314458: Laboratory Practice-II (Web Application Development)

TE-IT (2019 Course)

Semester – VI

Teaching Scheme	Credit Scheme:	Examination Scheme	
Practical :	02 Credits	Term work :	50 Marks
4 Hrs. / Week		Practical :	25 Marks

Sinhgad Institutes

LABORATORY MANUAL V 1.0

DEPARTMENT OF INFORMATION TECHNOLOGY

SMT. KASHIBAI NAVALE COLLEGE OF ENGINEERING, PUNE

2022-2023

VISION

To provide excellent Information Technology education by building strong teaching and research environment.

MISSION

1) To transform the students into innovative, competent and high quality IT professionals to meet the growing global challenges.
2) To achieve and impart quality education with an emphasis on practical skills and social relevance.
3) To endeavour for continuous up-gradation of technical expertise of students to cater to the needs of the society.
4) To achieve an effective interaction with industry for mutual benefits.

PROGRAM EDUCATIONAL OBJECTIVES

The students of Information Technology course after passing out will

PEO1: Possess strong fundamental concepts in mathematics, science, engineering and Technology to address technological challenges.

PEO2: Possess knowledge and skills in the field of Computer Science and Information Technology for analysing, designing and implementing complex engineering problems of any domain with innovative approaches.

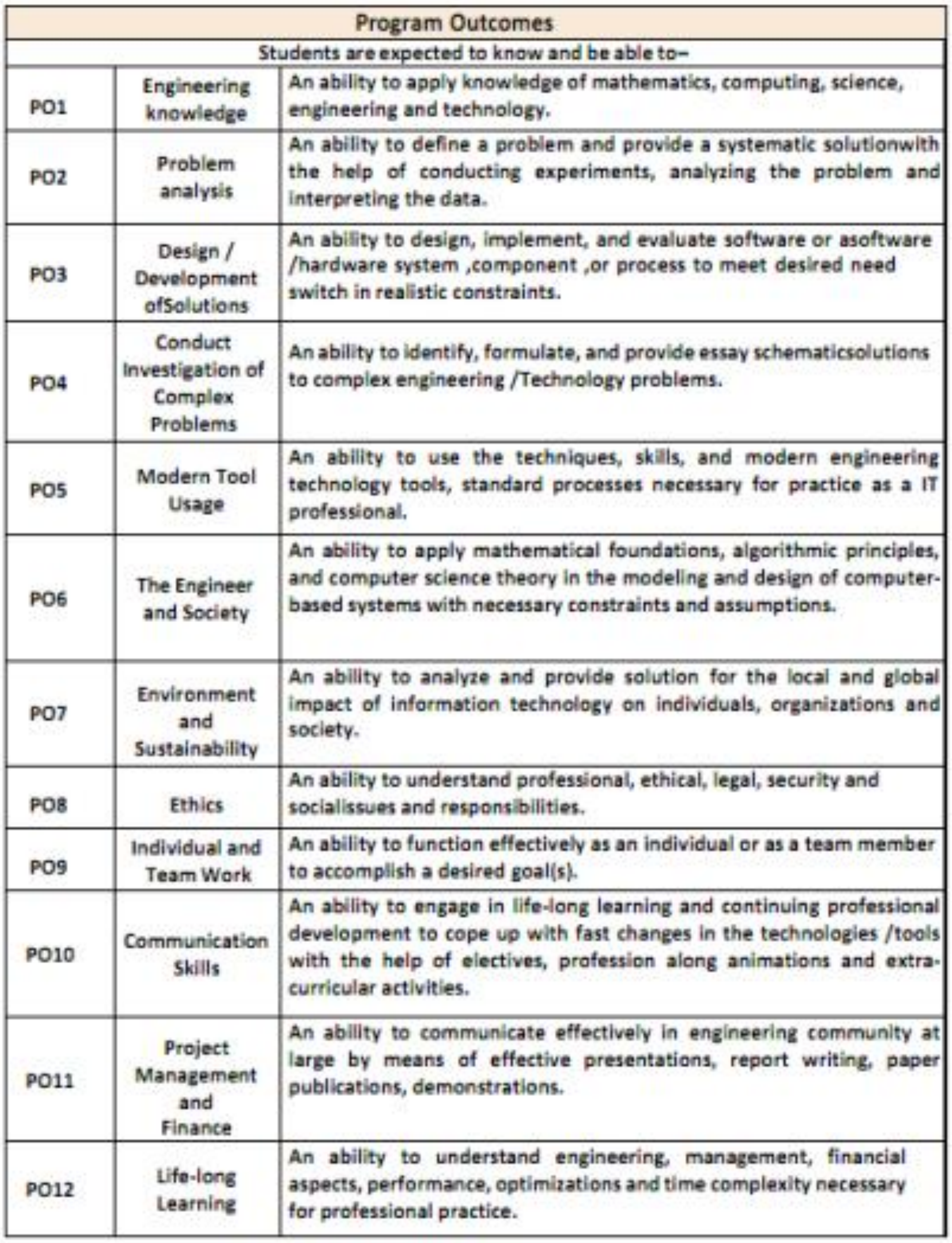
PEO3: Possess an attitude and aptitude for research, entrepreneurship and higher studies in the field of Computer Science and Information Technology.

PEO4: Have commitment ethical practices, societal contributions through communities and life-long learning.

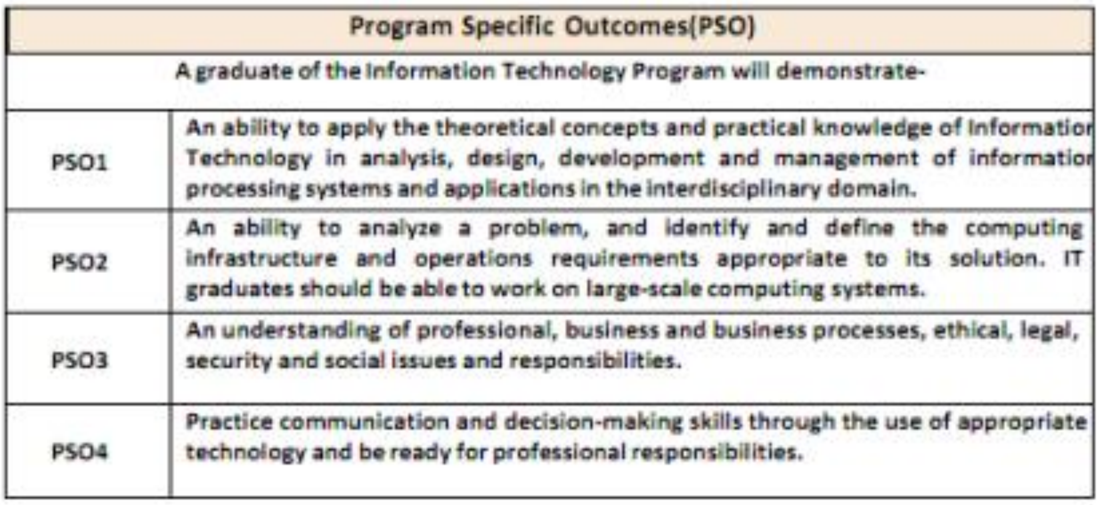
PEO5: Possess better communication, presentation, time management and team work skills leading to responsible & competent professional sand will be able to address challenges in the field of IT at global level.

PROGRAM OUTCOMES

The students in the Information Technology course will attain:



Program Outcomes		
Students are expected to know and be able to–		
PO1	Engineering knowledge	An ability to apply knowledge of mathematics, computing, science, engineering and technology.
PO2	Problem analysis	An ability to define a problem and provide a systematic solutionwith the help of conducting experiments, analyzing the problem and interpreting the data.
PO3	Design / Development ofSolutions	An ability to design, implement, and evaluate software or asoftware /hardware system ,component ,or process to meet desired need switch in realistic constraints.
PO4	Conduct Investigation of Complex Problems	An ability to identify, formulate, and provide essay schematicsolutions to complex engineering /Technology problems.
PO5	Modern Tool Usage	An ability to use the techniques, skills, and modern engineering technology tools, standard processes necessary for practice as a IT professional.
PO6	The Engineer and Society	An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems with necessary constraints and assumptions.
PO7	Environment and Sustainability	An ability to analyze and provide solution for the local and global impact of information technology on individuals, organizations and society.
PO8	Ethics	An ability to understand professional, ethical, legal, security and socialissues and responsibilities.
PO9	Individual and Team Work	An ability to function effectively as an individual or as a team member to accomplish a desired goal(s).
PO10	Communication Skills	An ability to engage in life-long learning and continuing professional development to cope up with fast changes in the technologies /tools with the help of electives, profession along animations and extra-curricular activities.
PO11	Project Management and Finance	An ability to communicate effectively in engineering community at large by means of effective presentations, report writing, paper publications, demonstrations.
PO12	Life-long Learning	An ability to understand engineering, management, financial aspects, performance, optimizations and time complexity necessary for professional practice.



Program Specific Outcomes(PSO)	
A graduate of the Information Technology Program will demonstrate-	
PSO1	An ability to apply the theoretical concepts and practical knowledge of Informatio Technology in analysis, design, development and management of informatio processing systems and applications in the interdisciplinary domain.
PSO2	An ability to analyze a problem, and identify and define the computing infrastructure and operations requirements appropriate to its solution. IT graduates should be able to work on large-scale computing systems.
PSO3	An understanding of professional, business and business processes, ethical, legal, security and social issues and responsibilities.
PSO4	Practice communication and decision-making skills through the use of appropriate technology and be ready for professional responsibilities.

Compliance

Document Control

Reference Code	SKNCOE-IT / Lab Manual Procedures
Version No	1.0
Compliance Status	Complete
Revision Date	20 Sep 2021
Security Classification	Department Specific
Document Status	Definitive
Review Period	Yearly

Authors

Mrs. A. S. Shinde

Assistant Professor (I.T.)

Mrs. A. Y. Kadam

Assistant Professor(I.T.)

INDEX

Sr. No.	Title	Page No.
1	**Assignment 1** **a.** Create a responsive web page which shows the ecommerce/college/exam admin dashboard with sidebar and statistics in cards using HTML, CSS and Bootstrap. **b.** 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.	6
2	**Assignment 2** **a.** Create version control account on GitHub and using Git commands to create repository and push your code to GitHub. **b.** Create Docker Container Environment (NVIDEIA Docker or any other). **c.** Create an Angular application which will do following actions: Register User, Login User, Show User Data on Profile Component	15
3	**Assignment 3** **a.** Create a Node.JS Application which serves a static website. **b**. Create four API using Node.JS, ExpressJS and MongoDB for CURD Operations on assignment 2.C.	23
4	**Assignment 4** **a.** Create a simple Mobile Website using jQuery Mobile. **b.** Deploy/Host Your web application on AWS VPC or AWS Elastic Beanstalk. Mini Project Develop a web application using full stack development technologies in any of the following domains: 1. Social Media 2. ecommerce 3. Restaurant 4. Medical 5. Finance 6. Education 7. Any other Reference	30

Assignment 1(a):

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:

1. HTML

- HTML stands for Hyper Text Mark-up Language
- HTML is the standard mark-up 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.

A simple HTML Document

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

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

- The <p> element defines a paragraph

2. 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>
```

3. 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

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootst
rap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></scr
ipt>
</head>
<body>
<div class="container">
  <h1>My First Bootstrap Page</h1>
  <p>This is some text.</p>
</div>
</body>
</html>
```

Bootstrap CDN

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.
min.css">

<!-- jQuery library -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
```

```
<!-- Latest compiled JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
```

Conclusion:

Hence we have successfully created static webpage using HTML, CSS and Bootstrap.

Assignment 1(b):

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 theresults, 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 independentof web server software.

A user can continue to use the application while the client program requests information from

the server in the background. Intuitive and natural user interaction. Clicking is not required, mouse movement is a sufficient event trigger. Data-driven as opposed to page-driven.

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.

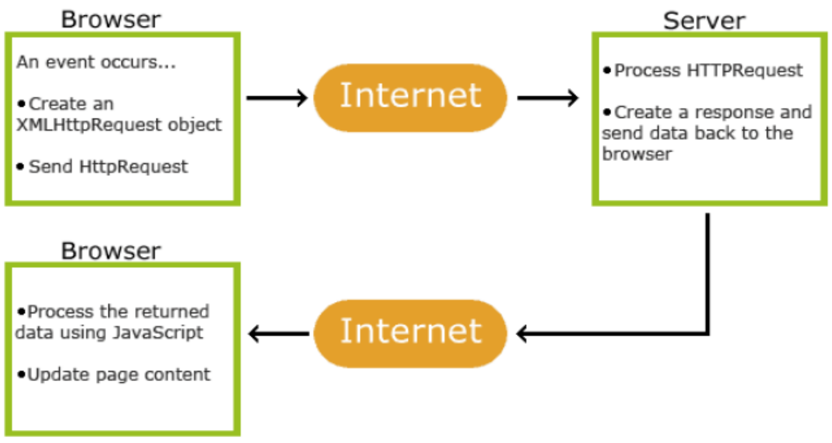
– Represents the structure of XML and HTML documents.

• CSS

– Allows for a clear separation of the presentation style from the contentand may be changed programmatically by JavaScript

• XMLHttpRequest

– JavaScript object that performs asynchronous interaction with the server.


Browser
An event occurs...
•Create an XMLHttpRequest object
•Send HttpRequest
Internet
Server
•Process HTTPRequest
•Create a response and send data back to the browser
Browser
•Process the returned data using JavaScript
•Update page content
Internet

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. • 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"

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 should 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.

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(){
varreq=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:

Hence we have successfully implemented AJAX POST method.

Assignment 2 (a)

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.

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	gitconfig --global user.name "w3schools-test" gitconfig --global user.email"test@w3schools.com"	Configure Git Change the user name and e-mail address to your own
4	mkdirmyproject cdmyproject	Creating Git Folder
5	gitinit	Initialize Git Initialized empty Git repository in /Users/user/myproject/.git/
6	git status	To check the status
7	gitadd index.html	Add file to staging environment
8	gitadd --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 *command* -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 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:

Hence we have studied Github commands successfully.

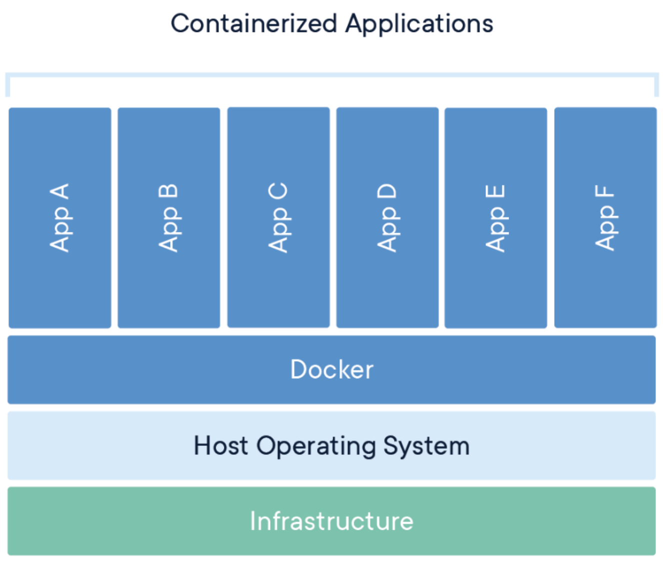
Assignment 2 (b):

Create Docker Container Environment (NVIDEIA Docker or any other).

Theory:

What is Docker?

Docker is the leading container platform which provides both hardware and software encapsulation by allowing multiple containers to run on the same system at the same time each with their own set of resources (CPU, memory, etc) and their own dedicated set of dependencies (library version, environment variables, etc.).

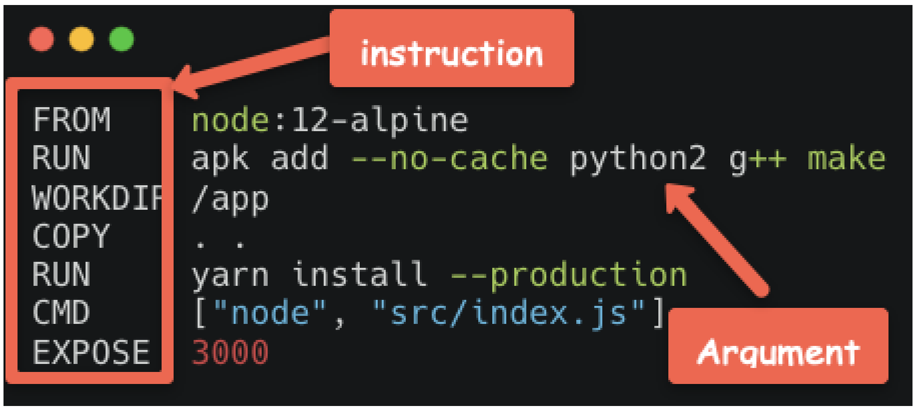


Requirements: -

1. Minimum Windows 10 (Home and All Other Editions)
2. Hyper-V (In-Built and can be Enabled)
3. Only 64bit Processor Architecture
4. Virtualization Enablement from Bios Level

The very basic building block of a Docker image is a Dockerfile. A Dockerfile is a simple text file with instructions and arguments. Docker can build images automatically by reading the instructions given in a Dockerfile.

In a Dockerfile Everything on left is **INSTRUCTION**, and on right is an **ARGUMENT** to those instructions. Remember that the file name is “Dockerfile” without any extension.



```
FROM     node:12-alpine
RUN      apk add --no-cache python2 g++ make
WORKDIR  /app
COPY     . .
RUN      yarn install --production
CMD      ["node", "src/index.js"]
EXPOSE   3000
```

The following table contains the important **Dockerfile** instructions and their explanation.

Dockerfile Instruction	**Explanation**
FROM	To specify the base image which can be pulled from a container registry(Docker hub, GCR, Quay, ECR, etc)
RUN	Executes commands during the image build process.
ENV	Sets environment variables inside the image. It will be available during build time as well as in a running container. If you want to set only build-time variables, use ARG instruction.
COPY	Copies local files and directories to the image
EXPOSE	Specifies the port to be exposed for the Docker container.
ADD	It is a more feature-rich version of the COPY instruction. It also allows copying from the URL that is the source and **tar file auto-extraction** into the image. However, usage of COPY command is recommended over ADD. If you want to download remote files, use curl or get using RUN.
WORKDIR	Sets the current working directory. You can reuse this instruction in a Dockerfile to set a different working directory. If you set WORKDIR, instructions like RUN, CMD, ADD, COPY, or ENTRYPOINT gets executed in that directory.
VOLUME	It is used to create or mount the volume to the Docker container
USER	Sets the user name and UID when running the container. You can use this instruction to set a non-root user of the container.

Dockerfile Instruction	**Explanation**
LABEL	It is used to specify metadata information of Docker image
ARG	Is used to set build-time variables with key and value. the ARG variables will not be available when the container is running. If you want to persist a variable on a running container, use ENV.
CMD	It is used to execute a command in a running container. There can be only one CMD, if multiple CMD there then it only applies to the last one. It **can be overridden** from the Docker CLI.
ENTRYPOINT	Specifies the commands that will execute when the Docker container starts. If you don't specify any ENTRYPOINT, it defaults to /bin/sh -c. You can also **override ENTRYPOINT** using the --entrypoint flag using CLI. Please refer CMD vs ENTRYPOINT for more information.

Steps to use Docker: -

1. Download Docker Desktop and install
2. Install Docker extension in Visual Studio
3. Create Docker environment in docker desktop
4. Create Directory in VS using terminal, using mkdir directory-name command (**mkdir xyz**) and move to directory using cd directory-name(**cd xyz**)
5. Create any type of file inside xyz directory (Example **a.js**)
6. Run file on terminal using node command (**node a.js**) just to check there are no problems with file
7. Create file inside xyz directory named as **Dockerfile** (D is always capital)
8. In Dockerfile, write Instructions and arguments for those instructions.

    FROM node:alpine (node is base image for nodejs and alpine is linux distribution)

    COPY . /cd

    CMD node /cd/a.js

9. Write command on terminal, **docker build –t xyz .**

This command will create image for your folder. You can check image using **docker image ls** or you can check in image part of docker desktop

10. Run docker using **docker run xyz** command.

Conclusion:

Hence we have successfully created Docker Container and environment.

Assignment 2 (c):

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

Theory:

What is Angular?

Angular is a typescript based free and open source web application framework developed by Google. Angular 10+ is a JavaScript framework which is used to create single page application. The Angular applications are created with the help of HTML and Typescript.

Angular CLI- command line interface tool that you use to initialize, develop and maintain angular application directly from command prompt or terminal.

Features:

High speed web application
Dynamic development
Full stack development

Pre-requisite for angular installation:

node.js and **npm** should be installed on your machine

How to install angular?

- **Step 1:** check your machine for node.js and npm
  ```
  node –v
  npm –v
  ```
- **Step 2:** Install Angular CLI
  ```
  npm install -g @angular/cli
  ```
- **Step 3:** check version of Angular
  ```
  ng version
  ```
- **Step 4:** Create workspace folder
  ng new Proj_name
  Would you like to add Angular routing? **Y**
  Which stylesheet format would you like to use? **CSS**
- **Step 5:** change directory to the proj_name folder and issue command
  ```
  ng serve -o
  ```

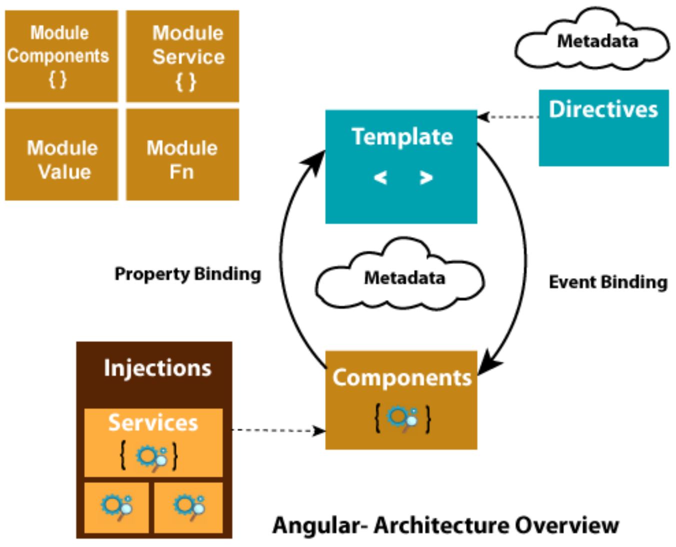
Angular Project Structure

The folder structure of Angular project is:

- **node_modules:** contains folders of packages which are installed
- **src folder:** this is the place where we need to put all our application source code

- **app folder:** When we want to create any component, service or module, we need to create it within this app folder.
- **assets folder:** you can store static assets like images, icons etc.
- **environment folder:** used to set up different environments.
- **favicon.ico:** It is the icon file that displays on the browser
- **index.html:** Starting point of our application.
- **main.ts** file
- **polyfills.ts:** used for browser-related configuration
- **angular. json file:** It contains the configuration of your angular project
- **test.ts** and **karma.config.js:** used for testing purpose
- **Package.json:** mandatory for every npm project

Angular Architecture



Components

Components and services both are simply classes with decorators that mark their types and provide metadata which guide Angular to do things.

Angular Modules

Angular 10 NgModules are different from other JavaScript modules. Every Angular 10 app has a root module known as AppModule. It provides the bootstrap mechanism that launches the application. Generally, every Angular 10 app contains many functional modules.

Angular Data Binding
Data binding defines the communication between components and its view.
Property binding: one way binding in which we can set the properties of the element to the user interface page.
Event binding: flow of data from view component.

Directives and Pipes
Directives is a technique in Angular that adds additional behavior to the elements in the Angular applications.
Pipes are used to transform the data.
The pipes are written using pipe operator which is denoted by |.

Angular Services and Dependency Injections (DI)
An Angular service is plain Typescript class having one or more methods along with @Injectable decorator.
The Services in angular are injected into the application using the dependency injection mechanism.
Dependency injection is a technique in which an object receives other objects that it depends on.

Conclusion:
Hence we have studied Angular and successfully implemented angular application.

Assignment 3 (a):

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

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

Theory:

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.
- **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

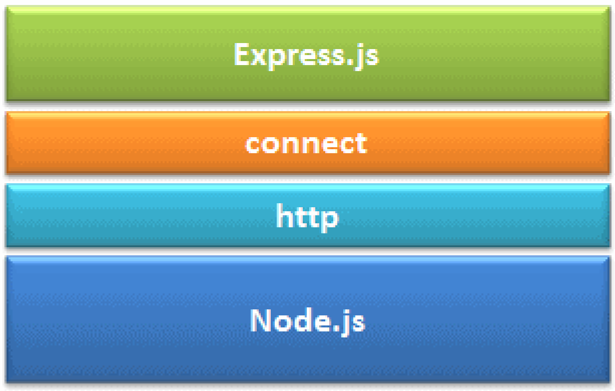
- **Programming interface isn't steady:** The Application Programming Interface (API) of Node can be challenging to work with. It changes regularly and doesn't remain stable.
- **No strong library support system:** JavaScript does not hold a strong library system. This limits the developers to implement 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.
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

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.

Conclusion:

Hence we have successfully created static website using Node JS.

Assignment 3 (b)

Create four API using Node.JS, ExpressJS and MongoDB for CURD Operations on assignment 2.C.

Theory:

REST HTTP Method APIs

- API (Application Programming Interface) are a set of functions and procedures that allow for the creation of applications that access data and features of other applications.
- REST (Representational State Transfer) is a set of rules that developers follow while creating API.
- REST uses various representation to represent a resource like text, JSON, XML but JSON is the most popular one.
- REST APIs enable you to develop all kinds of web applications having all possible CRUD (create, retrieve, update, delete) operations.
- The request consists of:
- **End Point:** An **endpoint** contains a *Uniform Resource Identifier (URI)* indicating where and how to find the resource on the Internet.
- **Method:** GET, POST, DELETE, PUT
- **Headers:** used to provide authentication and other useful information to client and server.
- **Data:** The DATA contains information which the client wants to send to the server. Preferred to send data in JSON format.

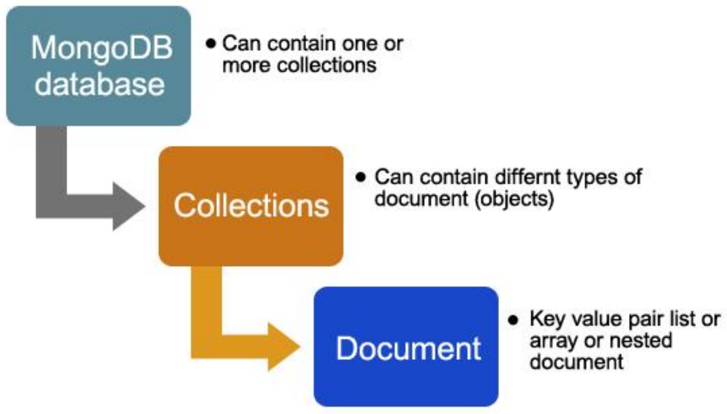
HTTP methods

- GET − This is used to provide a read only access to a resource.
- PUT − This is used to update a new resource.
- DELETE − This is used to remove a resource.
- POST − This is used to create a new resource.

What is MongoDB?

MongoDB is an open-source database management system (DBMS) that uses a document-oriented database model. MongoDB is a NoSQL Database. MongoDB stores data in JSON-like documents, which makes the database very flexible and scalable.

MongoDB is a document-oriented database model. Each MongoDB database contains collections and which in turn contains documents. Each document can be different and depends on the varying number of fields. The model of each document will be different in size and content from each other. The data model features allow you to store arrays and complex structured in a hierarchical relationship.


MongoDB database
Can contain one or more collections
Collections
Can contain differnt types of document (objects)
Document
Key value pair list or array or nested document

- **Collection:** Its group of MongoDB documents. This can be thought similar to a table in RDBMS like Oracle, MySQL. This collection doesn't enforce any structure. Hence schema-less MongoDB is so popular.
- **Document:** Document is referred to as a record in MongoDB collection.

MongoDB CURD Operation

CRUD operations create, read, update, and delete documents.

1. Create Operations

Create or insert operations add new documents to a collection. If the collection does not currently exist, insert operations will create the collection.

MongoDB provides the following methods to insert documents into a collection:

- db.collection.insertOne()
- db.collection.insertMany()

2. Read Operation

Read operations retrieve documents from a collection; i.e. query a collection for documents. MongoDB provides the following methods to read documents from a collection:

- db.collection.find()

3. Update Operation

MongoDB Update method is used to update the document from the collection.

We have used a $set operator at the time of updating the document.

Using the update method, we can update a single as well as multiple documents in one statement.

- db.collection.updateOne()
- db.collection.updateMany()
- db.collection.replaceOne()

4. Delete Operations

Delete operations remove documents from a collection. MongoDB provides the following methods to delete documents of a collection:

- db.collection.deleteOne()
- db.collection.deleteMany()

Conclusion:
Hence we have studied Node.js, ExpressJS and MongoDB and successfully created API.

Assignment 4(a):

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.

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.

Assignment 4 (b)

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

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 billon 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.

Applications of AWS services

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

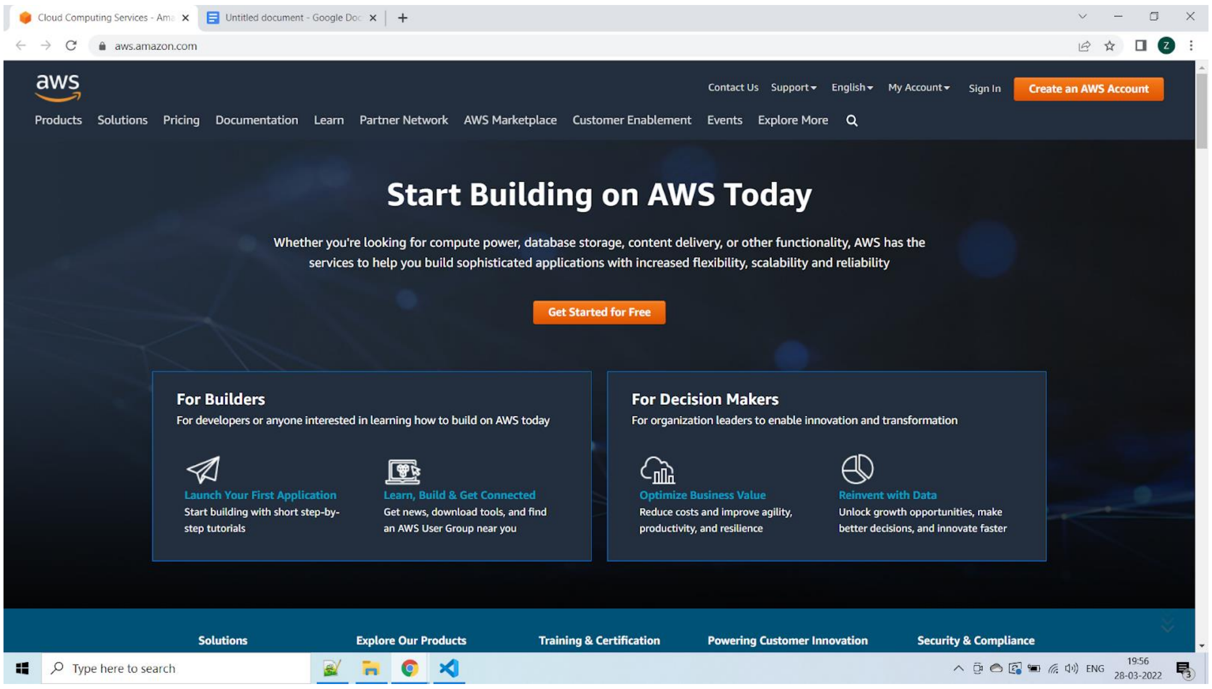
- Web site hosting
- Application hosting/SaaS hosting
- Media Sharing (Image/ Video)
- Mobile and Social Applications
- Content delivery and Media Distribution

- 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.**

<u>**Steps to follow are as follows:**</u>

Step 1 – Visiting the Signup Page
Go to https://aws.amazon.com
You should see something like below:



Start Building on AWS Today

Whether you're looking for compute power, database storage, content delivery, or other functionality, AWS has the services to help you build sophisticated applications with increased flexibility, scalability and reliability

For Builders
For developers or anyone interested in learning how to build on AWS today

Launch Your First Application
Start building with short step-by-step tutorials

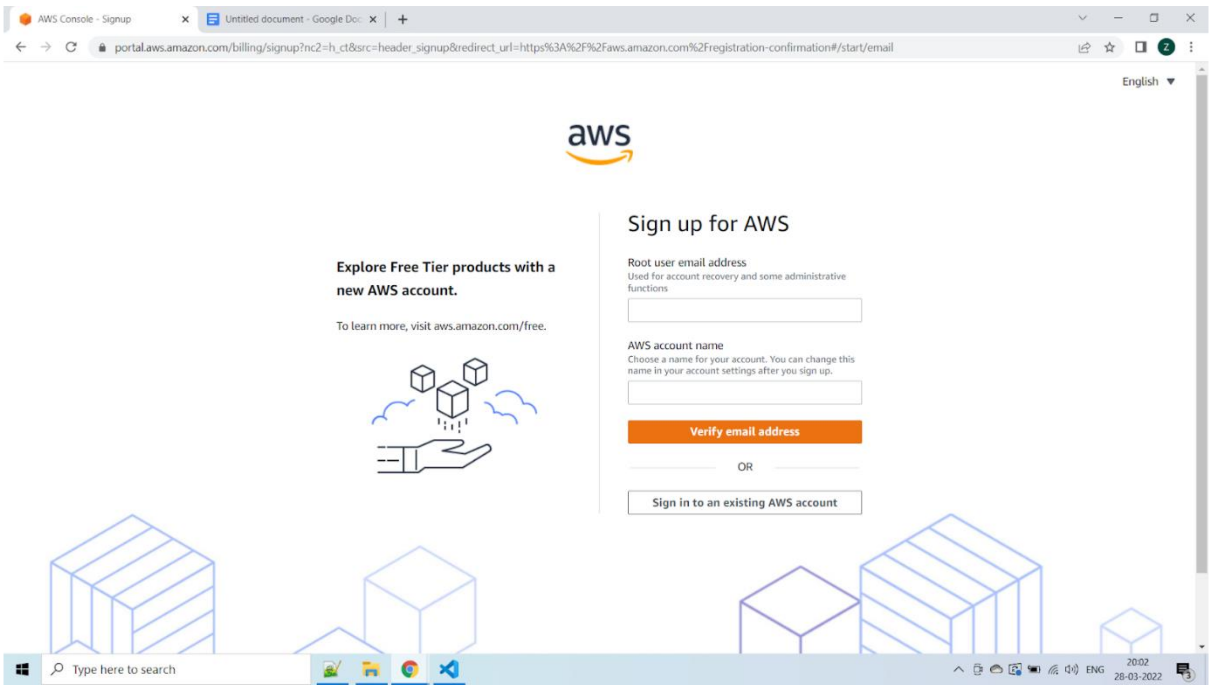
Learn, Build & Get Connected
Get news, download tools, and find an AWS User Group near you

For Decision Makers
For organization leaders to enable innovation and transformation

Optimize Business Value
Reduce costs and improve agility, productivity, and resilience

Reinvent with Data
Unlock growth opportunities, make better decisions, and innovate faster

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.


Sign up for AWS

Explore Free Tier products with a new AWS account.

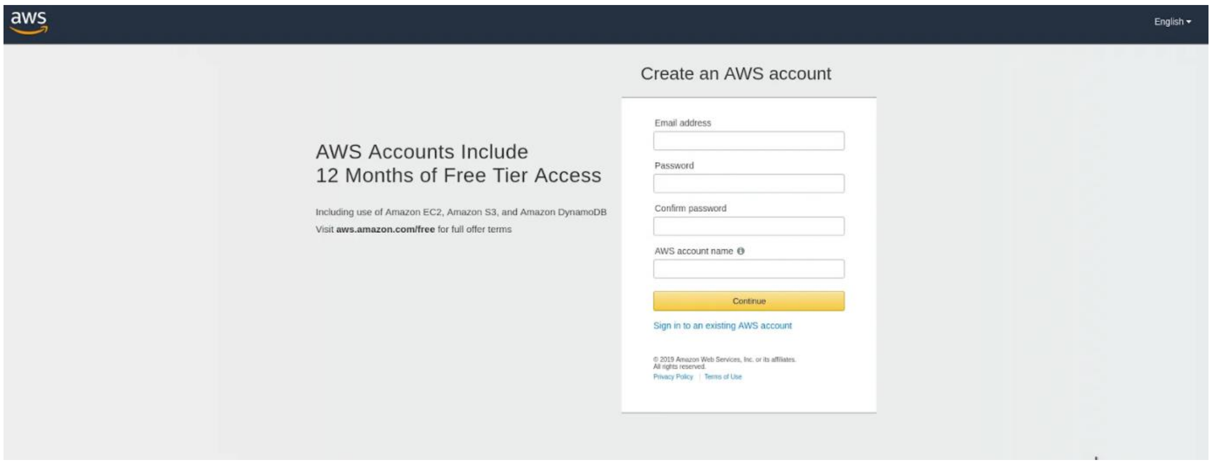
To learn more, visit aws.amazon.com/free.

Root user email address
Used for account recovery and some administrative functions

AWS account name
Choose a name for your account. You can change this name in your account settings after you sign up.

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

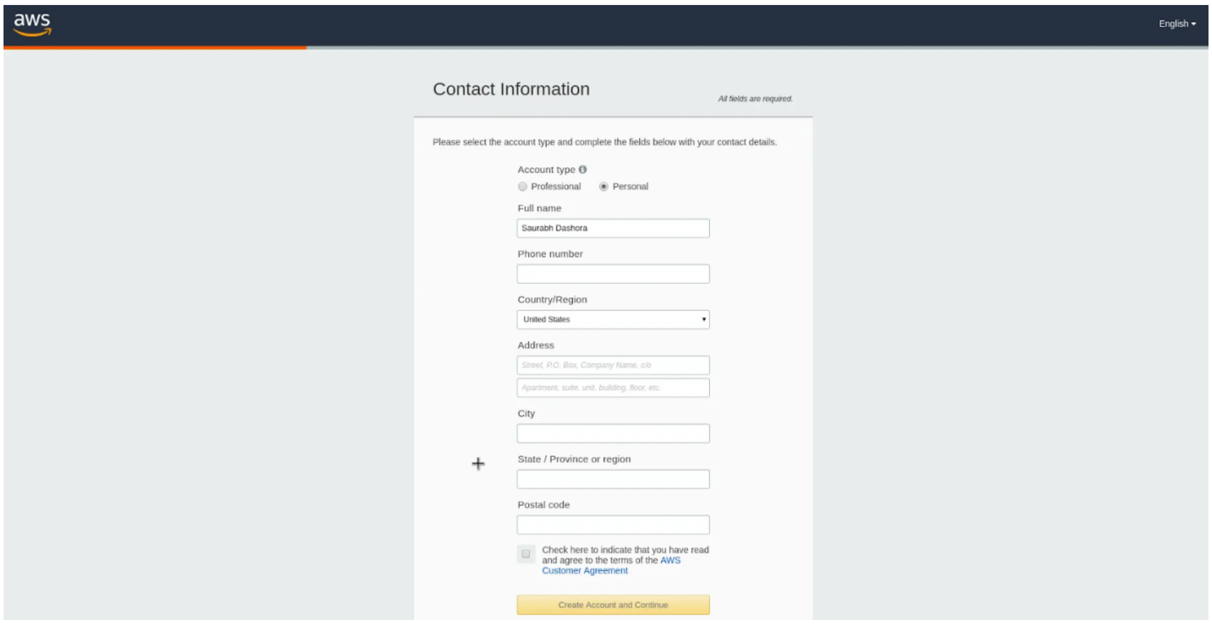
Including use of Amazon EC2, Amazon S3, and Amazon DynamoDB
Visit **aws.amazon.com/free** for full offer terms

Email address
Password
Confirm password
AWS account name

© 2019 Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy | Terms of Use

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.



Contact Information

All fields are required.

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

Account type
Professional [✓ Personal]

Full name
Saurabh Dashora

Phone number

Country/Region
United States

Address
Street, P.O. Box, Company Name, c/o
Apartment, suite, unit, building, floor, etc.

City

State / Province or region

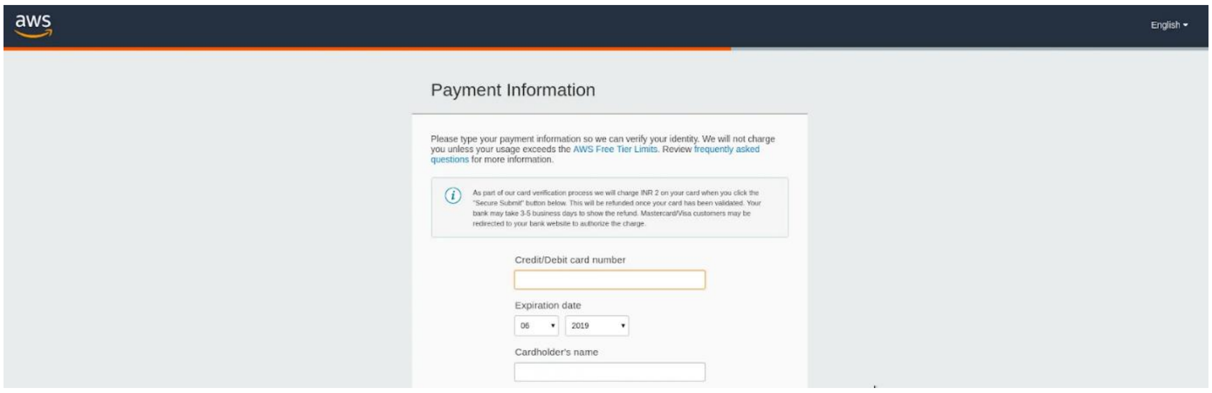
Postal code

Check here to indicate that you have read and agree to the terms of the AWS Customer Agreement

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 AWS Free Tier Limits. Review frequently asked questions for more information.

As part of our card verification process we will charge INR 2 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 show the refund. Mastercard/Visa customers may be redirected to your bank website to authorize the charge.

Credit/Debit card number

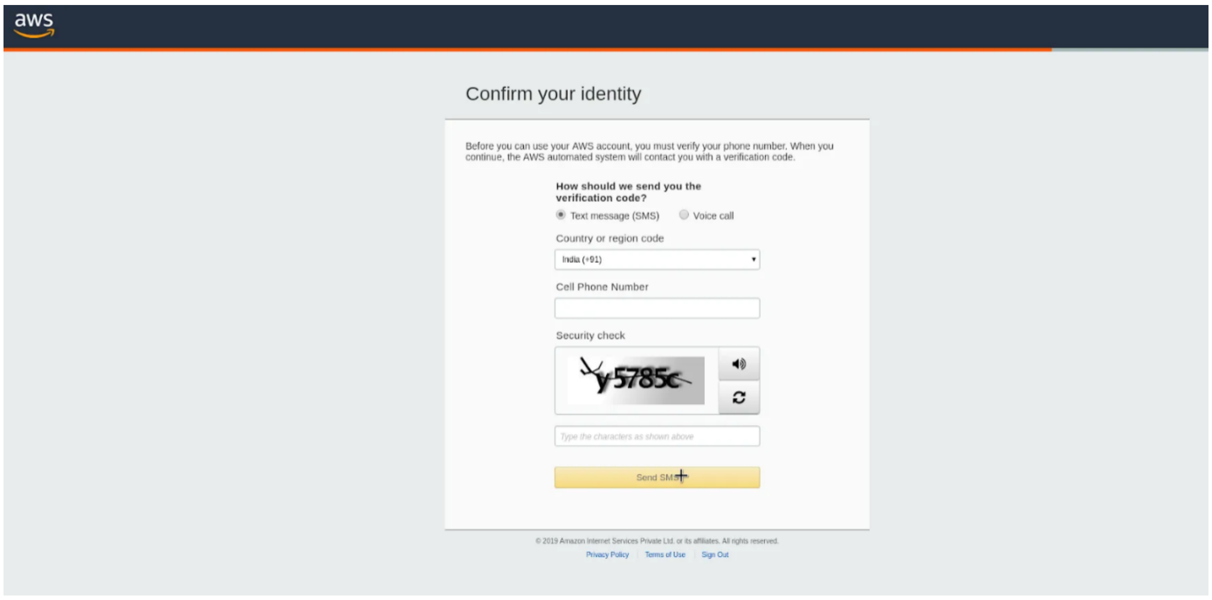
Expiration date
06 2019

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:



Confirm your identity

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

How should we send you the verification code?

[✓ Text message (SMS)] Voice call

Country or region code: India (+91)

Cell Phone Number

Security check

y5785c

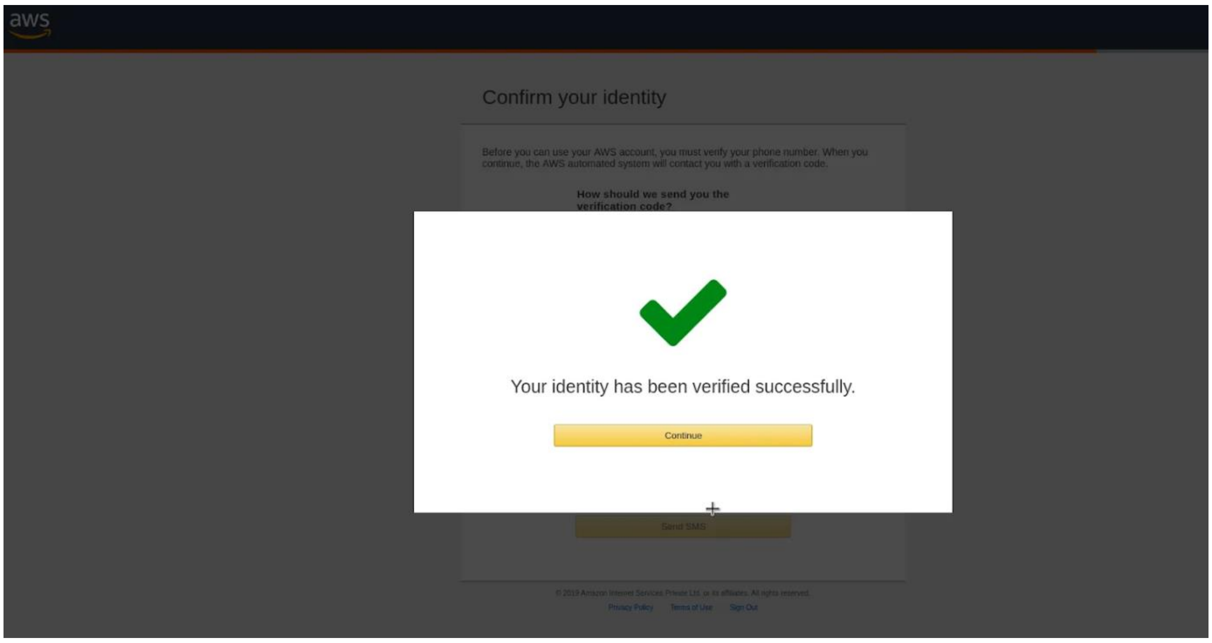
Type the characters as shown above

Send SMS

© 2019 Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

Privacy Policy | Terms of Use | Sign Out

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



Confirm your identity

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

How should we send you the verification code?

Your identity has been verified successfully.

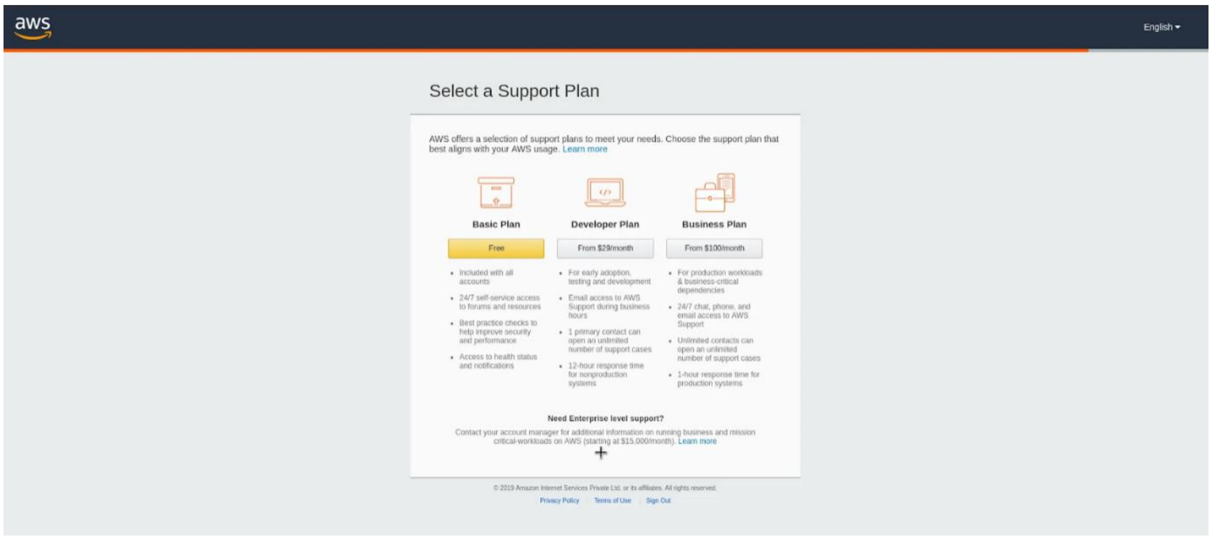
Continue

Send SMS

© 2019 Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

Privacy Policy | Terms of Use | Sign Out

Step 5 – Selecting a Support Plan


Select a Support Plan

AWS offers a selection of support plans to meet your needs. Choose the support plan that best aligns with your AWS usage. Learn more

Basic Plan	Developer Plan	Business Plan
Free	From $29/month	From $100/month
• Included with all accounts • 24/7 self-service access to forums and resources • Best practice checks to help improve security and performance • Access to health status and notifications	• For early adoption, testing and development • Email access to AWS Support during business hours • 1 primary contact can open an unlimited number of support cases • 12-hour response time for nonproduction systems	• For production workloads & business-critical dependencies • 24/7 chat, phone, and email access to AWS Support • Unlimited contacts can open an unlimited number of support cases • 1-hour response time for production systems

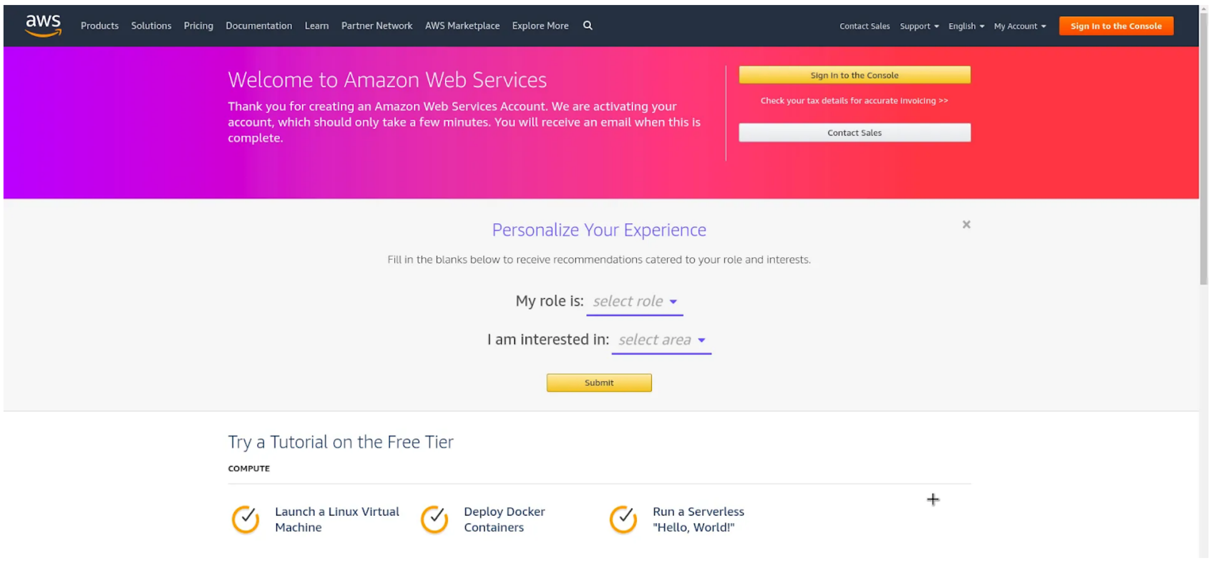
Need Enterprise level support?

Contact your account manager for additional information on running business and mission critical-workloads on AWS (starting at $15,000/month). Learn more

© 2019 Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

Privacy Policy | Terms of Use | Sign Out

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**.



Welcome to Amazon Web Services

Thank you for creating an Amazon Web Services Account. We are activating your account, which should only take a few minutes. You will receive an email when this is complete.

Personalize Your Experience

Fill in the blanks below to receive recommendations catered to your role and interests.

My role is: select role

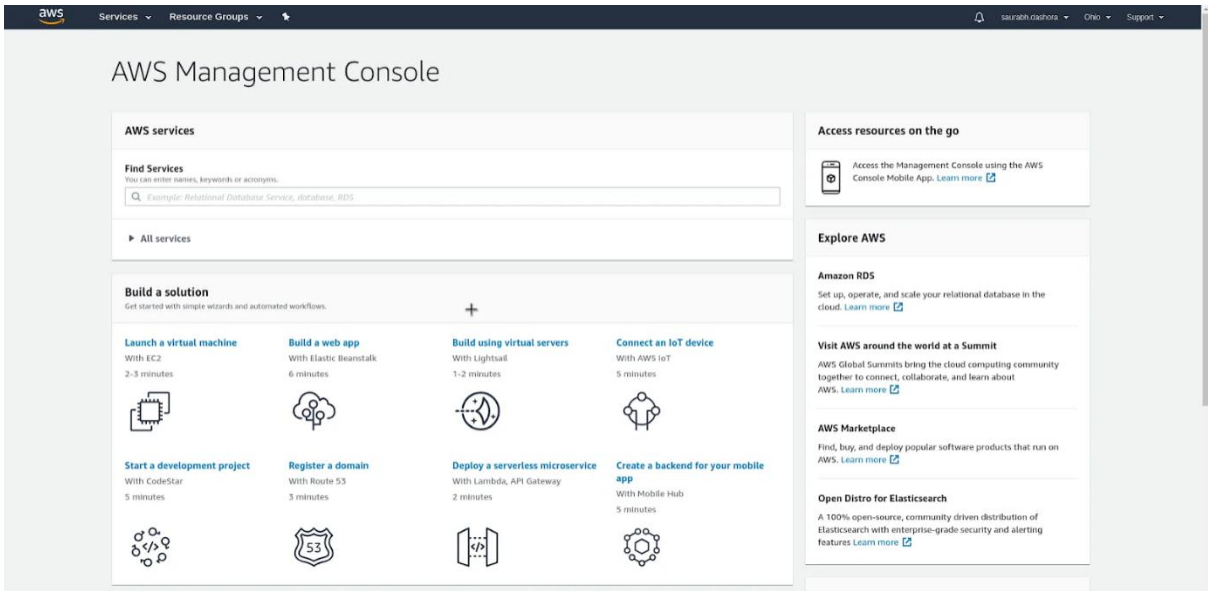
I am interested in: select area

Try a Tutorial on the Free Tier

COMPUTE

Launch a Linux Virtual Machine
Deploy Docker Containers
Run a Serverless "Hello, World!"

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



AWS Management Console

AWS services

Find Services
You can enter names, keywords or acronyms.
Example: Relational Database Service, database, RDS

All services

Build a solution
Get started with simple wizards and automated workflows.

Launch a virtual machine
With EC2
2-3 minutes

Build a web app
With Elastic Beanstalk
6 minutes

Build using virtual servers
With Lightsail
1-2 minutes

Connect an IoT device
With AWS IoT
5 minutes

Start a development project
With CodeStar
5 minutes

Register a domain
With Route 53
3 minutes

Deploy a serverless microservice
With Lambda, API Gateway
2 minutes

Create a backend for your mobile app
With Mobile Hub
5 minutes

Access resources on the go

Access the Management Console using the AWS Console Mobile App. Learn more

Explore AWS

Amazon RDS
Set up, operate, and scale your relational database in the cloud. Learn more

Visit AWS around the world at a Summit
AWS Global Summits bring the cloud computing community together to connect, collaborate, and learn about AWS. Learn more

AWS Marketplace
Find, buy, and deploy popular software products that run on AWS. Learn more

Open Distro for Elasticsearch
A 100% open-source, community driven distribution of Elasticsearch with enterprise-grade security and alerting features Learn more

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:

Hence we have successfully deployed web on AWS Elastic Beanstalk.