

DEPARTMENT OF INFORMATION TECHNOLOGY ENGINEERING
LABORATORY MANUAL

314458: Laboratory Practice-II (Web Application Development)
(TE- IT Sem- II)

Sr. No.	Description
I.	Institute and Department Vision, Mission, Quality Policy, Quality Objectives, PEOs, POs and PSOs
II.	List of Experiments
1.	Creation of responsive web page using HTML, CSS, Bootstrap
2.	JavaScript Program to get the user registration data
3.	Create account on GitHub and using Git commands perform various operation on it
4.	Create Docker Container Environment.
5.	Create an angular Applications
6.	Create a Node.JS Application
7.	Create four API using Node.JS, ExpressJS and MongoDB
8.	Create a simple Mobile Website using jQuery Mobile.
9.	Deploy/Host Your web application on AWS VPC or AWS Elastic Beanstalk

Prepared By

Approved By

Date of Issue

Vision and Mission of the Institute

Vision

To create opportunities for rural students to become able engineers and technocrats through continual excellence in engineering education.

Mission

Our mission is to create self-disciplined, physically fit, mentally robust and morally strong engineers and technocrats with high degree of integrity and sense of purpose who are capable to meet challenges of ever advancing technology for the benefit of mankind and nature. We, the management, the faculty and staff, therefore promise to strive hard and commit ourselves to achieve this objective through a continuous process of learning and appreciation of needs of time.

Vision and Mission of the Department

Vision

To transfer the rural learners into competent I.T. engineers and technocrats in emerging areas of I.T. Engineering education through continual excellence for the benefit of society.

Mission

M-1: To empower the youths in rural communities to be self-disciplined, physically fit, mentally robust and morally strong I.T. professionals.

M-2: To provides cutting-edge technical knowledge through continuous process in rapidly changing environment as per need of industry and surrounding world.

M-3: To provide opportunities for intellectual and personal growth of individuals in rural platform using high quality Information Technology education.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

To train the Information Technology students-

1. To develop competent I.T. graduate with knowledge of fundamental concepts In mathematics, science, engineering and ability to provide solution to complex engineering problem by analyzing, designing and designing using modern I.T. software and tools.
2. To prepare I.T. graduate with professional skills of better communication, teamwork to manage projects in I.T. field at global level and ability to conduct investigations of complex problems using research based knowledge and research methods.
3. To develop I.T graduates with ethical practices, societal contributions through communities understanding impact of professional engineering solutions in societal and environmental context and ability of lifelong learning.

PROGRAMME SPECIFIC OUTCOMES (PSOs)

1. Apply principles of science, mathematics along with programming paradigms and problem solving skills using appropriate tools, techniques to expedite solution in I.T. domain.
2. Demonstrate core competencies related to I.T. in domain of Data structures & algorithms, Software Engineering & Modeling, Hardware, Distributed Computing, Networking & security, Databases, Discrete mathematics & algebra, Machine Learning, Operating System.
3. Demonstrate leadership qualities and professional skills in modern I.T. platform for creating innovative carrier paths in placements, entrepreneurship and higher studies

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

- **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

QUALITY POLICY

The Amrutvahini College of Engineering is committed to develop in young minds the state – of – the – art technology and high academic ambience by synergising spiritual values and technological competence continually in a learning environment.

QUALITY OBJECTIVES

- To strive hard for academic excellence and synergizing spiritual & moral values.
- To improve overall development of student.
- To enhance industry-institute interaction.
- To provide assistance for placement & entrepreneurship development.
- To promote and encourage R&D activities.

AMRUTVAHINI COLLEGE OF ENGINEERING, SANGAMNER

CM/ADL-D-05	HTML, CSS & Bootstrap	Page	01/02
Experiment No.: 01-A	Semester – II	Rev.: 00	Date: 15-06-17

Aim/Title:

Create a responsive web page which shows the ecommerce/college/exam admin dashboard with sidebar and statistics in cards using HTML, CSS and Bootstrap.

Theory:

I. 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
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc

```
<!DOCTYPE html>

<html>
  <head>
    <title> Page Title </title>
  </head>
  <body>
    <h1>My heading </h1>
    <p> First Paragraph </p>
  </body>
</html>
```

CSS

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files
- CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

Simple CSS example

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: lightblue;
}

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

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

<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

III. Bootstrap

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. It solves many problems which we had once, one of which is the cross-browser compatibility issue. Nowadays, the websites are perfect for all the browsers (IE, Firefox, and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones).

Why Bootstrap?

- Faster and Easier Web Development.
- It creates Platform-independent web pages.
- It creates Responsive Web-pages.
- It is designed to be responsive to mobile devices too.
- It is Free! Available on www.getbootstrap.com
- Simple Example

Conclusion: We have created a responsive web page using HTML, CSS, & Bootstrap.

CM/ADL-D-05	JavaScript, Ajax	Page	01/04
Experiment No.: 01-B	Semester – II	Rev.: 00	Date: 15-06-17

Aim:

Write a JavaScript Program to get the user registration data and push to array/local storage with AJAX POST method and data list in new page.

Theory:

AJAX stands for Asynchronous JavaScript and XML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script. Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display.

Conventional web applications transmit information to and from the sever using synchronous requests. It means you fill out a form, hit submit, and get directed to a new page with new information from the server.

With AJAX, when you hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server. XML is commonly used as the format for receiving server data, although any format, including plain text, can be used. AJAX is a web browser technology independent of web server software.

A user can continue to use the application while the client program requests information from the server in the background.

AJAX is based on the following open standards –

- Browser-based presentation using HTML and Cascading Style Sheets (CSS).
- Data is stored in XML format and fetched from the server.
- Behind-the-scenes data fetches using XMLHttpRequest objects in the browser.
- JavaScript to make everything happen

AJAX cannot work independently. It is used in combination with other technologies to create interactive webpages.

• JavaScript

- Loosely typed scripting language.
- JavaScript function is called when an event occurs in a page.
- Glue for the whole AJAX operation.

• DOM

- API for accessing and manipulating structured documents.
- Represents the structure of XML and HTML documents.

• CSS

- Allows for a clear separation of the presentation style from the content and may be changed programmatically by JavaScript

• XMLHttpRequest

- JavaScript object that performs asynchronous interaction with the server.

CM/ADL-D-05	JavaScript, Ajax	Page	02/04
Experiment No.: 01-B	Semester – II	Rev.: 00	Date: 15-06-17

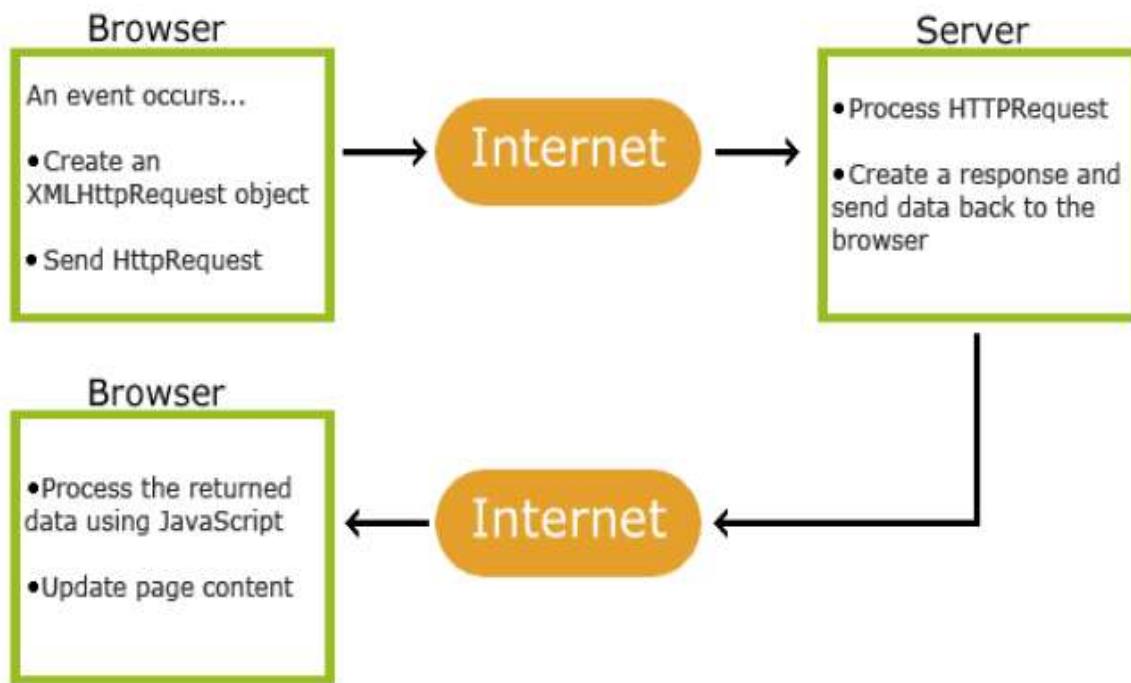


Fig 1. How AJAX works

AJAX – Events : onreadystatechange Event Properties

Property	Description
onReadyStateChange	It is called whenever readystate attribute changes. It must not be used with synchronous requests.
readyState	represents the state of the request. It ranges from 0 to 4. <ul style="list-style-type: none"> • 0: request not initialized (open() is not called.) • 1: server connection established (open is called but send() is not called.) • 2: request received (send() is called, and headers and status are available.) • 3: processing request (Downloading data; responseText holds the data.) • 4: request finished and response is ready (The operation is completed fully.)
Status	200: "OK" 403: "Forbidden" 404: "Page not found"

CM/ADL-D-05	JavaScript, Ajax	Page	03/04
Experiment No.: 01-B	Semester – II	Rev.: 00	Date: 15-06-17

XMLHttpRequest object properties

Property	Description
readyState	An integer from 0 . . . 4. (0 means the call is uninitialized, 4 means that the call is complete.)
onreadystatechange	Determines the function called when the objects readyState changes.
responseText	Data returned from the server as a text string (read-only).
responseXML	Data returned from the server as an XML document object (read-only).
status	HTTP status code returned by the server
statusText	HTTP status phrase returned by the server

XMLHttpRequest object methods

Method	Description
open('method', 'URL', asyn)	Specifies the HTTP method to be used (GET or POST as a string, the target URL, and whether or not the request should be handled asynchronously (asyn should be true or false, if omitted, true is assumed).
send(content)	Sends the data for a POST request and starts the request, if GET is used you hould call send(null).
setRequestHeader('x','y')	Sets a parameter and value pair x=y and assigns it to the header to be sent with the request.
getAllResponseHeaders()	Returns all headers as a string.
getResponseHeader(x)	Returns header x as a string.
abort()	Stops the current operation.

CM/ADL-D-05	JavaScript, Ajax	Page	04/04
Experiment No.: 01-B	Semester – II	Rev.: 00	Date: 15-06-17

Sample code:

ajaxcommunication.html

```
<html>
  <body>
    <div id="xyz">
      Hello Friends <br>
      Welcome to Pune!!!!<br>
      <button type="button" onclick="load()">
        Submit
      </button>
    </div>
    <script>
      function load(){
        var req=new XMLHttpRequest()
        req.onreadystatechange=function() {
          if(this.readyState == 4 && this.status == 200){
            document.getElementById("xyz").innerHTML=this.responseText
          }
        }
        req.open('GET','data.txt',true)
        req.send()
      }
    </script>
  </body>
</html>
```

Data.txt

I am enjoying learning JavaScript!!!!!!

Conclusion: We have created a registration form using javascript & store data to local storage using ajax post method.

AMRUTVAHINI COLLEGE OF ENGINEERING, SANGAMNER

CM/ADL-D-010	GitHub Repository	Page	01/03
Experiment No.: 02-A	Semester – II	Rev.: 00	Date: 15-06-17

Aim:

Create version control account on GitHub and using Git commands to create repository and push your code to GitHub.

Theory:

1. What is Git?

Git is a popular version control system. It was created by Linus Torvalds in 2005, and has been maintained by Junio Hamano since then.

It is used for:

- Tracking code changes
- Tracking who made changes
- Coding collaboration

2. What does Git do?

- Manage projects with Repositories
- Clone a project to work on a local copy
- Control and track changes with Staging and Committing
- Branch and Merge to allow for work on different parts and versions of a project
- Pull the latest version of the project to a local copy
- Push local updates to the main project

3. Working with Git

- Initialize Git on a folder, making it a Repository
- Git now creates a hidden folder to keep track of changes in that folder
- When a file is changed, added or deleted, it is considered modified
- You select the modified files you want to Stage
- The Staged files are Committed, which prompts Git to store a permanent snapshot of the files
- Git allows you to see the full history of every commit.
- You can revert back to any previous commit.
- Git does not store a separate copy of every file in every commit, but keeps track of changes made in each commit!

4. Why Git?

- Over 70% of developers use Git!
- Developers can work together from anywhere in the world.
- Developers can see the full history of the project.
- Developers can revert to earlier versions of a project.

CM/ADL-D-010	GitHub Repository	Page	02/03
Experiment No.: 02-A	Semester – II	Rev.: 00	Date: 15-06-17

5. What is GitHub

- Git is not the same as GitHub.
- GitHub makes tools that use Git.
- GitHub is the largest host of source code in the world, and has been owned by Microsoft since 2018.

6. Steps to Push and PULL version control repository to GitHub

Step No	Command	Description
1	Git Installation	Download Git from the website: https://www.git-scm.com/
2	Command line >git –version	If Git is installed, it should show something like git version X.Y
3	git config --global user.name "w3schools-test" git config --global user.email "test@w3schools.com"	Configure Git Change the user name and e-mail address to your own
4	mkdir myproject cd myproject	Creating Git Folder
5	git init	Initialize Git Initialized empty Git repository in /Users/user/myproject/.git/
6	git status	To check the status
7	git add index.html	Add file to staging environment
8	git add –all	add all files in the current directory to the Staging Environment:
9	git commit -m "First release of Hello World!"	The committ command performs a commit, and the -m "message" adds a message.
10	git commit -a -m "Updated index.html with a new line"	Skips staging environment
11	git log	To view the history of commits for a repository, you can use the log command
12	git <i>command</i> -help	See all the available options for the specific command
13	git help --all	See all possible commands
14	git commit -help	See help for specific command
15	git branch hello-world-images	a branch is a new/separate version of the main

AMRUTVAHINI COLLEGE OF ENGINEERING, SANGAMNER

CM/ADL-D-010	GitHub Repository	Page	03/03
Experiment No.: 02-A	Semester – II	Rev.: 00	Date: 15-06-17
		repository. This command creates a new branch hello-world-images	

16	git checkout hello-world-images	checkout is the command used to check out/ move to a branch
17	git checkout master	Used to switch between branches
18	https://github.com/	Create a new account on github
19		Create a Repository on GitHub
20	git remote add origin https://github.com/w3schools-test/hello-world.git	Push Local Repository to GitHub
21	git push --set-upstream origin master	push master branch to the origin url,
22		go back into GitHub and see that the repository has been updated:
23	git fetch origin	fetch gets all the change history of a tracked branch/repo
24	git merge origin/master	merge combines the current branch, with a specified branch.
25	git pull origin	pull is a combination of fetch and merge It is used to pull all changes from a remote repository into the branch you are working on.

Conclusion: We have created a version control account on GitHub and by using Git commands push the code to GitHub.

AMRUTVAHINI COLLEGE OF ENGINEERING, SANGAMNER

CM/ADL-D-09	Angular Application	Page	01/03
Experiment No.: 02-C	Semester – II	Rev.: 00	Date: 15-06-17

Aim/Title:

Create an Angular application which will do following actions: Register User, Login User, Show User Data on Profile Component.

Theory:

1. Angular requires a current, active LTS(long term support) or maintenance LTS version of Node.js and NPM.

install node.js <https://nodejs.org/>
It will automatically install NPM - node package manager

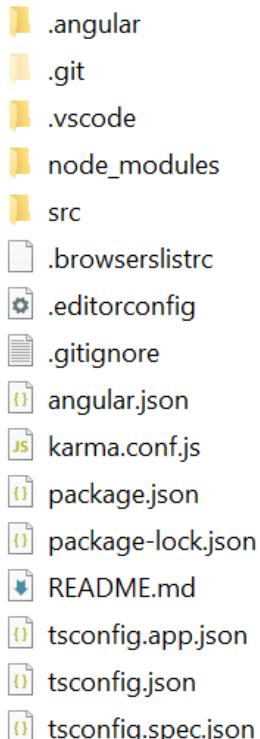
2. Install Angular CLI

npm install -g @angular/cli@latest
To Create Angular 2 Application Angular CLI is required

3. To create new project through CLI go to folder of the new project Give command as -

ng new project-name
press ENTER

The project will be created as directory structure below –



AMRUTVAHINI COLLEGE OF ENGINEERING, SANGAMNER

CM/ADL-D-09	Angular Application	Page	02/03
Experiment No.: 02-C	Semester – II	Rev.: 00	Date: 15-06-17

Open folder src/app

Modify app.module.ts for form application

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import {FormsModule} from '@angular/forms'

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule,
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule {}
```

Open app.component.html

Write html code for form (representative code is mentioned here, modify for multiple inputs)

```
<h1>Simple Form</h1>
<form #simpleForm = "ngForm" (ngSubmit) = "getValues(simpleForm.value)">
  <input type ="text" ngModel name = "user" placeholder = "Enter Name">
  <br> <br>
  <input type ="text" ngModel name = "age" placeholder = "Enter age">
  <br> <br>
  <input type ="text" ngModel name = "city" placeholder = "Enter city">
  <br> <br>
  <button>Get user value</button>
</form>
```

AMRUTVAHINI COLLEGE OF ENGINEERING, SANGAMNER

CM/ADL-D-09	Angular Application	Page	03/03
Experiment No.: 02-C	Semester – II	Rev.: 00	Date: 15-06-17

Make changes in app.component.ts

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'AngProj1';
  getValues(val:any)
  {
    console.log(val);
  }
}
```

Here getValue() function which is called in form file is defined.
You can check inputted values through form in console.

Build Application

1. Use Angular CLI command `ng serve -o` to build an application.
The `-o` indicates to open it automatically in the default browser.
2. Use NPM command '`npm start`' to build an application
<http://localhost:4200> to see the application home page.
3. Open the terminal in VS Code from menu Terminal -> New Terminal, and type `ng serve -o` command and press enter,

You can send the form contents from console to other page.
On the basis of above implementation, you can design login user, show user data.

Conclusion: We have created a angular application which shows registration, login, & profile page portal.

CM/ADL-D-13	NodeJs Application	Page	01/05
Experiment No.: 03-A	Semester – II	Rev.: 00	Date: 15-06-17

Aim: Create a Node.JS Application which serves a static website.

Installation : Node.js (site – Node.js), Express.js(installed through cmd)

Node.js overview

In basic terms, Node is an open source cross-platform library for server-side programming that permits clients to develop web applications rapidly. With Node, we can execute JavaScript applications or network applications. Its basic modules are engraved in JavaScript.

It is generally utilized for server applications in real-time. Node.js permits JavaScript to execute locally on a machine or a server.

Node.js gives numerous systems to utilize. One of such structures is Express.js. It is more valuable and mainstream than the different structures of Node.js.

Features of Node.js

- **Versatility:** Node is incredibly adaptable as the server reacts in a non-blocking way.
- **Zero Buffering:** Applications yield the measurements in enormous pieces. This gives the advantage of ‘No buffering’ to developers.
- **Network:** Node.js upholds an open-source community. This is the main explanation that numerous glorious modules have been added to Node.js applications over time.
- **Occasion driven Input and output:** APIs of Node.js are non-blocking, meaning that the server won’t wait for the arrival of information from an API. Rather, it will move to another API.

Advantages of Node.js

- **Easy to learn:** Node is quite simple for developers to utilize and learn. Learning Node.js is less difficult than React.
- **Better Performance:** Node.js takes the code of JavaScript via Google’s V8 JavaScript engine. The main advantage of this process is that it complies with the JavaScript code directly into the machine code
- **Freedom:** Node.js offers a lot of freedom when it comes to development. There are generally less constraints with Node.
- **Extended support for tools:** Another advantage of Node.js is that developers have more community support.
- **Extensible:** Node.js is known to be quite extensible. You can utilize JSON to give the degree to trade of information between the web server and the client.

CM/ADL-D-13	NodeJs Application	Page	02/05
Experiment No.: 03-A	Semester – II	Rev.: 00	Date: 15-06-17

- **Scalability:** Node.js makes it simple to scale applications in horizontal as well as vertical directions. The applications can be scaled even by the option of extra hubs to the current framework.

Limitations of Node.js

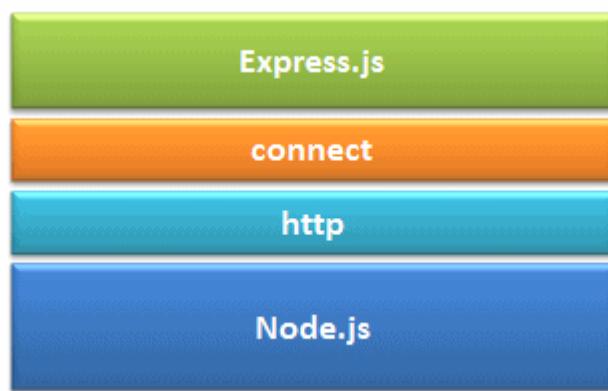
- **Programming interface isn't steady:** The Application Programming Interface (API) of Node can be challenging to work with. It changes regularly and doesn't remain stable.
- **No strong library support system:** JavaScript does not hold a strong library system. This limits the developers to implement even common programming tasks using Node.js.
- **Programming model is not synchronous:** Many developers find this programming model tougher in comparison to linear blocking I/O programming. In asynchronous programming, the codes become clumsier, and developers have to depend on the nes

Express.js

"Express is a fast, unopinionated minimalist web framework for Node.js" - official web site: Expressjs.com

Express.js is a web application framework for Node.js. It provides various features that make web application development fast and easy which otherwise takes more time using only Node.js.

Express.js is based on the Node.js middleware module called **connect** which in turn uses **http** module. So, any middleware which is based on connect will also work with Express.js.



Advantages of Express.js

1. Makes Node.js web application development fast and easy.
2. Easy to configure and customize.
3. Allows you to define routes of your application based on HTTP methods and URLs.
4. Includes various middleware modules which you can use to perform additional tasks on request and response.
5. Easy to integrate with different template engines like Jade, Vash, EJS etc.
6. Allows you to define an error handling middleware.

CM/ADL-D-13	NodeJs Application	Page	03/05
Experiment No.: 03-A	Semester – II	Rev.: 00	Date: 15-06-17

7. Easy to serve static files and resources of your application.
8. Allows you to create REST API server.
9. Easy to connect with databases such as MongoDB, Redis, MySQL

Steps:

1. Install Node.js
2. Setting up express.js
3. Structuring files
4. Creating your express server
5. Servicing your static files
6. Building your web page
7. Running your project

Open Node.js command terminal & run the following in your terminal-

Setting up express.js

1. Create a new directory for your project - mkdir your-project-name
2. Change into your new directory - cd your-project-name
3. Initialize a new Node project with defaults. This will set a package.json file to access your dependencies: npm init -y
4. Create your entry file, index.js. This is where you will store your Express server: if you are working on Linux, you can run : touch index.js. if you are working on windows, you can edit in VSCode
5. Install Express as a dependency : npm install express –save
6. Edit package.json. Within your package.json, update your start script to include node and your index.js file.

Let express-static-file-tutorial is your project name

Package.json

```
{  
  "name": "express-static-file-tutorial",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "start": "node index.js" // change start value as node index.js  
  }, // This will allow you to use the npm start command in your terminal to  
  launch  
  "keywords": [], // your express server  
  "author": "Paul Halliday",  
  "license": "MIT"}
```

CM/ADL-D-13	NodeJs Application	Page	04/05
Experiment No.: 03-A	Semester – II	Rev.: 00	Date: 15-06-17

Structuring Your Files

To store your files on the client-side, create a public directory and include an index.html file
express-static-file-tutorial

```
|- index.js
|- public
  |- index.html
```

Creating Your Express Server

Edit index.js file

Index.js

```
const express = require('express');
const app = express();
const PORT = 3000;

app.use(express.static('public')) // represents application is serving static
//webpage in public directory

app.get('/', (req, res) => {
  res.send('Hello World!');
});

app.listen(PORT, () => console.log(`Server listening on port: ${PORT}`));
```

First of all, import the Express.js module.

In the above example, we imported Express.js module using require() function. The express module returns a function. This function returns an object which can be used to configure Express application (app in the above example).

The app object includes methods for routing HTTP requests, configuring middleware, rendering HTML views and registering a template engine.

The app.listen() function creates the Node.js web server at the specified host and port. It is identical to Node's http.Server.listen() method. Instead of Get(), post(), put() and delete() methods can be used.

CM/ADL-D-13	NodeJs Application	Page	05/05
Experiment No.: 03-A	Semester – II	Rev.: 00	Date: 15-06-17

Building Your Web Page – client side

Navigate to your index.html file in the public directory. Populate the file with body and image elements:

[label index.html]

```
<html>
<head>
<title>Hello World!</title>
</head>
<body>
<h1>Hello, World!</h1>
 //download & store image in public directory
</body>
</html>
```

(Instead of building Hello world application, building applications like student's Registration form/main page of website is recommended)

Running Your Project

In your terminal, launch your Express project

npm start

It will display

Server listening on port : 3000

Open your web browser, and navigate to <http://localhost:3000>.

Conclusion: Thus we have created a Node.JS Application which serves a static website.

CM/ADL-D-11	Create a simple Mobile Website using jQuery Mobile.	Page	01/02
Experiment No.: 04-A	Semester – II	Rev.: 00	Date: 15-06-17

Aim: Create a simple Mobile Website using jQuery Mobile.

Theory:

jQuery Mobile

JQuery Mobile is a user interface framework, built on jQuery Core and used for developing responsive websites or applications that are accessible on mobile, tablet, and desktop devices. It uses features of both jQuery and jQueryUI to provide API features for mobile web applications. This tutorial will teach you the basics of jQuery Mobile framework. We will also discuss some detailed concepts related to jQuery Mobile.

Why Use jQuery Mobile?

- It creates web applications that it will work the same way on the mobile, tablet, and desktop devices.
- It is compatible with other frameworks such as PhoneGap, Whitelight, etc.
- It provides a set of touch-friendly form inputs and UI widgets.

Features of jQuery Mobile

- It is built on jQuery Core and "write less, do more" UI framework.
- It is an open source framework, and cross-platform as well as cross-browser compatible.
- It is written in JavaScript and uses features of both jQuery and jQuery UI for building mobile-friendly sites.

Download jQuery Mobile

When you open the link <https://jquerymobile.com/>, you will see there are two options to download jQuery mobile library.

CM/ADL-D-11	Create a simple Mobile Website using jQuery Mobile.	Page	02/02
Experiment No.: 04-A	Semester – II	Rev.: 00	Date: 15-06-17



Click the *Stable* button, which leads directly to a ZIP file containing the CSS and JQuery files, for the latest version of jQuery mobile library. Extract the ZIP file contents to a jQuery mobile directory.

This version contains all files including all dependencies, a large collection of demos, and even the library's unit test suite. This version is helpful to getting started.

Conclusion: Thus we have created a simple Mobile Website using jQuery Mobile.

CM/ADL-D-11	Deploy/Host Your web application on AWS VPC or AWS Elastic Beanstalk.	Page	01/07
Experiment No.: 04-B	Semester – II	Rev.: 00	Date: 15-06-17

Aim: Deploy/Host Your web application on AWS VPC or AWS Elastic Beanstalk.

Theory:

What is Cloud Computing?

Cloud computing is a term referred to storing and accessing data over the internet. It doesn't store any data on the hard disk of your personal computer. In cloud computing, you can access data from a remote server.

What is AWS?

The full form of AWS is Amazon Web Services. It is a platform that offers flexible, reliable, scalable, easy-to-use and, cost-effective cloud computing solutions.

AWS is a comprehensive, easy to use computing platform offered Amazon. The platform is developed with a combination of infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS) offerings.

It is a secure cloud services platform, offering compute power, database storage, content delivery and other functionality to help businesses scale and grow.

In simple words AWS allows you to do the following things- Running web and application servers in the cloud to host dynamic websites.

History of AWS

- 2002- AWS services launched
- 2006- Launched its cloud products
- 2012- Holds first customer event
- 2015- Reveals revenues achieved of \$4.6 billion
- 2016- Surpassed \$10 billion revenue target
- 2016- Release snowball and snowmobile
- 2019- Offers nearly 100 cloud services
- 2021- AWS comprises over 200 products and services

Important AWS Services

Amazon Web Services offers a wide range of different business purpose global cloud-based products. The products include storage, databases, analytics, networking, mobile, development tools, enterprise applications, with a pay-as-you-go pricing model.

CM/ADL-D-11	Deploy/Host Your web application on AWS VPC or AWS Elastic Beanstalk.	Page	02/07
Experiment No.: 04-B	Semester – II	Rev.: 00	Date: 15-06-17

Applications of AWS services

Amazon Web services are widely used for various computing purposes like:

- Web site hosting
- Application hosting/SaaS hosting
- Media Sharing (Image/ Video)
- Mobile and Social Applications
- Content delivery and Media Distribution
- Storage, backup, and disaster recovery
- Development and test environments
- Academic Computing
- Search Engines
- Social Networking

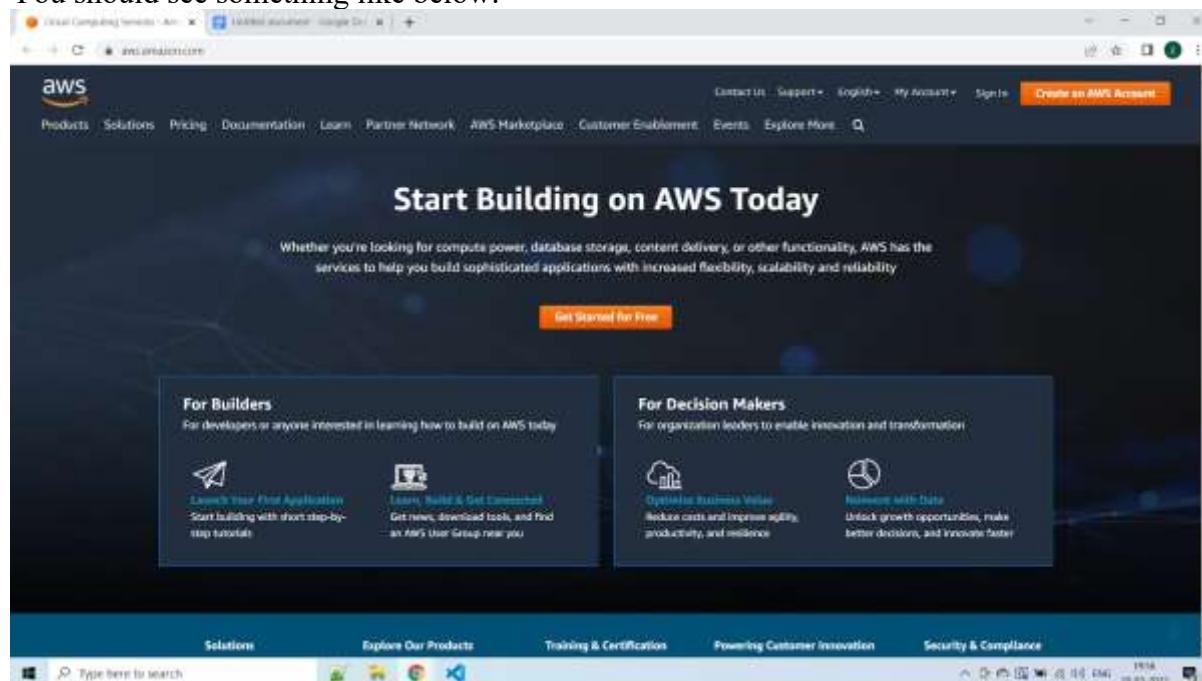
Creating an AWS Account is the first step you need to take in order to learn **Amazon Web Services**.

Steps to follow are as follows :

Step 1 – Visiting the Signup Page

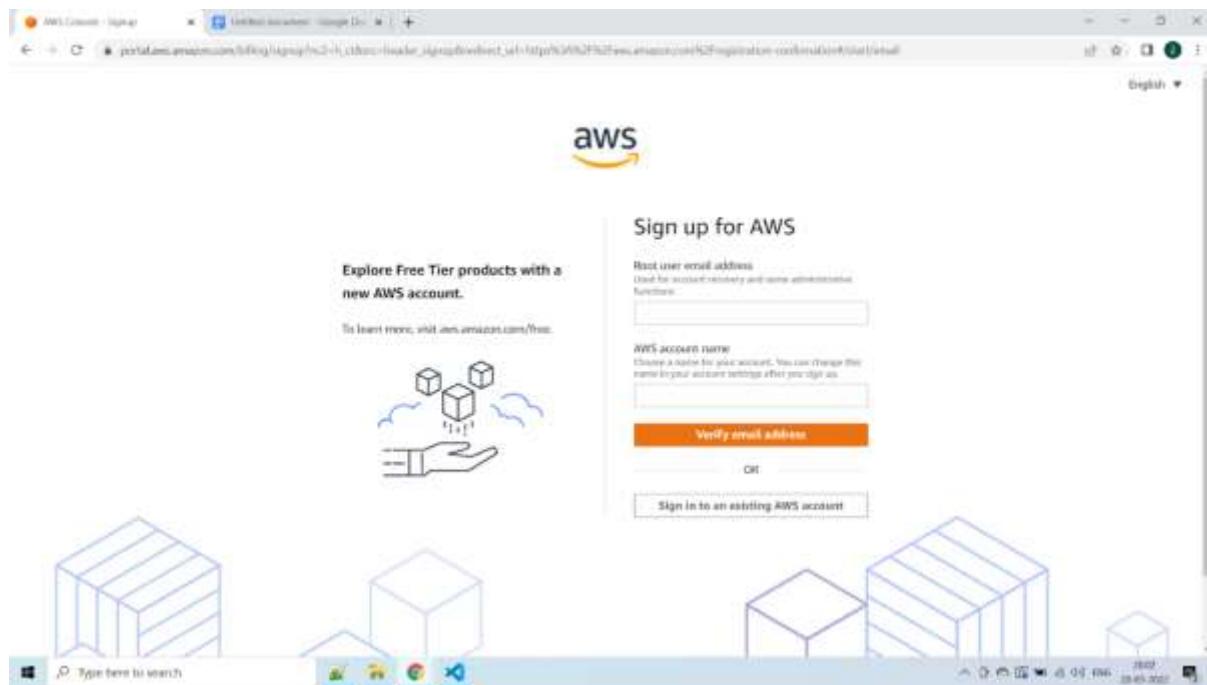
Go to <https://aws.amazon.com>

You should see something like below:



In order to continue, click the **Complete Sign Up** button in the middle of the screen or on the top right corner of the screen. You will see the below screen.

CM/ADL-D-11	Deploy/Host Your web application on AWS VPC or AWS Elastic Beanstalk.	Page	03/07
Experiment No.: 04-B	Semester – II	Rev.: 00	Date: 15-06-17



Step 2 – Entering User Details

After you have chosen to **Create a new AWS account**, you will see the below screen asking for few details.

Create an AWS account	
AWS Accounts Include 12 Months of Free Tier Access <small>Including use of Amazon EC2, Amazon S3, and Amazon DynamoDB. Visit aws.amazon.com/free for full offer terms.</small>	
<input style="width: 100%; height: 20px; border: 1px solid #ccc; margin-bottom: 5px;" type="text"/> <input style="width: 100%; height: 20px; border: 1px solid #ccc; margin-bottom: 5px;" type="password"/> <input style="width: 100%; height: 20px; border: 1px solid #ccc; margin-bottom: 5px;" type="password"/> <input style="width: 100%; height: 20px; border: 1px solid #ccc; margin-bottom: 5px;" type="text"/>	<input style="width: 100%; height: 30px; background-color: #ff9900; color: white; border: none; font-weight: bold; margin-bottom: 5px;" type="button" value="Continue"/> Sign in to an existing AWS account
<small>© 2013 Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Service</small>	

You can fill up the details as per your requirements and click **Continue**.

Next you will be asked to fill up your contact details such contact number, country, address and so on. You should fill them up properly because your contact number is important for further steps.

AMRUTVAHINI COLLEGE OF ENGINEERING, SANGAMNER

CM/ADL-D-11	Deploy/Host Your web application on AWS VPC or AWS Elastic Beanstalk.	Page	04/07
Experiment No.: 04-B	Semester – II	Rev.: 00	Date: 15-06-17

Contact Information

Please select the account type and complete the fields below with your contact details:

Account type Professional Personal

Full name: Saurabh Chaitanya

Phone number: _____

Country/Region: India

Address: 1000 ITI, Dulewadi, Pune, Maharashtra, India

City: _____

State / Province or Region: _____

Postal Code: _____

I acknowledge that you have read and agree to the terms of the AWS Customer Agreement.

Create Account and Continue

After filling up the details, click on the **Create Account and Continue** button at the bottom of the form.

Step 3 – Filling up the Credit Card details

For **Creating an AWS Account**, you need to enter your **Credit Card** details.

Payment Information

Please type your payment information so we can verify your identity. We will not charge you unless your usage exceeds the \$0.00 Free Tier Limit. Review frequently asked questions for more information.

Important: As part of our card verification process, we will charge \$0.00 on your card when you click the "Secure Submit" button below. This will be refunded once your card has been validated. Your bank may take 3-5 business days to issue the refund. Please note this amount may be deducted by your bank or held by a authorized for charge.

Credit/Debit card number: _____

Expiration date: 08 / 2020

Cardholder's name: _____

After entering the details, click on **Secure Submit** button. It might take a while to process the request depending on your bank/credit card company servers.

Step 4 – Identity Confirmation

Once the credit card details are confirmed, you will need to complete the **Identity Confirmation** step. You will see the below screen:

AMRUTVAHINI COLLEGE OF ENGINEERING, SANGAMNER

CM/ADL-D-11	Deploy/Host Your web application on AWS VPC or AWS Elastic Beanstalk.	Page	05/07
Experiment No.: 04-B	Semester – II	Rev.: 00	Date: 15-06-17

Confirm your identity

Before you can use your AWS account, you must verify your phone number. When you confirm, the AWS account system will contact you with a verification code.

How should we send you the verification code?

Text message (SMS) VISA call

Country or region code

India (+91)

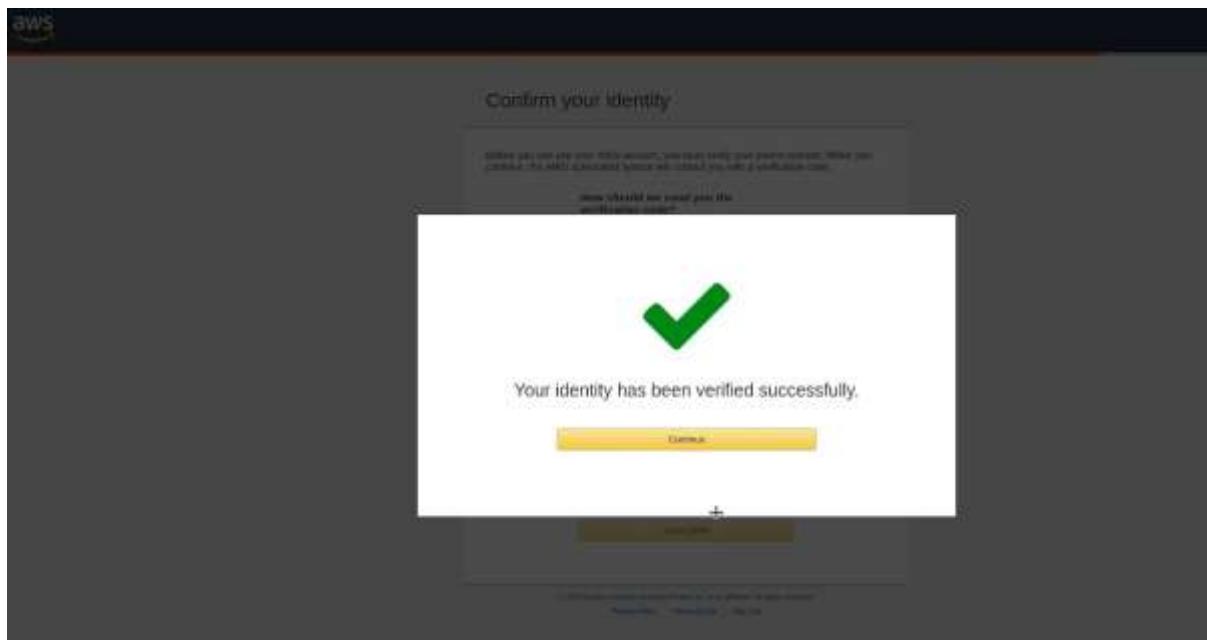
Cell Phone Number

Security check

y5785c

Next Step

Once you have verified successfully, you should see a screen like below:



CM/ADL-D-11	Deploy/Host Your web application on AWS VPC or AWS Elastic Beanstalk.	Page	06/07
Experiment No.: 04-B	Semester – II	Rev.: 00	Date: 15-06-17

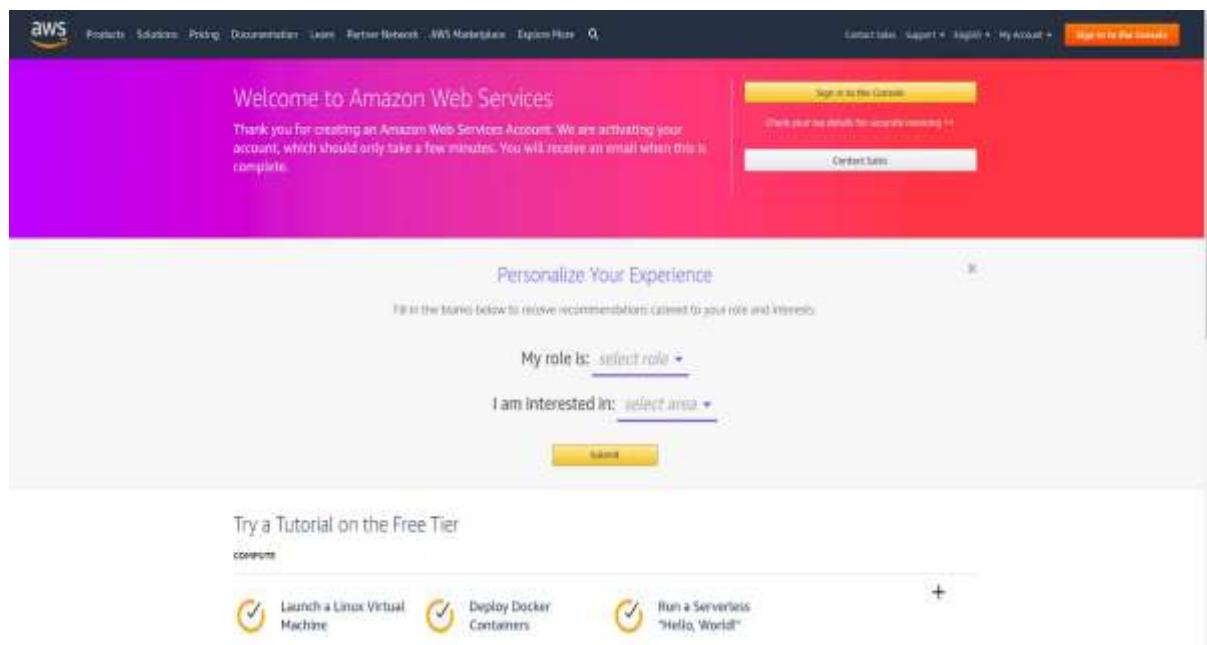
Step 5 – Selecting a Support Plan



Go for **Basic Plan**. It is Free of cost and great for learning purposes.

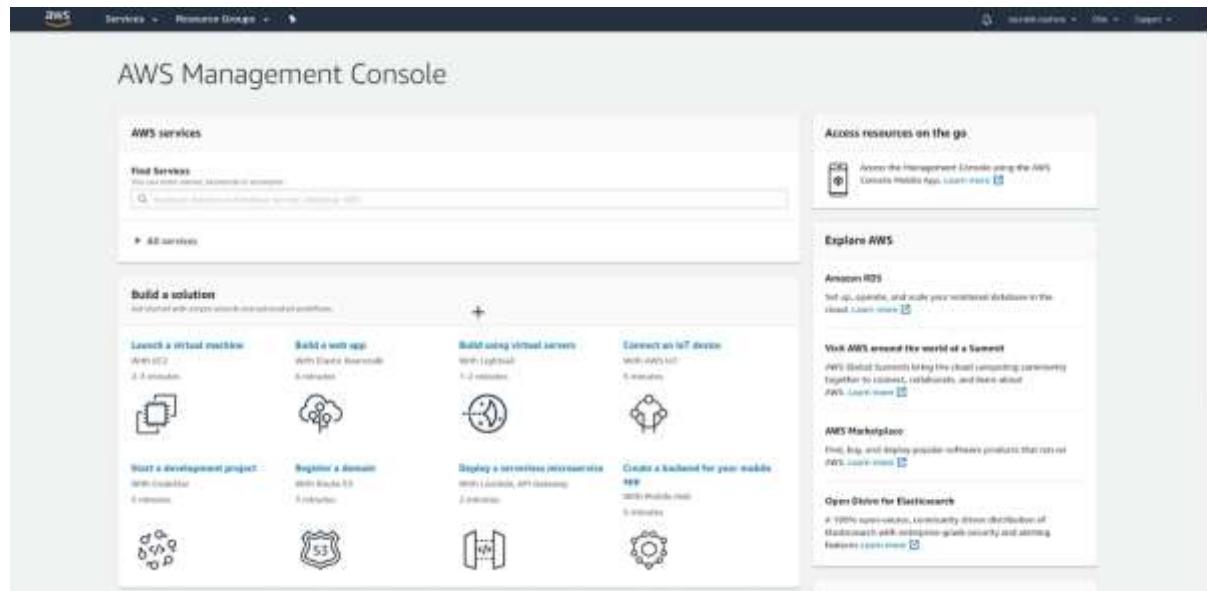
The other plans are **Developer Plan** and a **Business Plan**. But both of them are paid options.

Once you select your plan, you will see the below **Welcome** screen. From here on, you can Sign in to your **AWS Console**.



Finally, after logging in, you should be able to see the **AWS Management Console** as below:

CM/ADL-D-11	Deploy/Host Your web application on AWS VPC or AWS Elastic Beanstalk.	Page	07/07
Experiment No.: 04-B	Semester – II	Rev.: 00	Date: 15-06-17



If you have reached this far, you have successfully finished **Creating an AWS Account**.

Deployment Steps:

Step 1: Launch a Windows Server Amazon EC2 instance.

Step 2: Configure your source content to deploy to the Windows Server Amazon EC2 instance.

Step 3: Upload your "hello, world!" ...

Step 4: Deploy your Hello World application.

Step 5: Update and redeploy your "hello, world!" ...

Step 6: Clean up your "hello, world!"

Conclusion : We have created a web application & deploy it on AWS VPC or AWS Elastic Beanstalk.