><a class="btn btn-
primary" target="_blank" href="{{ url_for('static',filename='Resume_Ayush_Kalla.pdf') }}" data-aos="zoom-
in" data-aos-anchor="data-aos-anchor">Download CV</a>
</div>

```

```

</div>
<div class="section">
  <div class="container">
    <div class="button-container"><a class="btn btn-default btn-round btn-lg btn-icon" target = "_blank"
href="https://www.facebook.com/ayush.kalla/" rel="tooltip" title="Follow me on Facebook"><i class="fa fa-
facebook"></i></a><a class="btn btn-default btn-round btn-lg btn-icon" target="_blank"
href="https://twitter.com/KallaAyush" rel="tooltip" title="Follow me on Twitter"><i class="fa fa-
twitter"></i></a><a class="btn btn-default btn-round btn-lg btn-icon" target="_blank"
href="https://github.com/akalla123" rel="tooltip" title="Follow me on Github"><i class="fa fa-
github"></i></a><a class="btn btn-default btn-round btn-lg btn-icon" target="_blank"
href="https://www.linkedin.com/in/ayushkalla/" rel="tooltip" title="Follow me on LinkedIn"><i class="fa fa-
linkedin"></i></a></div>
    </div>
  </div>
</div>
</div>
</div>
<div class="section" id="about">
  <div class="container">
    <div class="card" data-aos="fade-up" data-aos-offset="10">
      <div class="row">
        <div class="col-lg-6 col-md-12">
          <div class="card-body">
            <div class="h4 mt-0 title">About</div>
            <p>A Full Stack Developer is someone who works with the Back End — or server side — of the
application as well as the Front End, or client side. Full Stack Developers have to have some skills in a wide
variety of coding niches, from databases to graphic design and UI/UX management in order to do their job
well.</p>
            </div>
          </div>
        <div class="col-lg-6 col-md-12">
          <div class="card-body">
            <div class="h4 mt-0 title">Basic Information</div>
            <div class="row">
              <div class="col-sm-4"><strong class="text-uppercase">Age:</strong></div>
              <div class="col-sm-8">20</div>
            </div>
            <div class="row mt-3">
              <div class="col-sm-4"><strong class="text-uppercase">Email:</strong></div>
              <div class="col-sm-8">johndoe@gmail.com</div>
            </div>
            <div class="row mt-3">
              <div class="col-sm-4"><strong class="text-uppercase">Phone:</strong></div>
              <div class="col-sm-8">9998887776</div>
            </div>
            <div class="row mt-3">
              <div class="col-sm-4"><strong class="text-uppercase">Address:</strong></div>
              <div class="col-sm-8">Hoboken, New Jersey, U.S.A</div>
            </div>
            <div class="row mt-3">
              <div class="col-sm-4"><strong class="text-uppercase">Language:</strong></div>
              <div class="col-sm-8">English</div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
<div class="section" id="skill">

```

```

<div class="container">
  <div class="h4 text-center mb-4 title">Professional Skills</div>
  <div class="card" data-aos="fade-up" data-aos-anchor-placement="top-bottom">
    <div class="card-body">
      <b>Programming Languages:</b> Python, R, SQL, JavaScript<br><br>
      <b>Tools:</b> Scikit-Learn, NumPy, Pandas, Matplotlib, Tableau, MS Excel, OpenCV, Keras,
Tensorflow<br><br>
      <b>Database Management Systems:</b> MySQL, SQLite3<br><br>
      <b>Internet Technologies:</b> Flask, HTML5, CSS, XML, jQuery
    </div>
  </div>
</div>
</div>

<div class="section" id="experience">
  <div class="container cc-experience">
    <div class="h4 text-center mb-4 title">Work Experience</div>
    <div class="card">
      <div class="row">
        <div class="col-md-3 bg-primary" data-aos="fade-right" data-aos-offset="50" data-aos-duration="500">
          <div class="card-body cc-experience-header">
            <p>June 2020 - Present</p>
            <div class="h5">MATRiX ANALYTICS CORPORATION</div>
          </div>
        </div>
        <div class="col-md-9" data-aos="fade-left" data-aos-offset="50" data-aos-duration="500">
          <div class="card-body">
            <div class="h5">Full Stack Developer</div>
            <p>
              <ul>
                <li>Delivered Model Risk Management Dashboards to track model performance post deployment for
a lending client.</li>
                <li>Working on building code, enhanced templates and defining a best practice for one of the key
milestones of XAI model
development process of Credit risk management models of MATRiX.</li>
                <li>Built AI based time-series forecasting models to perform demand forecasting of inventory. Used
various approaches of AI
Long short-term memory (LSTM), Spline regression, ARIMA, Holt-Winter and multi dimensional
macroeconomic model
adjusted predictions for a retail company client.</li>
                <li>Working on developing a deep learning solution for a retail client to determine the composition of
products using images.
Used algorithms such as Convolutional Neural Network and Masked RCNN</li>
                <li>Built budget, cashflow and monthly sales and purchase monitoring Tableau dashboards for retail
clients</li>
              </ul>
            </p>
          </div>
        </div>
      </div>
    </div>
    <div class="card">
      <div class="row">
        <div class="col-md-3 bg-primary" data-aos="fade-right" data-aos-offset="50" data-aos-duration="500">
          <div class="card-body cc-experience-header">
            <p>June 2019 - April 2020</p>
            <div class="h5">MATRiX ANALYTICS CORPORATION</div>
          </div>
        </div>
        <div class="col-md-9" data-aos="fade-left" data-aos-offset="50" data-aos-duration="500">

```

```

<div class="card-body">
  <div class="h5">Data Scientist Intern</div>
  <p>
    <ul>
      <li>Built suite of AI based revenue and balance models for one of the major banking client.</li>
      <li>Automated the task of Exploratory Data Analysis by writing a Python Script to perform analysis
on any given dataset.</li>
      <li>Developed front-end forms for clients to get user-input and run Machine Learning models for the
client using JavaScript.</li>
      <li>Built Dashboards using IoT data from sensors for a retail client.</li>
      <li>Built budget, cashflow and monthly sales and purchase monitoring Tableau dashboards for retail
clients</li>
      <li>Automated the task of data processing by performing Exclusion Analysis by preparing a general
template to work for any type
of incoming data.</li>
    </ul>
  </p>
</div>
</div>
</div>
</div>
<div class="card">
  <div class="row">
    <div class="col-md-3 bg-primary" data-aos="fade-right" data-aos-offset="50" data-aos-duration="500">
      <div class="card-body cc-experience-header">
        <p>September 2017 - December 2017</p>
        <div class="h5">Central Drug Research Institute</div>
      </div>
    </div>
    <div class="col-md-9" data-aos="fade-left" data-aos-offset="50" data-aos-duration="500">
      <div class="card-body">
        <div class="h5">Research Intern</div>
        <p>
          <ul>
            <li>Analyzed over 30 research papers and data sources to gather data on the spread of Parkinson's
Disease</li>
            <li>Visualized various trends in the spread of Parkinson's Disease across the Globe using
Tableau.</li>
            <li>Determined various factors directly or indirectly affecting the disease.</li>
          </ul>
        </p>
      </div>
    </div>
  </div>
</div>
</div>
</div>
<div class="section">
  <div class="container cc-education">
    <div class="h4 text-center mb-4 title">Education</div>
    <div class="card">
      <div class="row">
        <div class="col-md-3 bg-primary" data-aos="fade-right" data-aos-offset="50" data-aos-duration="500">
          <div class="card-body cc-education-header">
            <p>2018 - 2020</p>
            <div class="h5">Master's Degree</div>
          </div>
        </div>
        <div class="col-md-9" data-aos="fade-left" data-aos-offset="50" data-aos-duration="500">
          <div class="card-body">

```

```

<div class="h5">M.S. in Information Systems</div>
<p class="category">STEVENS INSTITUTE OF TECHNOLOGY</p>
<p>Relevant Coursework: Web Mining, Knowledge Discovery in Databases, Marketing Analytics</p>
<p>
  <b>Projects:</b><br>
  <ul>
    <li>Human Trafficking Knowledge Graph - A knowledge graph generating project that helped identify
over 100 cases of
    Human Trafficking around NYC. Judged as the best solution for the competition.</li>
    <li>Hate Comment Detector – A flask powered website to automatically filter hate comments from
Twitter and Live News using
    Logistic Regression algorithm.</li>
  </ul>
</p>
</div>
</div>
</div>
<div class="card">
  <div class="row">
    <div class="col-md-3 bg-primary" data-aos="fade-right" data-aos-offset="50" data-aos-duration="500">
      <div class="card-body cc-education-header">
        <p>2014 - 2018</p>
        <div class="h5">Bachelor's Degree</div>
      </div>
    </div>
    <div class="col-md-9" data-aos="fade-left" data-aos-offset="50" data-aos-duration="500">
      <div class="card-body">
        <div class="h5">B.Tech in Information Technology</div>
        <p class="category">UNIVERSITY OF MUMBAI</p>
        <p>Relevant Coursework: Web Development, Advanced Database Management Systems, Object
Oriented Programming, Data Mining
        and Business Intelligence</p>
        <p>
          <b>Projects:</b><br>
          <ul>
            <li>AutoPlay - A Machine Learning model to predict the outcome of a cricket game beforehand, based
on the past records
            and performances.</li>
          </ul>
        </p>
      </div>
    </div>
  </div>
</div>

```

```

<div class="section" id="contact">
  <div class="cc-contact-information" style="background-image: url('static/images/map.JPG')">
    <div class="container">
      <div class="cc-contact">
        <div class="row">
          <div class="col-md-9">
            <div class="card mb-0" data-aos="zoom-in">
              <div class="h4 text-center title">Contact Me</div>
              <div class="row">
                <div class="col-md-6">
                  <div class="card-body">
                    <form action="/sendemail/" method="POST">
                      <div class="p pb-3"><strong>Feel free to contact me </strong></div>

```

```

        <div class="row mb-3">
            <div class="col">
                <div class="input-group"><span class="input-group-addon"><i class="fa fa-user-
circle"></i></span>
                    <input class="form-control" type="text" name="name" placeholder="Name"
required="required"/>
                </div>
            </div>
        </div>
        <div class="row mb-3">
            <div class="col">
                <div class="input-group"><span class="input-group-addon"><i class="fa fa-file-
text"></i></span>
                    <input class="form-control" type="text" name="Subject" placeholder="Subject"
required="required"/>
                </div>
            </div>
        </div>
        <div class="row mb-3">
            <div class="col">
                <div class="input-group"><span class="input-group-addon"><i class="fa fa-
envelope"></i></span>
                    <input class="form-control" type="email" name="_replyto" placeholder="E-mail"
required="required"/>
                </div>
            </div>
        </div>
        <div class="row mb-3">
            <div class="col">
                <div class="form-group">
                    <textarea class="form-control" name="message" placeholder="Your Message"
required="required"></textarea>
                </div>
            </div>
        </div>
        <div class="row">
            <div class="col">
                <button class="btn btn-primary" type="submit">Send</button>

            </div>
        </div>
    </form>
</div>
</div>
<div class="col-md-6">
    <div class="card-body">
        <p class="mb-0"><strong>Address </strong></p>
        <p class="pb-2">235 Hudson Street, Unit 1202, Hoboken, New Jersey, 07030</p>
        <p class="mb-0"><strong>Phone</strong></p>
        <p class="pb-2">9998887776</p>
        <p class="mb-0"><strong>Email</strong></p>
        <p>johndoe@gmail.com</p>
    </div>
</div>
</div>
</div>
</div>
</div>

```

```

    </div>
  </div>
</div>
</div>
</div>
<div class="footer">
  <div class="container text-center"><a class="cc-facebook btn btn-link" target="_blank"
href="https://www.facebook.com/ayush.kalla/"><i class="fa fa-facebook fa-2x " aria-
hidden="true"></i></a><a class="cc-twitter btn btn-link " target="_blank"
href="https://twitter.com/KallaAyush"><i class="fa fa-twitter fa-2x " aria-hidden="true"></i></a><a class="cc-
google-plus btn btn-link" target="_blank" href="https://github.com/akalla123"><i class="fa fa-github fa-2x"
aria-hidden="true"></i></a><a class="cc-instagram btn btn-link" target="_blank"
href="https://www.linkedin.com/in/ayushkalla/"><i class="fa fa-linkedin fa-2x " aria-
hidden="true"></i></a></div>
  <div class="h4 title text-center">John Doe</div>
  <div class="text-center text-muted">
    <p>&copy; Creative CV. All rights reserved.<br>Design - <a class="credit"
href="https://templateflip.com" target="_blank">TemplateFlip</a></p>
  </div>
</footer>
<script src="{{ url_for('static',filename='js/core/jquery.3.2.1.min.js') }}"></script>
<script src="{{ url_for('static',filename='js/core/popper.min.js') }}"></script>
<script src="{{ url_for('static',filename='js/core/bootstrap.min.js') }}"></script>
<script src="{{ url_for('static',filename='js/now-ui-kit.js') }}"></script>
<script src="{{ url_for('static',filename='js/aos.js') }}"></script>
<script src="{{ url_for('static',filename='scripts/main.js') }}"></script>
</body>
</html>

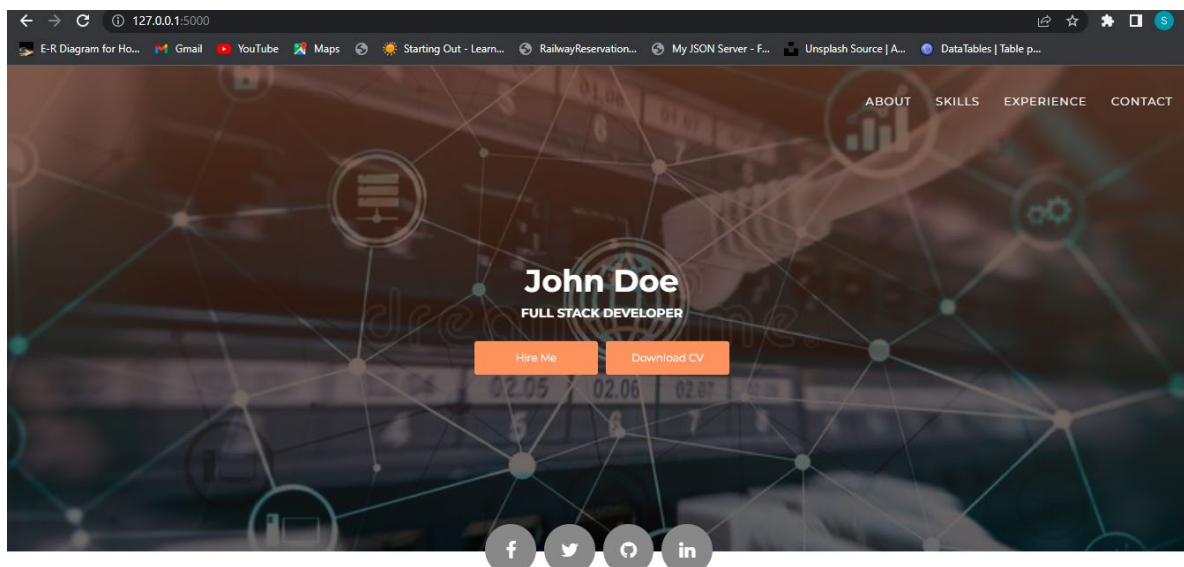
```

## ● requirements.txt

flask==1.0.3

itsdangerous==2.0.1

## Output:





## About

A Full Stack Developer is someone who works with the Back End — or server side — of the application as well as the Front End, or client side. Full Stack Developers have to have some skills in a wide variety of coding niches, from databases to graphic design and UI/UX management in order to do their job well.

## Basic Information

AGE:	20
EMAIL:	johnndoe@gmail.com
PHONE:	9998887776
ADDRESS:	Hoboken, New Jersey, U.S.A
LANGUAGE:	English

## Professional Skills

## Conclusion:

Designed a Portfolio Website using Flask.