314458: Laboratory Practice –II
(Web Application Development)

Third Year – Information Technology

(2019 Course)

Academic Year 2022-23 Semester – II

Teaching Scheme:	Credit Scheme:	Examination Scheme:	
Practical (PR): 04 hrs/week	02 Credit	PR: 25 Marks	TW: 50Marks

Sinhgad Institutes

LABORATORY MANUAL V 3.0

DEPARTMENT OF INFORMATION TECHNOLOGY

Sinhgad 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 endeavor 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 the Information Technology course after passing out will:

Sr. No.	Description
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 analyzing, 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 a commitment to 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 professionals and will be able to address challenges in the field of IT at the global level.

PROGRAM OUTCOMES

The students in the Information Technology course are expected to know and be able to:

Sr. No.	PO's	Description
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 solution with the help of conducting experiments, analyzing the problem and interpreting the data.
PO3	Design/Development of Solutions	An ability to design, implement, and evaluate software or a software/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 schematic solutions to complex engineering /Technology problems.
PO5	Modern Tool Usage	An ability to use the techniques, skills, and modern engineering technology tools, and standard processes necessary for practice as an 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 solutions 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 social issues and responsibilities.
PO9	Individual and Team Work	An ability to function effectively as an individual or a sate am 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

A graduate of the Information Technology Program will demonstrate-

Sr. No.	Description
PSO1	An ability to apply the theoretical concepts and practical knowledge of Information Technology in analysis, design, development and management of information 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.

DOCUMENT CONTROL

Reference Code	SCOE-IT / Lab Manual Procedures
Version No	3.0
Compliance Status	Complete
Date of Compliance	01-02-2023
Security Classification	Department Specific
Document Status	Definitive
Review Period	Yearly

	Author
Signature	
Name	**Mr. A. S. Shinde**
Designation	Assistant Professor

Document History

Revision No.	Revision Date	Reason For Change
1	01-07-2021	Update
2	01-07-2022	Update
3	01-02-2023	Updating the theory content of a few assignments

Summary of Changes to Laboratory Practice-II (Web Application Development)

Sr. No	Changes	Change type
1	Assignment 2	Theory content added
2		

LABORATORY CODE

Sr. No.	Laboratory Code
1	Students should report to the concerned laboratory as per the time table.
2	Keep your bags in rack.
3	While entering in lab remove your shoes and keep it in shoe stand.
4	Turn computer monitors off when asked by your teacher
5	Do not go on banned websites.
6	No food or drinks near the keyboard.
7	Only use your assigned computer and workstation.
8	Do not change the settings on the computer
9	Ask permission to download.
10	Ask permission to print documents
11	Save your work often.
12	If you are the last class of the day, please POWER DOWN all computers and monitors.

SYLLABUS

Savitribai Phule Pune University, Pune

Third Year Information Technology (2019 Course)

314458: Laboratory Practice-II (Web Application Development)

Teaching Scheme:	Credit Scheme:	Examination Scheme:	
Practical (PR) : 04 hrs/week	02 Credit	PR :25 Marks	TW : 50Marks

Prerequisite Courses:

- Programming languages C++, Java

Course Objectives:

1. To understand basic concepts of web programming and scripting languages.
2. To learn Version Control Environment.
3. To learn front end technologies and back end technologies.
4. To understand mobile web development.
5. To comprehend web application deployment.

Course Outcomes:

On completion of the course, students will be able to

CO1: Develop Static and Dynamic responsive website using technologies HTML, CSS, Bootstrapand AJAX.

CO2: Create Version Control Environment.

CO3: Develop an application using front end and backend technologies.

CO4: Develop mobile website using JQuery Mobile.

CO5: Deploy web application on cloud using AWS.

LIST OF ASSIGNMENTS

Group A-(WAD)

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.

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

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.

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 Books:

1. Kogent Learning Solutions Inc, Web Technologies: HTML, JAVASCRIPT, PHP, JAVA, JSP, XML and AJAX, Blackbook, Dreamtech Press, Second Edition, ISBN: 9788177228496.
2. Raymond Camden, Andy Matthews, jQuery Mobile Web Development Essentials, Packt Publishing, Second Edition, 9781782167891.
3. Steven M. Schafer, “HTML, XHTML and CSS”, Wiley India Edition, Fourth Edition, 978- 81-265- 1635-3
4. Dr.HirenJoshi, Web Technology and Application Development, DreamTech, First,ISBN:978-93- 5004-088-1
5. Steven M. Schafer, “HTML, XHTML and CSS”, Wiley India Edition, Fourth Edition, 978- 81-265- 1635-3
6. Ivan Bayross,”Web Enabled Commercial Application Development Using HTML, JavaScript, DHTML and PHP,BPB Publications,4th Edition,ISBN:978-8183330084.
7. Brain Fling, Mobile Design and Development, O'REILLY, First Edition, ISBN: 13:978-81- 8404-817-
8. Adam Bretz & Colin J Ihrig, Full Stack Javascript Development with MEAN, SPD, First Edition, ISBN:978-0992461256.

- Books / E- Learning References

1. https://www.meanacademy.in/web-technologies
2. https://www.meanacademy.in/angular
3. https://www.meanacademy.in/mongodb
4. https://www.meanacademy.in/nodejs
5. https://www.meanacademy.in/aws

ACADEMIC YEAR 2022-23, SEMESTER-II

Name of Student	:		**PRN No. :**	
Student Roll No.	:		**Class:**	Third Year
Subject	:	Laboratory Practice-II (Web Application Development)	**Batch:**	

I N D E X

Ass. No.	Title of Assignment	Pg. No.	Given Date	Submission Date	Re-Mark	Sign
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.		23	23		
	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.		23	23		
2	**a.** Create version control account on GitHub and using Git commands to create repository and push your code to GitHub.		23	23		
	b. Create Docker Container Environment (NVIDEIA Docker or any other.		23	23		

	c. Create an Angular application which will do following actions: Register User, Login User, Show User Data on Profile Component				23			23		
3	**a.** Create a Node.JS Application which serves a static website.				23			23		
	b. Create four API using Node.JS, ExpressJS and MongoDB for CURD Operations on assignment 2.C.				23			23		
4	**a.** Create a simple Mobile Website using jQuery Mobile.				23			23		
	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				23			23		

MOCK PRACTICAL & ASSIGNMENT EVALUATION

A. MOCK PRACTICAL EVALUATION

Sr. No	Performance Indicator Tools		Mock – I Marks	Mock - II Marks	Average Marks (out of 25)
	Date of Mock				
1	Attendance	(05)			
2	Logic	(05)			
3	Code Reliability	(05)			
4	Execution	(10)			
		Total Marks(25):			

B. ASSIGNMENT EVALUATION

Assignment No.	Regularity (06)	Presentation (06)	Understanding (08)	Timely Submission (05)	Total Marks (25)
1					
2					
3					
4					
				Average Marks(Out of 25):	
				Average Marks (A +B):	

Signature of Subject In charge:	

PRACTICAL ASSESSMENT/ATTAINMENT (In-direct)

Sr. No	Performance Indicator Tools	Marks	Remark
1	Fundamental Knowledge		
2	Analyze and Interpretation of Data		
3	Constraint Design or Procedure		
4	Identify/Formulate/Evaluate/Solution of Problem		
5	Lab. Ethics and Communication		
6	Oral/Written/Graphical Form		
7	Result and Measurement Error		
8	Analysis and Theory Application		
9	Accurate Documentation		
10	Regular Attendance		
	Average Marks (out of 25):		

(Remarks: <=10: Poor, 11-15: Satisfactory, 16-20: Intermediate, 20-25: Excellent)

ASSIGNMENT NO. :01	**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.
GIVEN DATE:	/ /2023
SUBMISSION DATE:	/ /2023
SIGN. OF STUDENT:	

REGULARITY		PRESENTATION		UNDERSTANDING		TIMELY SUBMISSION		TOTAL MARKS	
	06		06		08		05		25

SIGN. OF FACULTY:	
REMARKS:	

ASSIGNMENT NO. : 01(A)

AIM:

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

PRE-REQUISITE:

1. Knowledge of HTML5.
2. Knowledge of JavaScript.
3. Knowledge of CSS.

OBJECTIVE:

1. To study basic HTML and CSS commands.
2. To install Microsoft Visual Code.
3. Create a responsive web application.

THEORY:

What is HTML?

HTML (Hyper Text Markup Language):-

Hyper Text Markup Language (HTML) is the predominant markup language for web pages.HTML elements are the basic building-blocks of web pages. HTML is written in the form of HTML elements consisting of tags, enclosed in angle brackets (like <html>), within the web page content. HTML tags most commonly come in pairs like <h1> and </h1>, although some tags, known as empty elements, are unpaired, for example <img>. The first tag in a pair is the start tag, the second tag is the end tag (they are also called opening tags and closing tags). In between these tags web designers can add text, tags, comments and other types of text-based content.

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

Origins:

In 1980, physicist Tim Berners-Lee, who was a contractor at CERN, proposed and prototyped ENQUIRE, a system for CERN researchers to use

and share documents. In 1989, Berners-Lee wrote a memo proposing an Internet-based hypertext system. Berners-Lee specified HTML and wrote the browser and server software in the last part of 1990.

The first publicly available description of HTML was a document called "HTML Tags", first mentioned on the Internet by Berners-Lee in late 1991. It describes 20 elements comprising the initial, relatively simple design of HTML. Except for the hyperlink tag. Thirteen of these elements still exist in HTML 4.

Structure of an HTML document:

```
<!DOCTYPE  html>          ← Tells the document type
  <html>                  ← The Root Element
    <head>                ← Contains the header information
     <title> Title of the Page</title>   ← Defines Title of the Page
    </head>
    <body>                ← Holds the content of the Page
          Tags related to layout and formatting
    </body>
  </html>
```

Basic HTML Tags:

Sr. No.	Tag Name	Description
1.	<html> </html>	Creates an HTML document
2.	<head> </head>	Sets off the title and other information that isn't displayed on the web page itself.
3	<body> </body>	Sets off Visible portion of the document
4	<title> </title>	Puts the name of document in the title bar.
5.	<cite> </cite>	Creates a citation, usually italic.

HTML File Paths:

Sr. No.	Tag Name	Description
1.	< img src="picture.jpg">	The "picture.jpg" file is located in the same folder as the current page
2.	<img src="images/picture.jpg">	The "picture.jpg" file is located in the images folder in the current folder.

3	<img src="/images/picture.jpg">	The "picture.jpg" file is located in the images folder at the root of the current web.

Sample HTML Code:

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

What is CSS?

CSS stands for Cascading Style Sheets. CSS is the language we use to style a Web page. 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 style sheets are stored in CSS files. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL. CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.

Sample CSS Code:

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            body {
                background-color: lightblue;
              }
            h1 {
                color: white;
                text-align: center;
              }
            p {
                font-family: verdana;
                font-size: 20px;
              }
        </style>
```

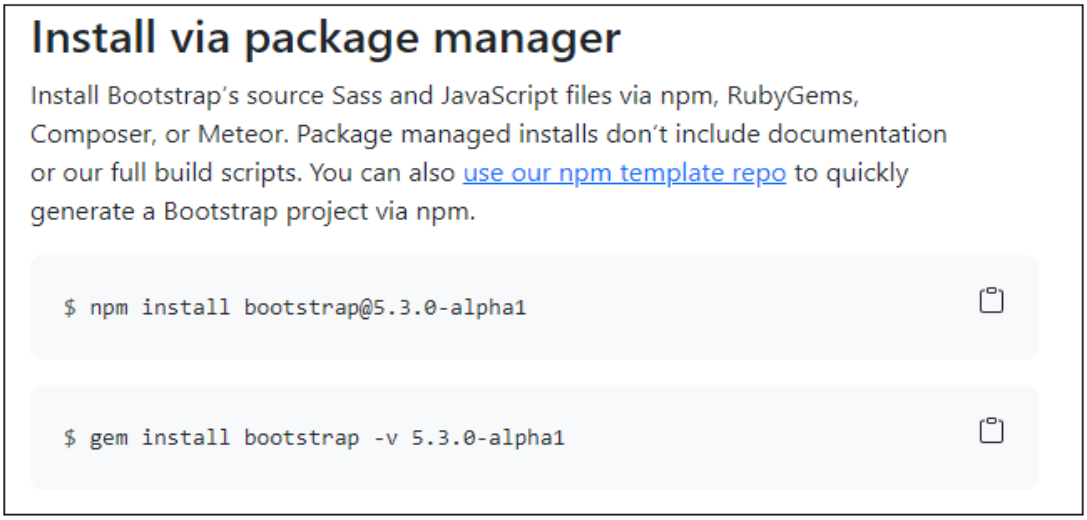
```
    </head>
    <body>
        <h1>My First CSS Example</h1>
        <p>This is a paragraph.</p>
    </body>
</html>
```

What is Bootstrap?

Bootstrap is the most popular CSS Framework for developing responsive and mobile-first websites. Bootstrap makes responsive web design a reality. It makes it possible for a web page or app to detect the visitor's screen size and orientation and automatically adapt the display accordingly.

How to Use Bootstrap in your code:

Step 1: Open Goggle Chrome and search Bootstrap.
Step 2: Open link https://getbootstrap.com/
Step 3: Install via Package Manager You can also use our npm template repo to quickly generate a Bootstrap project via npm.



Step 4: When you only need to include Bootstrap's compiled CSS or JS, you can use jsDelivr. See it in action with our simple quick start, or browse the examples to jumpstart your next project. You can also choose to include Popper and our JS separately.

CONCLUSION:

OUTPUT:

(Attach Screenshots of your output in sequence)

ASSIGNMENT NO. : 01(B)

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.

PRE-REQUISITE:

1. Knowledge of HTML5.
2. Knowledge of JavaScript.
3. Knowledge of CSS.
4. Knowledge of AJAX.

OBJECTIVE:

1. To study basics of JavaScript.
2. To install extensions required for JavaScript in Microsoft Visual Studio Code.
3. To study the GET and POST Method.

THEORY:

What is JavaScript?

JavaScript is a scripting or programming language that allows you to implement complex features on web pages — every time a web page does more than just sit there and display static information for you to look at — displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, etc. — you can bet that JavaScript is probably involved. It is the third layer of the layer cake of standard web technologies, two of which (HTML and CSS) we have covered in much more detail in other parts of the Learning Area. JavaScript is a scripting language that enables you to create dynamically updating content, control multimedia, animate images, and pretty much everything else.

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as LiveScript, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name LiveScript. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

Sr. No	Description	
1	Designed by	Brendan Eich of Netscape initially; others have also contributed to the ECMAScript standard
2	First appeared	December 4, 1995; 27 years ago
3	Stable release	ECMAScript 2021 / June 2021
4	Paradigm	Multi-paradigm: event-driven, functional, imperative, procedural, object-oriented programming.

Client-Side JavaScript:

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser. It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field. The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.

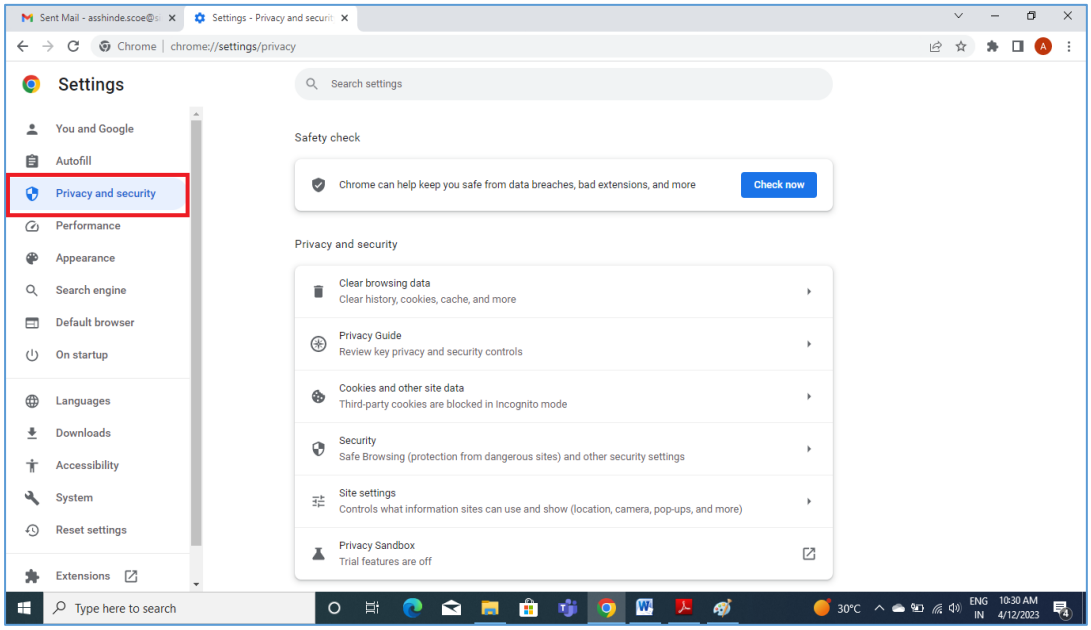
JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

JavaScript Development Tools:

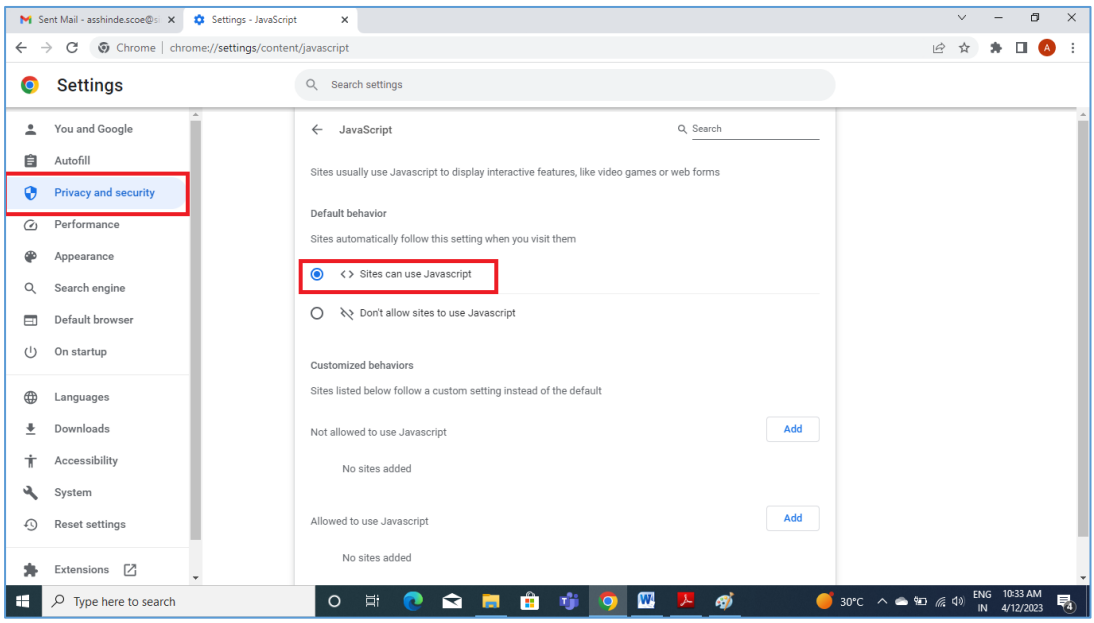
1. Microsoft FrontPage: Microsoft has developed a popular HTML editor called FrontPage. FrontPage also provides web developers with a number of JavaScript tools to assist in the creation of interactive websites.
2. Macromedia Dreamweaver MX : Macromedia Dreamweaver MX is a very popular HTML and JavaScript editor in the professional web development crowd. It provides several handy prebuilt JavaScript components, integrates well with databases, and conforms to new standards such as XHTML and XML.
3. Macromedia HomeSite 5**:** HomeSite 5 is a well-liked HTML and JavaScript editor from Macromedia that can be used to manage personal websites effectively.

How activate JavaScript in your browser:

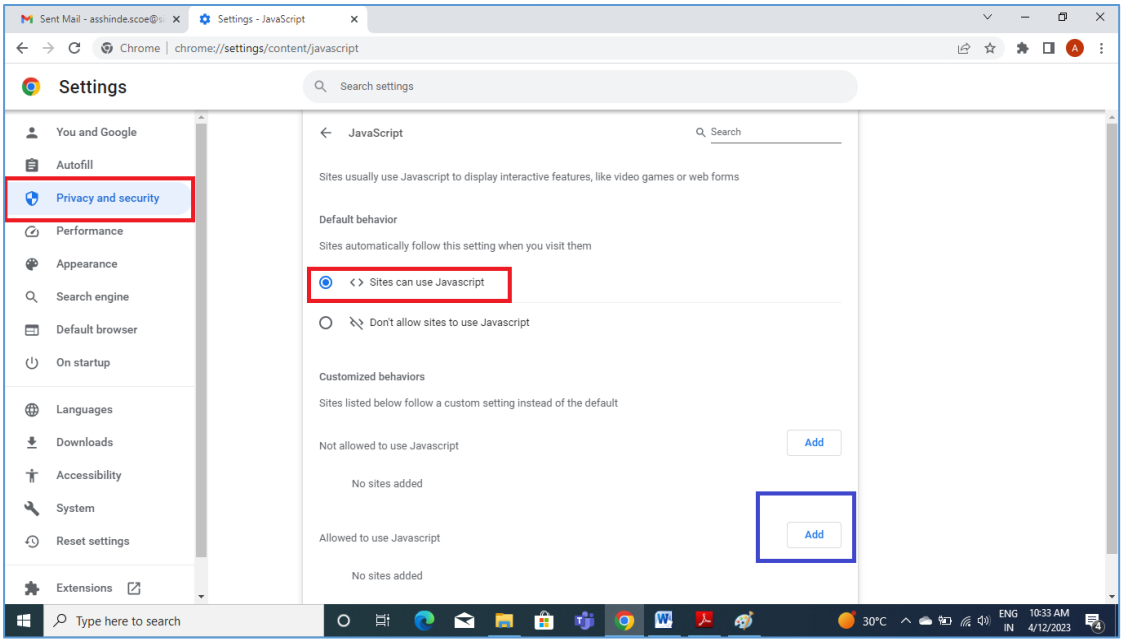
1. Open Chrome on your computer.
2. Click Settings.
3. Click Privacy and Security.



4. Click Site settings.
5. Click JavaScript.



6. Select Sites can use JavaScript.



Sample JavaScript Code:

```
<html>
   <body>
      <script language = "javascript" type = "text/javascript">
      <!--
         document.write("Hello World!")
      //-->
      </script>
   </body>
</html>
```

What is AJAX?

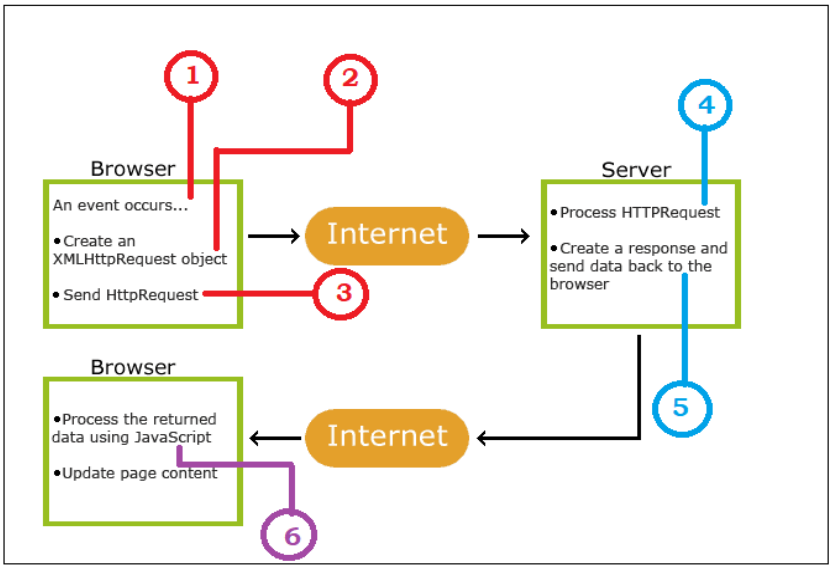
AJAX stands for Asynchronous JavaScript and XML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script.

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

How to use AJAX in JavaScript file?

1. An event occurs in a web page (the page is loaded, a button is clicked)
2. An XMLHttpRequest object is created by JavaScript.
3. The XMLHttpRequest object sends a request to a web server.
4. The server processes the request.
5. The server sends a response back to the web page.
6. The response is read by JavaScript.



Sample AJAX Code:

```
<!DOCTYPE html>
<html>
 <body>
  <div id="demo">
   <h2>Let AJAX change this text</h2>
   <button type="button" onclick="loadDoc()">Change Content</button>
  </div>
 </body>
</html>
```

What is AJAX POST method:

Sends an asynchronous http POST request to load data from the server. Its general form is: jQuery. post(url [, data] [, success] [, data Type]) url : is the only mandatory parameter. When to use AJAX POST?

Use POST if your call is going to write any data at all to the server. In fact, you should not only use this criterion for selecting between GET and POST for your Ajax calls but also for when selecting which should be used for processing forms on your web page. A POST method is used to send data to a server via API. It also creates subordinate resources, such as a file in a directory. An AJAX request is a request made by an AJAX application. Typically, it is an HTTP request made by (browser-resident) JavaScript that uses XML to encode the request data and/or response data.

How to send a POST request in AJAX:

1. Example: jQuery Ajax Request. $.ajax('/jquery/getdata', // request url { success: function (data, status, xhr) {// success callback function $('p').append(data); } });
2. Example: Get JSON Data.
3. Example: ajax() Method.
4. Example: Send POST Request.

CONCLUSION:

_____

_____

_____

_____

_____

OUTPUT:

(Attach Screenshots of your output in sequence)

<table>
<tr><td colspan="3">ASSIGNMENT NO. :02</td><td colspan="6">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</td></tr>
<tr><td colspan="3">GIVEN DATE:</td><td colspan="6">/ /2023</td></tr>
<tr><td colspan="3">SUBMISSION DATE:</td><td colspan="6">/ /2023</td></tr>
<tr><td colspan="3">SIGN. OF STUDENT:</td><td colspan="6"></td></tr>
<tr><td colspan="2">REGULARITY</td><td colspan="2">PRESENTATION</td><td colspan="2">UNDERSTANDING</td><td colspan="2">TIMELY SUBMISSION</td><td colspan="2">TOTAL MARKS</td></tr>
<tr><td></td><td>06</td><td></td><td>06</td><td></td><td>08</td><td></td><td>05</td><td></td><td>25</td></tr>
<tr><td colspan="3">SIGN. OF FACULTY:</td><td colspan="6"></td></tr>
<tr><td colspan="3">REMARKS:</td><td colspan="6"></td></tr>
</table>

ASSIGNMENT NO. : 02(A)

AIM:

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

PRE-REQUISITE:

There are no prerequisites to learn GIT basics.

OBJECTIVE:

1. To study basics Github.
2. To create Github account and repository.
3. Study various Git commands

THEORY:

What is Git?

GIT, which stands for Global Information Tracker, is a powerful and widely-used version control system commonly used for software development and other collaborative projects. Git is a DevOps tool used for source code management. It is a free and open-source version control system used to handle small to very large projects efficiently. Git is used to tracking changes in the source code, enabling multiple developers to work together on non-linear development. While Git is a tool that's used to manage multiple versions of source code edits that are then transferred to files in a Git repository, Github serves as a location for uploading copies of a Git repository. In a sense, then, there's no comparison when it comes to Git vs. GitHub as far as their function.

Git is a free and open-source version control system, originally created by Linus Torvalds in 2005. Unlike older centralized version control systems such as SVN and CVS, Git is distributed: every developer has the full history of their code repository locally. This makes the initial clone of the repository slower, but subsequent operations such as commit, blame, diff, merge, and log dramatically faster. Git also has excellent support for branching, merging, and rewriting repository history, which has led to many innovative and powerful workflows and tools. Pull requests are one such popular tool that allows teams to collaborate on Git branches and efficiently review each other's code. Git is the most widely used version control system in the world today and is considered the modern standard for software development.

How to use Git:

1. Create a "repository" (project) with a git hosting tool (like Bitbucket)
2. Copy (or clone) the repository to your local machine

3. Add a file to your local repo and "commit" (save) the changes
4. "Push" your changes to your main branch
5. Make a change to your file with a git hosting tool and commit
6. "Pull" the changes to your local machine
7. Create a "branch" (version), make a change, commit the change
8. Open a "pull request" (propose changes to the main branch)
9. "Merge" your branch to the main branch

What is GitHub?

GitHub is a Git repository hosting service that provides a web-based graphical interface. It is the world's largest coding community. Putting a code or a project into GitHub brings it increased, widespread exposure. Programmers can find source codes in many different languages and use the command-line interface, Git, to make and keep track of any changes.

GitHub helps every team member work together on a project from any location while facilitating collaboration. You can also review previous versions created at an earlier point in time. So now we know what Git and GitHub are. Time to gain a better understanding of the importance and relevance of what is GitHub by exploring its features.

GitHub is an increasingly popular programming resource used for code sharing. It's a social networking site for programmers that many companies and organizations use to facilitate project management and collaboration. GitHub is not only an affordable resource but also features a great open-source community. Since it is a cloud-based tool, the code is conveniently visible across the entire client organization, facilitating every participant's contributions.

GitHub allows collaboration with developers from all over the world. Open-source solutions like GitHub enable potential developers to contribute and share their knowledge to benefit the global community.

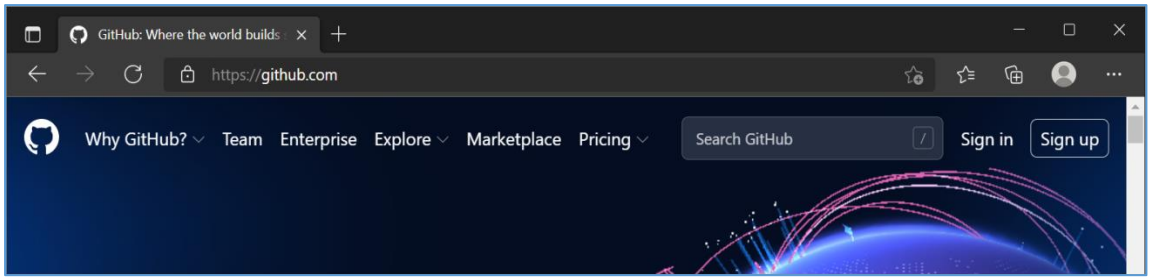
What is Version Control System?

The Git version control system, as the name suggests, is a system that records all the modifications made to a file or set of data so that a specific version may be called up later if needed. The system makes sure that all the team members are working on the file's latest version, and everyone can work simultaneously on the same project.

How to create a GitHub account to use with Visual Studio

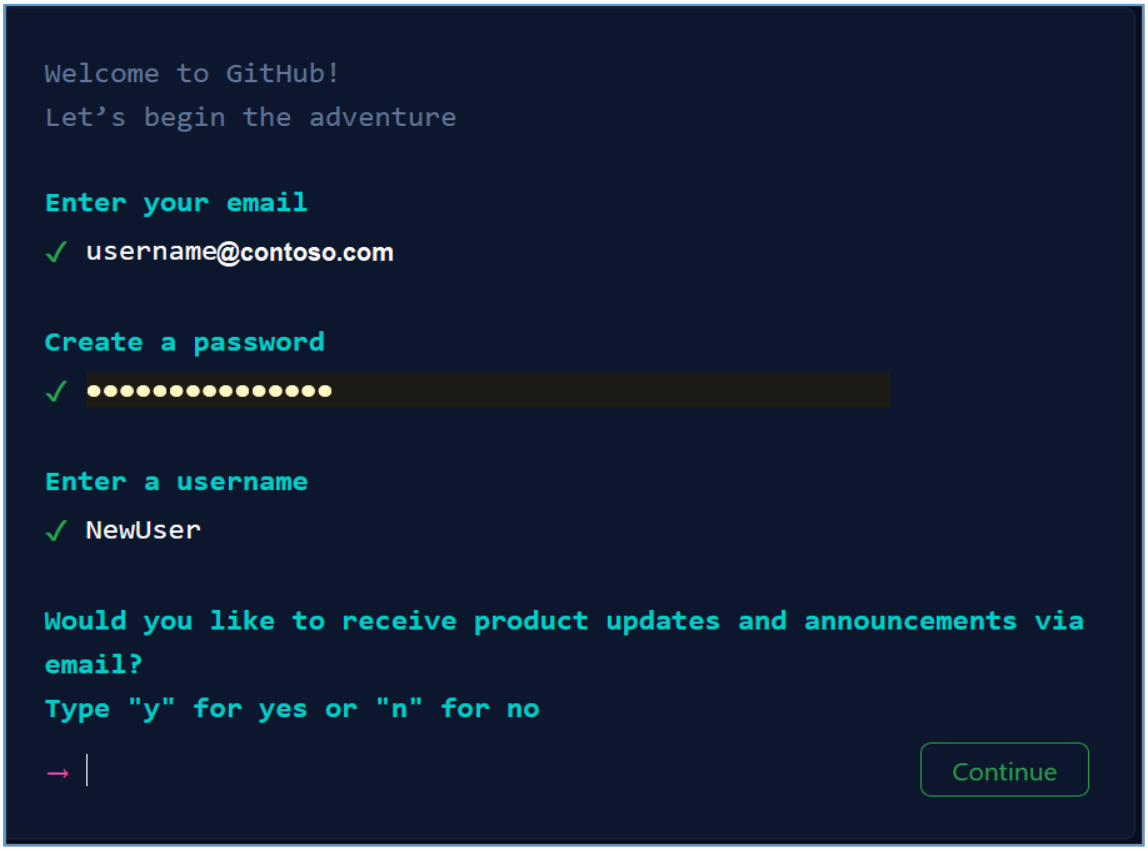
If you don't already have a GitHub account, here's how to create one.

Step 1: Open https://github.com in a web browser, and then select **Sign up**.



Step 2: Enter your email address.

Step 3: Create a password for your new GitHub account, and Enter a username, too. Next, choose whether you want to receive updates and announcements via email, and then select Continue.



Step 4: Verify your account by solving a puzzle. Select the Start Puzzle button to do so, and then follow the prompts.

Step 5: After you verify your account, select the Create account button.

Step 6: GitHub sends a launch code to your email address. Type that launch code in the **Enter code** dialog, and then press **Enter**.

Step 7: GitHub asks you some questions to help tailor your experience. Choose the answers that apply to you in the following dialogs:

- How many team members will be working with you?
- What specific features are you interested in using?

Step 8: On the Where teams collaborate and ship screen, you can choose whether you want to use the Free account or the Team account. To choose the Free account, select the Skip personalization button.

Commands to PUSH and PULL to version control repository to GitHub:

Sr. No.	Git Commands	Description
1	git init	Initializes a new Git repository. If you want to place a project under revision control, this is the first command you need to learn.
2	git clean	Removes untracked files from the working directory. This is the logical counterpart to git reset, which (typically) only operates on tracked files.
3	git clone	Creates a copy of an existing Git repository. Cloning is the most common way for developers to obtain a working copy of a central repository.

4	git commit	Takes the staged snapshot and commits it to the project history. Combined with git add, this defines the basic workflow for all Git users.
5	git commit --amend	Passing the --amend flag to git commit lets you amend the most recent commit. This is very useful when you forget to stage a file or omit important information from the commit message.
6	git config	A convenient way to set configuration options for your Git installation. You'll typically only need to use this immediately after installing Git on a new development machine.
7	git fetch	Fetching downloads a branch from another repository, along with all of its associated commits and files. But, it doesn't try to integrate anything into your local repository. This gives you a chance to inspect changes before merging them with your project.
8	git log	Lets you explore the previous revisions of a project. It provides several formatting options for displaying committed snapshots.
9	git merge	A powerful way to integrate changes from divergent branches. After forking the project history with git branch, git merge lets you put it back together again.
10	git pull	Pulling is the automated version of git fetch. It downloads a branch from a remote repository, then immediately merges it into the current branch. This is the Git equivalent of svn update.
11	git push	Pushing is the opposite of fetching (with a few caveats). It lets you move a local branch to another repository, which serves as a convenient way to publish contributions. This is like svn commit, but it sends a series of commits instead of a single changeset.
12	git rebase	Rebasing lets you move branches around, which helps you avoid unnecessary merge commits. The resulting linear history is often much easier to understand and explore.
13	git reflog	Git keeps track of updates to the tip of branches using a mechanism called reflog. This allows you to go back to changesets even though they are not referenced by any branch or tag.

CONCLUSION:

_____

_____

_____

_____

_____

OUTPUT:

(Attach Screenshots of your output in sequence)

ASSIGNMENT NO. :02(B)

AIM:

Create Docker Container Environment (NVIDEIA Docker or any other)

PRE-REQUISITE:

1. Knowledge of Visual Studio Code.
2. Knowledge of Python, JavaScript.
3. Knowledge of Node.js

OBJECTIVE:

1. To study basics of Docker.
2. To install Dockers on System
3. Add Docker (from Microsoft) extension in Visual Studio Code.

THEORY:

What is Docker?

The Docker (previously called “dot-docker”) platform allows you to package up your application(s) and deliver them to the cloud without any dependencies. If you have begun creating cloud-based applications, you should get a strong understanding of the benefits of Docker. This platform is a great way to create isolated environments and automatically scale them up or down. Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications.

Docker is a software platform that allows you to build, test, and deploy applications quickly. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime. Docker's container-based platform allows for highly portable workloads. Docker containers can run on a developer's local laptop, on physical or virtual machines in a data center, on cloud providers, or in a mixture of environments.

The benefits of Docker:

The benefits of Docker in building and deploying applications are

1. Caching a cluster of containers
2. Flexible resource sharing
3. Scalability - many containers can be placed in a single host
4. Running your service on hardware that is much cheaper than standard servers

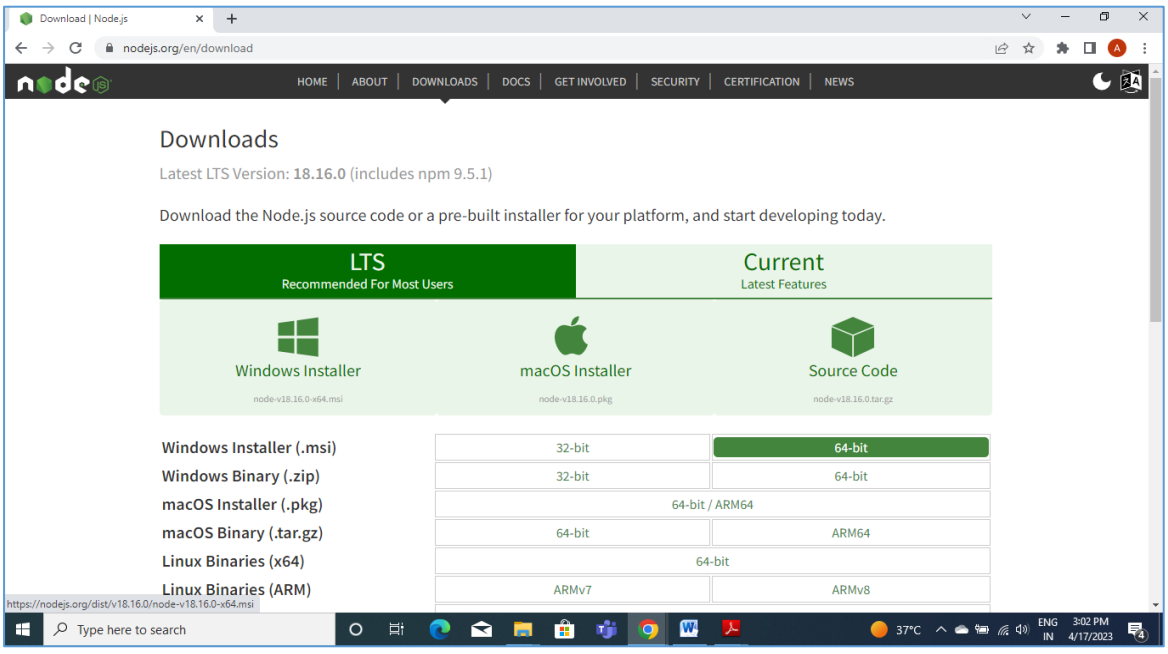
5. Fast deployment, ease of creating new instances, and faster migrations.
6. Ease of moving and maintaining your applications
7. Better security, less access needed to work with the code running inside containers, and fewer software dependencies

NVIDIA Container:

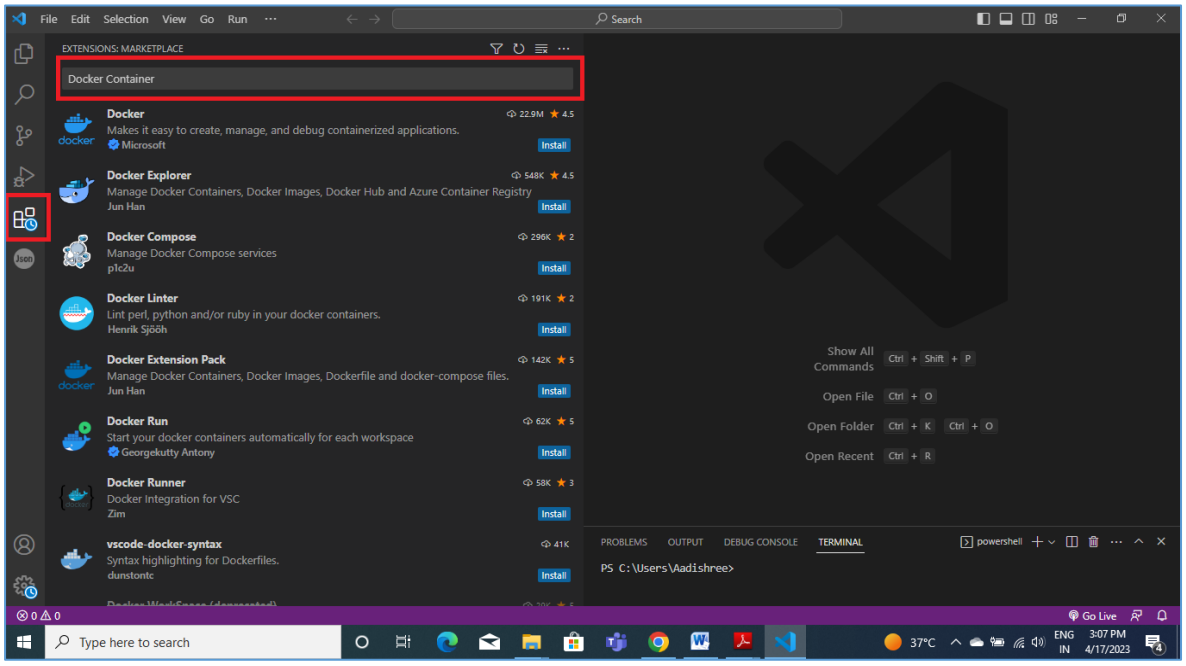
NVIDIA-Docker is essentially a wrapper around the Docker command that transparently provisions a container with the necessary components to execute code on the GPU. It is only absolutely necessary when using NVIDIA - Docker run to execute a container that uses GPUs. The NVIDIA Container Runtime for Docker, also known as nvidia-docker2 enables GPU-based applications that are portable across multiple machines, in a similar way to how Docker® enables CPU-based applications to be deployed across multiple machines. It accomplishes this through the use of Docker containers. A container is a standard unit of software that packages up code and all its dependencies, so the application runs quickly and reliably from one computing environment to another.

Basic requirements for Docker Installation:

1. Install Node.js from below link. https://bit.ly/3ailfen



2. Add Docker (from Microsoft) extension in VS Code.



Steps to Implement the Assignment:

Step-1: Create a folder (Note: Folder name should be in lowercase). For Example: Give the folder name as wadpartb

Step-2: Create any file with any extension (such as .js, .txt, .py, etc....) in the same above folder (Here, wadpartb) and write some content in the file. Let the file name be: main.js

The file contains the following code:

```
console.log ("Name: Your Name");
console.log ("Class: TE");
console.log ("Subject: LP-II Web Application Development");
console.log ("We are implementing Assignment-2b");
```

Step-3: Create file in the same folder with file name as- Docker file and press enter key. (Note: Don't add any file extension after the file name) File: Docker file

Step-4: Write following code in Docker file (Created in Step-3) and save it.: FROM node: alpine COPY. / wadpartb CMD node / wadpartb /main.js

Step-5: Open terminal in VS Code and type the following command: Docker –version. (We will see the version of your Docker if it is installed successfully.

Step-6: In the same terminal in VS Code, type the following command: node main.js (It will display the contents in your file which is created in Step-2.)

Step-7: In the same terminal in VS Code, type the following command:

Docker build -t wadpartb. (We will see that it is successfully built.)

Step-8: In the same terminal in VS Code, type the following command: Docker images

CONCLUSION:

_____

_____

_____

_____

_____

OUTPUT:

(Attach Screenshots of your output in sequence)

ASSIGNMENT NO. : 02(C)

AIM:

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

PRE-REQUISITE:

1. Knowledge of HTML, CSS & JavaScript.
2. Knowledge of Node. js and npm
3. Basic idea of the MVC architecture & TypeScript.

OBJECTIVE:

1. To study basics of Angular.
2. To create independent, reusable, and testable modules and components
3. To develop web application with Front End Technologies.

THEORY:

What is Angular?

Angular JS is an open source JavaScript framework that is used to build web applications. It can be freely used, changed and shared by anyone. Angular Js is developed and maintained by Google. It is an excellent framework for building single phase applications and line of business applications. It can be added to an HTML page with a <script> tag. AngularJS extends HTML attributes with Directives, and binds data to HTML with Expressions.

Angular is a TypeScript-based, free and open-source web application framework led by the Angular Team at Google and by a community of individuals and corporations. Angular is a complete rewrite from the same team that built AngularJS. Angular is a development platform, built on TypeScript. As a platform, Angular includes: A component-based framework for building scalable web applications. A collection of well-integrated libraries that cover a wide variety of features, including routing, forms management, client-server communication. Its chief purpose is to develop single-page applications. Angular is a framework that provides developers with a standard structure and lets users create large, scalable applications in a way that can be easily maintained.

Angular is considered one of the most popular JavaScript libraries for building Web applications. It lets you use HTML as a template language and allows users to extend HTML syntax to express their application's components succinctly and clearly. In addition, Angular's data binding and dependency injection dispose of much of the code developers

would otherwise have to write, saving time and making the entire development process much more efficient. Even better, the development process happens within the browser, which through APIs enables developers to work with the back-end infrastructure more efficiently.

What Is Angular CLI?

Angular CLI is a command-line interface tool (that's what the CLI stands for) used to develop, initialize, scaffold, and maintain Angular applications directly from a command shell. It is the official tool used for initializing and working with Angular projects. In addition, Angular CLI automates your development workflow saving you from dealing with complex configurations and building tools like TypeScript, Webpack, etc. Angular CLI is a tool a command-line interface that builds applications based on your input and helps you add new services, directives, and components from the command line.

Sample Angular Code:

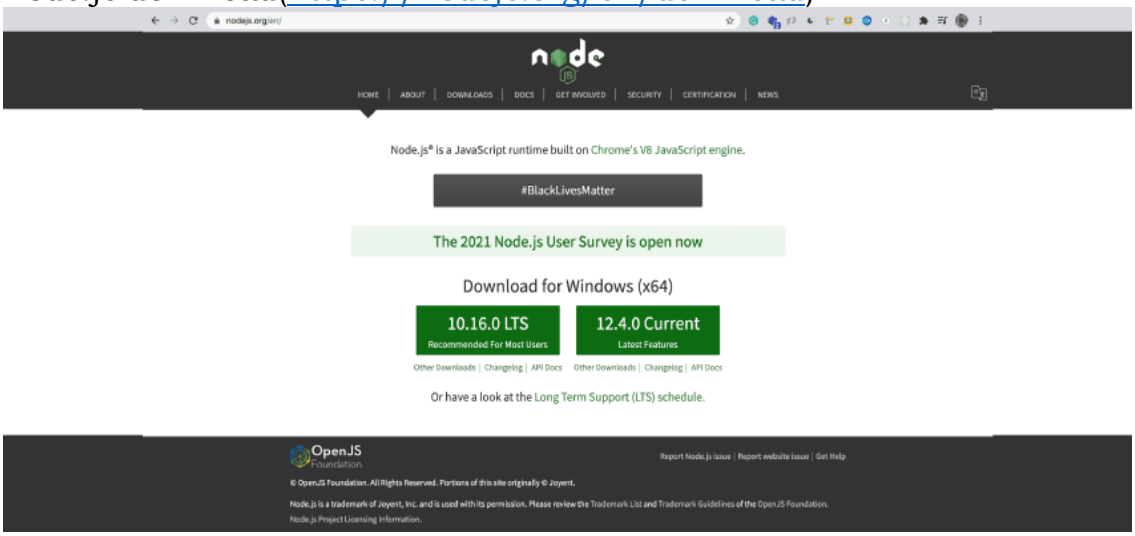
```
<!DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angul
ar.min.js"></script>
<body>

  <p>Try to change the names.</p>
  <div ng-app="myApp" ng-controller="myCtrl">
       First Name: <input type="text" ng-model="firstName"><br>
       Last Name: <input type="text" ng-model="lastName"><br>
      <br>
      Full Name: {{firstName + " " + lastName}}
  </div>
  <script>
       var app = angular.module('myApp', []);
       app.controller('myCtrl', function($scope) {
        $scope.firstName= "FirstName";
        $scope.lastName= "LastName";
      });
  </script>
</body>
</html>
```

How to install Angular :

Step 1: Install Node/npm on your machine.

- Download the Node.js installer and install it on your system.(Link for Node.js download(https://nodejs.org/en/download)



- Once you have installed Node.js it will automatically install NPM on your machine, and you can check the version installed using the below command. Open the command prompt and type “npm-v” to check the version and installation.

```
npm -v
```

Step 2: Install TypeScript on your machine.

- Open this link and download the TypeScript installer, placing it in your device.(https://www.npmjs.com/package/typescript)
- To install TypeScript, enter the following command in the Terminal Window.

```
$ npm install typescript --save-dev      //As dev dependency
$ npm install typescript -g              //Install as a global module
$ npm install typescript@latest -g       //Install latest  version
```

- To verify the installation was successful, enter the command $ tsc -v in the Terminal Window.

Step 3: Installing Angular CLI

a. Before you install Angular on your local Windows system, you must have these resources in place.

- Node.js. Angular needs an active LTS or maintenance LTS version. You should already have this installed back in Step 1. You can verify your Node.js installation by typing "$ node -v" in the command prompt.

- NPM Package Manager. Angular, Angular CLI, and all Angular applications rely on npm packages to accommodate most of their functions and features. Therefore, you will need an NPM packager manager to download and install the npm packages. Use the "npm-v" command to verify that you have the client installed.

- If you can't use a node version manager. If your system doesn't let you use an nvm, you can use this Node.js installer or this Node Source installer.

b. Install Angular CLI with the npm package manager.

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

Step 4: Setting Up a Basic Angular CLI Workflow.

- First, invoke the tool on the command line via the ng executable. If you need online help, it's available on the command line. Enter the following command to list the commands or options for any given command (e.g., generate) plus a short description.

```
ng help
ng generate –help
```

- When you're ready to create, build, and serve a new, basic Angular project on your development server, go to your new workspace's parent directory and use the following commands:

```
ng new my-first-project
cd my-first-project
ng serve
```

Step 5: Creating a Workspace and an Initial Application.

Now that you have all these resources installed, it's time to create a workspace and your initial application.

- Run the CLI command "ng new," then provide the name "my-app" as seen below:

```
ng new my-app
```

- This "ng new" command will prompt you for information regarding features you want to include in the initial app. Press "Enter" or the "Return" key to accept the defaults.

Angular CLI will install the required npm packages and any other dependencies, although the process may take a few minutes. But when it's complete, you have a new workspace, ready to go!

Step 6: Running Your Application

Angular CLI features a server so you can build and serve your app locally.

- Access your workspace folder, such as “my-app.”
- Run this command:

cd my-app ng serve –open

“Ng serve” launches the server, monitors your files, and rebuilds the app when you make any file changes. The “—open” (alternately, just “-o”) option will automatically open your browser to http://localhost:4200/. If the installation and setup were successful, you would see a page like the following:

CONCLUSION:

_____

_____

_____

_____

_____

OUTPUT:

(Attach Screenshots of your output in sequence)

ASSIGNMENT NO. :03	**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.
GIVEN DATE:	/ /2023
SUBMISSION DATE:	/ /2023
SIGN. OF STUDENT:	

REGULARITY		PRESENTATION		UNDERSTANDING		TIMELY SUBMISSION		TOTAL MARKS	
	06		06		08		05		25

SIGN. OF FACULTY:	
REMARKS:	

ASSIGNMENT NO. : 03(A)

AIM:

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

PRE-REQUISITE:

1. Knowledge of HTML5 & CSS.
2. Knowledge of JavaScript.
3. Understanding of other web technologies such as AJAX.

OBJECTIVE:

1. To study Node.JS.
2. To install extensions required to execute Node.js application in Visual Studio Code.
3. Create a web application using Node.JS.

THEORY:

What is Node.JS?

Node.js is an open source server environment. Node.js is a very powerful JavaScript-based platform built on Google Chrome's JavaScript V8 Engine. It is used to develop I/O intensive web applications like video streaming sites, single-page applications, and other web applications. Node.js is open source, completely free, and used by thousands of developers around the world.

Node.JS is an open-source, cross-platform JavaScript library and runtime environment. You can use it to run web apps outside the client's browser. Ryan Dahl developed Node.js in 2009, and the latest version, v18, was released in October 2022. This IDE is used to develop server-side web apps, and since it uses an asynchronous, event-driven model, it is the best option for creating data-intensive apps.

Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js was developed by Ryan Dahl in 2009. Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices. Node.js is an cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

Node.js = Runtime Environment + JavaScript Library

What is Npm:

NPM is a package manager for Node.js packages, or modules if you like. www.npmjs.com hosts thousands of free packages to download and use.

The NPM program is installed on your computer when you install Node.js. npm is the package manager for the Node JavaScript platform. It puts modules in place so that node can find them, and manages dependency conflicts intelligently. It is extremely configurable to support a wide variety of use cases. Most commonly, it is used to publish, discover, install, and develop node programs.

How to Install Node.js and NPM on Windows:

We have decided to create an application using Node.JS, first of all, you have to install Node.JS on your Windows system.

Step 1: Download the Installer.

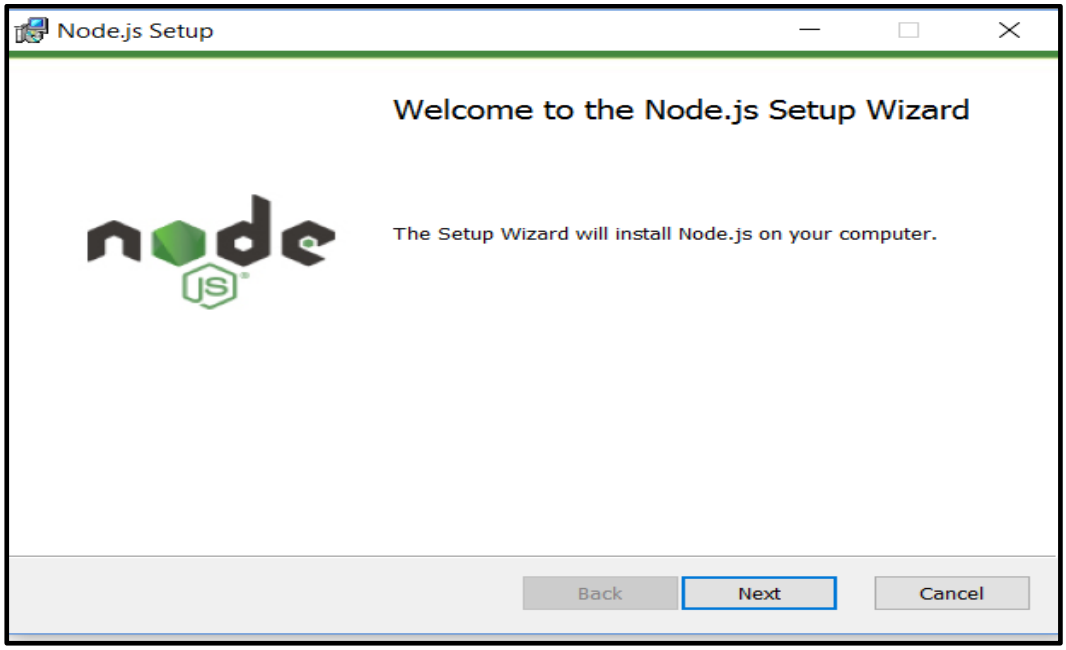
Download the Windows Installer from Node.JS official website. Make sure you have downloaded the latest version of Node.JS. It includes the NPM package manager.

After the download of the installer package, install it with a double-click on it.

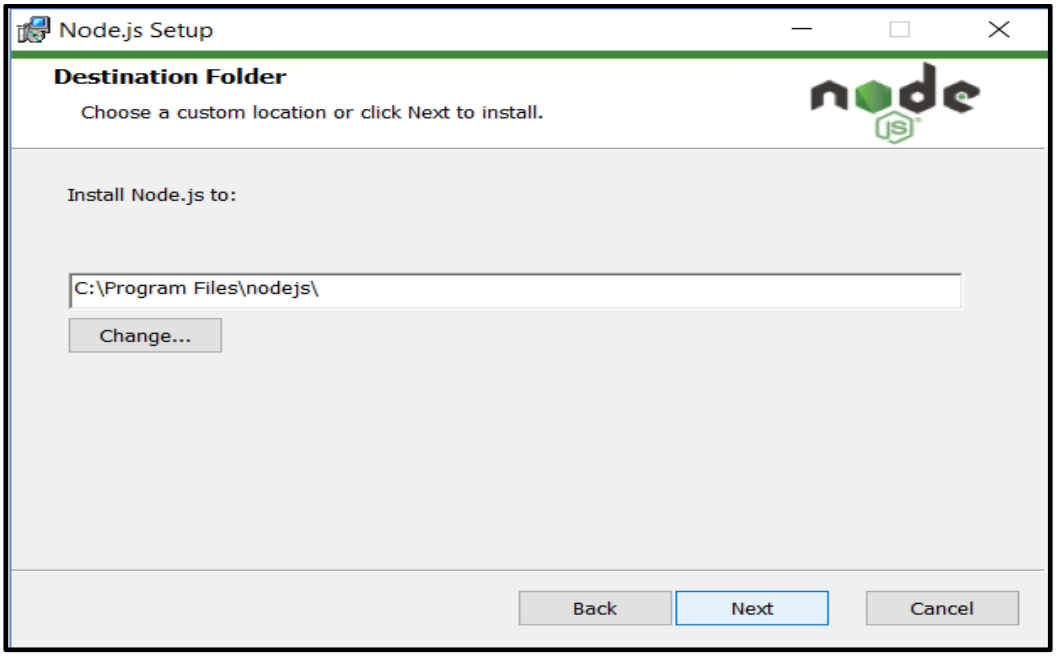
Step 2: Install Node.js and NPM.

After choosing the path, double-click to install .msi binary files to initiate the installation process. Then give access to run the application.

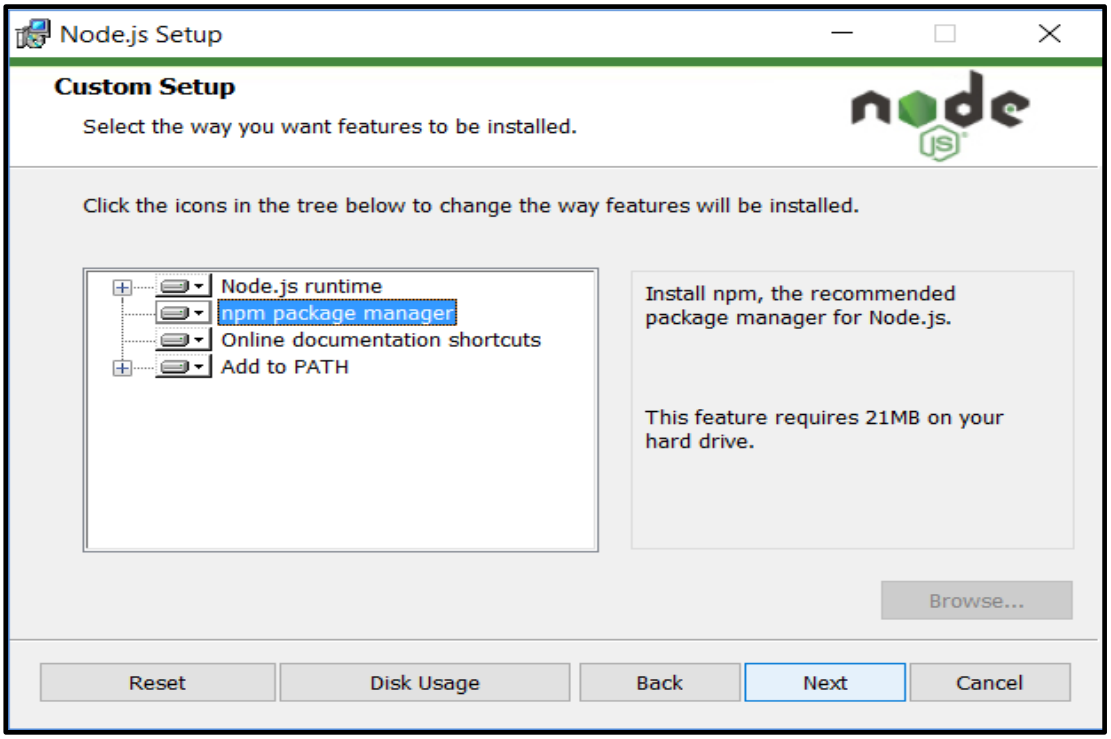
You will get a welcome message on your screen and click the “Next” button. The installation process will start



Step 3: Choose the desired path where you want to install Node.js.



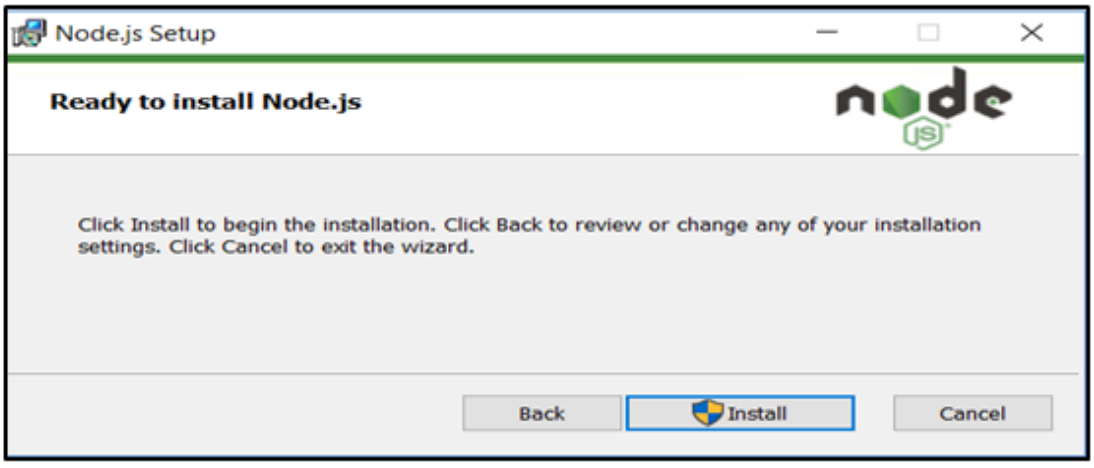
Step 4: By clicking on the Next button, you will get a custom page setup on the screen. Make sure you choose npm package manager , not the default of Node.js runtime . This way, we can install Node and NPM simultaneously.



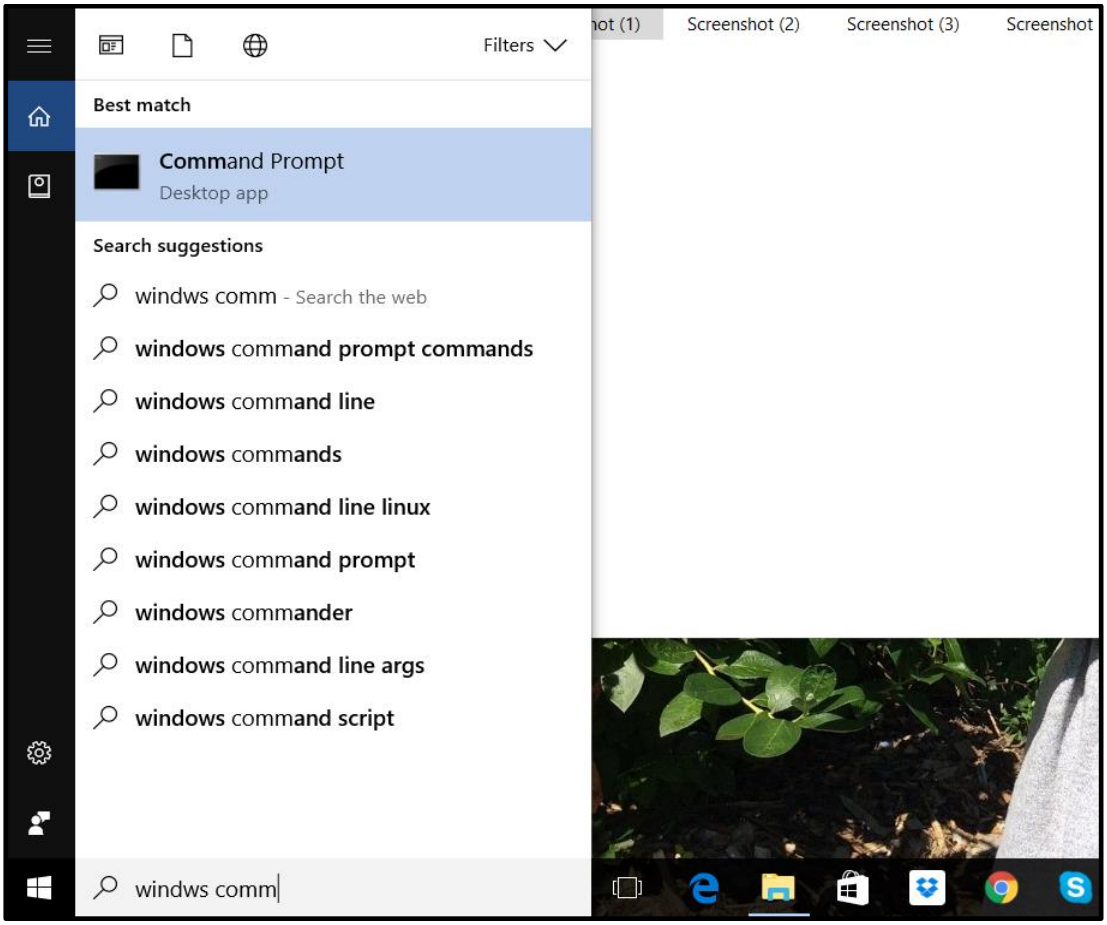
You should have 143MB of space to install Node.js and npm features. The following features will be installed by default:

- Node.js runtime
- Npm package manager
- Online documentation shortcuts
- Add to Path

Step 5: The setup is ready to install Node and NPM. Let's click on the **Install** button.



Step 6: Check Node.js and NPM Version



Command Prompt window will appear on the screen.
To confirm Node installation, type **node -v** command.
To confirm NPM installation, type **npm -v** command.

Official Website to download Node.js: https://nodejs.org

Sample Node.js Code:

```
var http = require('http');
http.createServer(function (req, res) {
res.writeHead(200, {'Content-Type': 'text/html'});
res.end('Hello World!');
}).listen(8080);
```

Save the file on your computer: C:\Users\Your Name\myfirst.js

The file you have just created must be initiated by Node.js before any action can take place.
Start your command line interface, write node myfirst.js and hit enter:

Initiate "myfirst.js":

```
C:\Users\Your Name>node myfirst.js
```

Now, your computer works as a server!

If anyone tries to access your computer on port 8080, they will get a "Hello World!" message in return!
Start your internet browser, and type in the address: http://localhost:8080

CONCLUSION:

_____

_____

_____

_____

_____

OUTPUT:

(Attach Screenshots of your output in sequence)

ASSIGNMENT NO. : 03(B)

AIM:

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

PRE-REQUISITE:

1. Knowledge of HTML5 & CSS.
2. Knowledge of JavaScript.
3. Knowledge of Angular

OBJECTIVE:

1. To study Node.JS.
2. To install extensions required to execute Express.JS application in Visual Studio Code.
3. Create a web application using Node.JS and Express.JS.
4. Create a web application using MongoDB.

THEORY:

What is Express.JS?

Express was developed by TJ Holowaychuk and is maintained by the Node.js foundation and numerous open source contributors. Express.js, or simply Express, is a back end web application framework for building RESTful APIs with Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js. Express is a node js web application framework that provides broad features for building web and mobile applications. It is used to build a single page, multipage, and hybrid web application. It's a layer built on the top of the Node js that helps manage servers and routes.

Express is a fast, assertive, essential and moderate web framework of Node.js. It provides a robust set of features to develop web and mobile applications. Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. It is an open source framework developed and maintained by the Node.js foundation.

ExpressJS is a web application framework that provides you with a simple API to build websites, web apps and back ends. Express provides a minimal interface to build our applications. It provides us the tools that are required to build our app. It is flexible as there are numerous modules available on npm, which can be directly plugged into Express.

Features of Express framework:

- It can be used to design single-page, multi-page and hybrid web applications.
- It allows to setup middleware's to respond to HTTP Requests.
- It defines a routing table which is used to perform different actions based on HTTP method and URL.
- It allows to dynamically render HTML Pages based on passing arguments to templates.

How to Install Express.js on Windows:

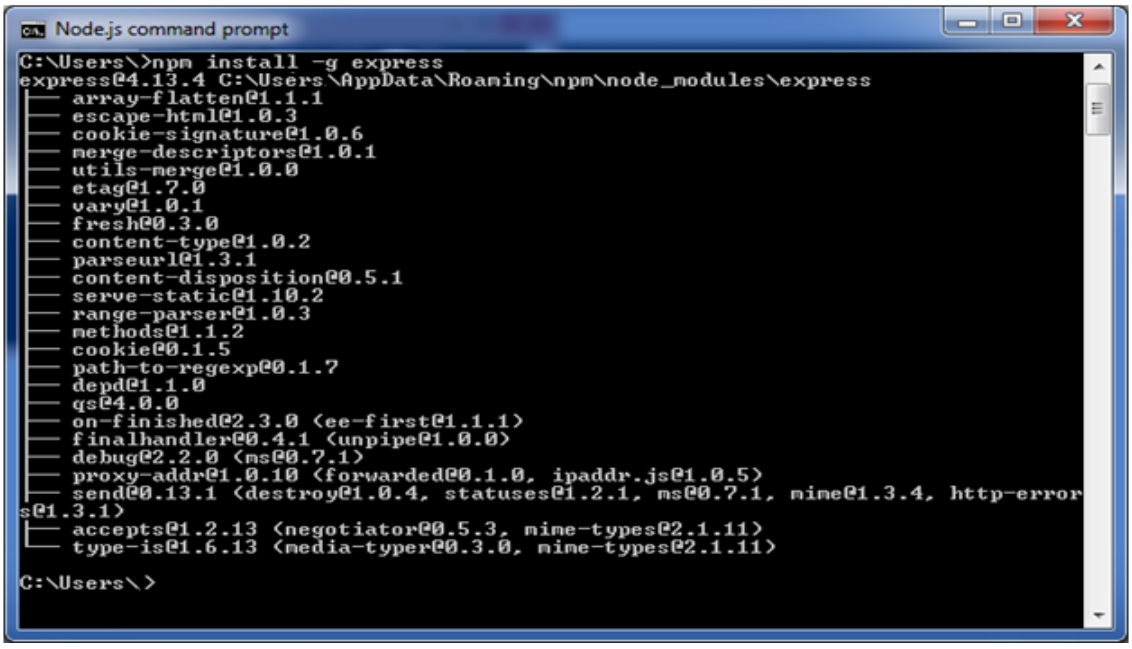
We have decided to create an application using Node.JS, first of all, you have to install Node.JS on your Windows system.

Firstly, you have to install the express framework globally to create web application using Node terminal.

Step 1: Install Express framework

Use the following command to install express framework globally.

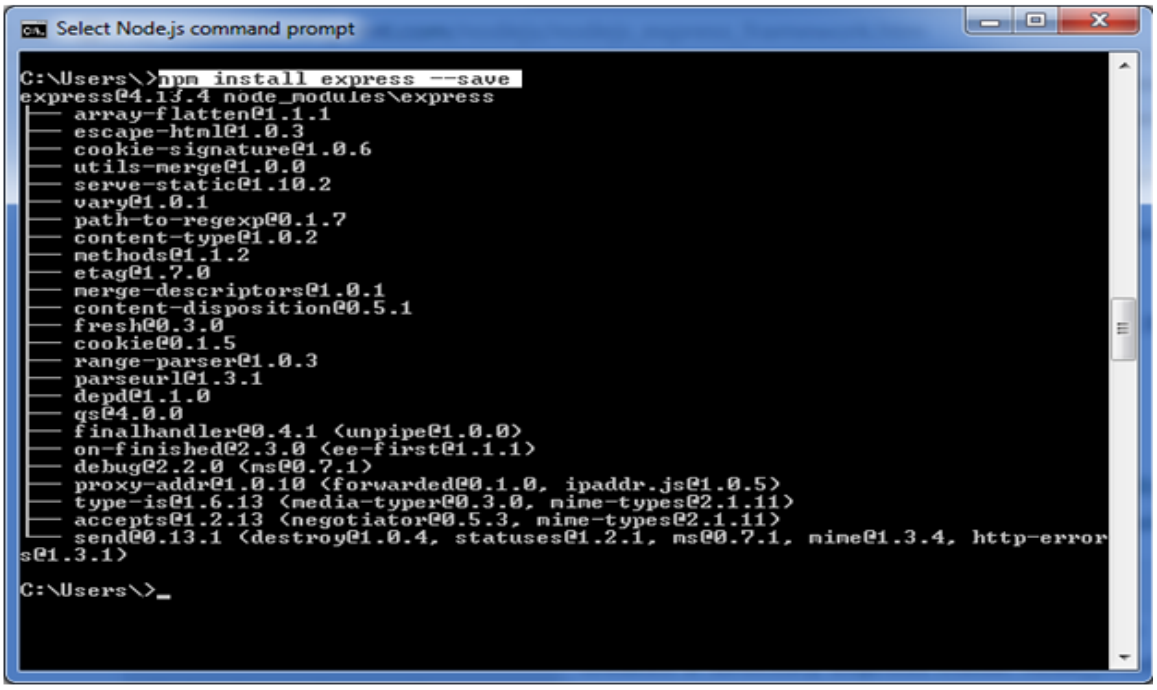
```
npm install -g express
```



Step 2: Install Express

Use the following command to install express:

```
npm install express --save
```



The above command install express in node_module directory and create a directory named express inside the node_module. You should install some other important modules along with express. Following is the list:

- **Body-parser:** This is a node.js middleware for handling JSON, Raw, Text and URL encoded form data.
- **Cookie-parser:** It is used to parse Cookie header and populate req.cookies with an object keyed by the cookie names.
- **Multer:** This is a node.js middleware for handling multipart/form-data.

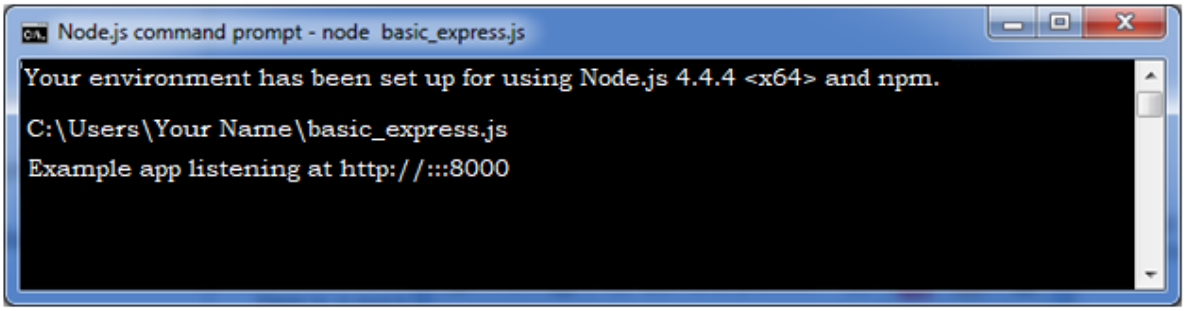
Sample express.js Code:

File: basic_express.js

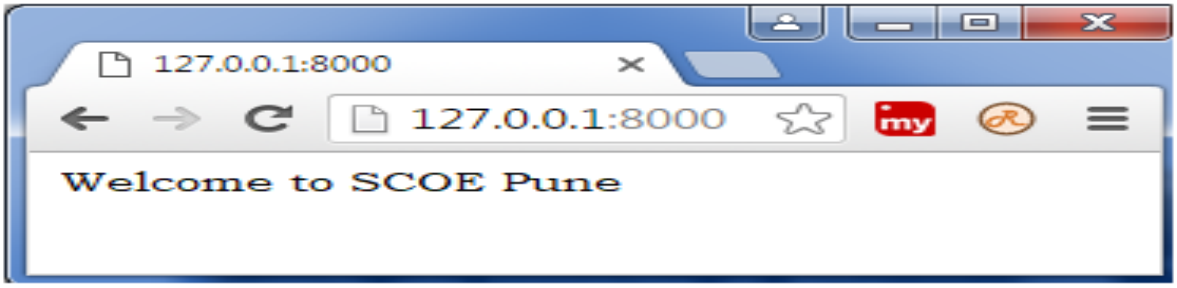
```
var express = require('express');
var app = express();
app.get('/', function (req, res) {
res.send('Welcome to SCOE Pune!');
});
var server = app.listen(8000, function () {
var host = server.address().address;
var port = server.address().port;
```

```
console.log('Example app listening at http://%s:%s', host, port);
});
```

Save the file on your computer: C:\Users\Your Name\basic_express.js



Output:



What is CRUD Operations:

The CRUD paradigm stands for the four primitive database operations that are CREATE, READ, UPDATE and DELETE. CRUD (Create, Read, Update, Delete) operations allow you to work with the data stored in MongoDB. The CRUD operation documentation is categorized in two sections: Read Operations find and return documents stored within your MongoDB database. Write Operations insert, modify, or delete documents in your MongoDB database.

CRUD operations in Express js and MongoDB:

1. Node.js MongoDB Rest CRUD API overview.
2. Demo Video.
3. Create Node.js App.
4. Setup Express web server.
5. Configure MongoDB database & Mongoose.
6. Define Mongoose.
7. Define the Mongoose Model.
8. Create the Controller. Create a new object.

How to create a CRUD in Node js:

Step 1: We create the repository and install the dependencies.

The entry point is the server.js file.

```
mkdir express-api
cd express-api
npm init
npm install express helmet morgan body-parser monk joi dotenv --save
npm install nodemon --save-dev
```

About the packages

express: It is a minimal and flexible Node.js web application framework.

helmet: It helps in securing HTTP headers in express applications.

morgan: It is an HTTP request logger middleware for Node. js

body-parser: It is responsible for parsing the incoming request bodies.

monk: A tiny layer that provides substantial usability improvements for MongoDB usage.

joi: It is an object schema description language and object validator.

dotenv: It loads environment variables from a .env file.

nodemon: It restarts automatically the node application when file changes in the directory have been detected.

Step 2: Setup Express Web Server

./src/server.js

After choosing the path, double-click to install .msi binary files to initiate the installation process. Then give access to run the application.

You will get a welcome message on your screen and click the “Next” button. The installation process will start

```
const express = require('express');
const morgan = require('morgan');
const helmet = require('helmet');
const bodyParser = require('body-parser');

require('dotenv').config();

const app = express();
const monk = require('monk');

app.use(helmet());
app.use(morgan('dev'));
app.use(bodyParser.json());

const port = process.env.PORT || 8080;
app.listen(port, () => {
  console.log(`Listening on port ${port}`);
});
```

Step 3: Create and configure the .env file

./.env

It contains all the environment variables that we use.

```
DB_URL = localhost/my-employees
TEST_DB_URL = localhost/test-my-employees
PORT = 5000
```

Step 4: Create the data schema and define the validation rules

Create the data schema and define the validation rules that the properties

```
const Joi = require('joi');
const schema = Joi.object({
    name: Joi.string()
        .min(3)
        .max(30)
        .required(),
    job: Joi.string()
        .min(3)
        .max(30)
        .required(),
})
module.exports = schema;
```

Step 5: Connect to the database

./src/db/connection.js

Connect to the database

```
const monk = require('monk');

let dbUrl = process.env.DB_URL;

if (process.env.NODE_ENV === 'test') {
  dbUrl = process.env.TEST_DB_URL;
}

const db = monk(dbUrl);

module.exports = db;
```

CONCLUSION:

_____

_____

_____

_____

_____

OUTPUT:

(Attach Screenshots of your output in sequence)

ASSIGNMENT NO. :04		**a.** Create a simple Mobile Website using jQuery Mobile. **b.** Deploy/Host Your web application on AWS VPC or AWS Elastic Beanstalk					
GIVEN DATE:		/ /2023					
SUBMISSION DATE:		/ /2023					
SIGN. OF STUDENT:							
REGULARITY		PRESENTATION		UNDERSTANDING		TOTAL MARKS	
	06		06		09		20
SIGN. OF FACULTY:							
REMARKS:							

ASSIGNMENT NO. : 04(A)

AIM:

Create a simple Mobile Website using jQuery Mobile.

PRE-REQUISITE:

1. Knowledge of HTML5 & CSS.
2. Knowledge of JQuery.
3. Knowledge of JavaScript.

OBJECTIVE:

1. To study JQuery Mobile.
2. To install extensions required to execute Node.js application in Visual Studio Code.
3. Create a mobile website using jQuery Mobile.

THEORY:

What is jQuery Mobile?

It was developed by the jQuery project team in the year 2010 and written in JavaScript. jQuery Mobile is an HTML5 based user interface system designed to make responsive websites and apps that are accessible on all smartphone, tablet, and desktop devices. jQuery Mobile is a framework for creating mobile web applications. jQuery Mobile works on all popular smartphones and tablets. jQuery Mobile uses HTML5 & CSS3 for laying out pages with minimal scripting.

jQuery Mobile framework takes the "write less, do more" mantra to the next level: Instead of writing unique applications for each mobile device or OS, the jQuery mobile framework allows you to design a single highly-branded responsive website or application that will work on all popular smartphone, tablet, and desktop platforms.

jQuery Mobile is a user interface framework, which is built on jQuery Core and used for developing responsive websites or applications that are accessible on mobile, tablet, and desktop devices. It uses the features of both jQuery and jQuery UI to provide API features for mobile web applications.

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.

- The progressive enhancement brings a unique functionality to all mobile, tablet, and desktop platforms and adds efficient page loads and wider device support.

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.
- It integrates HTML5, CCS3, jQuery and jQuery UI into one framework for creating pages with minimal scripting.
- It includes Ajax navigation system that uses animated page transitions.

Advantages of jQuery Mobile:

- It is easy to learn and develop applications if you have knowledge of HTML5, CSS3 features.
- It is cross-platform and cross-browser compatible so you don't have to worry about writing different code for each device resolution.
- You can create the custom theme using ThemeRoller without writing the line of code. It supports all HTML5 browsers.
- It uses HTML5 along with JavaScript for easy development of web applications.
- It is built in a way that allows the same code to automatically scale from the mobile screen to desktop screen.

Disadvantages of jQuery Mobile:

- There are limited options for CSS themes, so sites can look similar which are built by these themes.
- Applications which are developed using jQuery Mobile are slower on mobiles.
- It becomes more time consuming when you combine jQuery mobile with other mobile frameworks.
- Difficult to provide complete customized visual design.
- All the features in a device cannot be accessed by JavaScript in a browser.

How to use?

We can add jQuery Mobile to our project in two ways:

1. Downloading the package files
2. Using CDN Link

1. Downloading Package:

When you will open the main website of jQuery Mobile i.e. https://www.jquerymobile.com/, you will see there are two options to download jQuery mobile library.

1.1 Custom Download: This option is used to download a customized version of library.

1.2 Latest Stable: This option is used to get the stable and latest version of jQuery mobile library.

2. Using CDN Link:

We need to include three CDN links for including jQuery Mobile to our project. These three links are for:

2.1 Including jQuery Mobile stylesheets

2.2 Including the jQuery library

2.3 Including the jQuery Mobile library

<link rel=”stylesheet” href=”https://code.jquery.com/mobile/1.4.5/jquery.Mobile-1.4.5.min.css”>
<script src=”http://code.jquery.com/jquery-1.11.1.min.js”></script>
<script src=”http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js”> </script>

jQuery-mobile Getting started with jQuery-mobile Setup:

1. Create a library (or lib) folder inside of your application (inside the www folder).
2. Create a jQuery-mobile folder inside the lib folder.
3. Place all of the jQuery-mobile files that you have extracted inside of the jQuery-mobile folder.

How to install jQuery UI in Visual Studio Code?

1. Create a folder and add some file in it (or open an existing one)
2. Install TSD globally using the following command: npm install tsd -g.
3. Install the jQuery type definitions file in our project using the following command: tsd install jquery --save.
4. Add a reference to our tsd. d. ts file in our app.

Sample jQuery Mobile Code:

This is a basic sample code of a website using some jQuery Mobile classes.

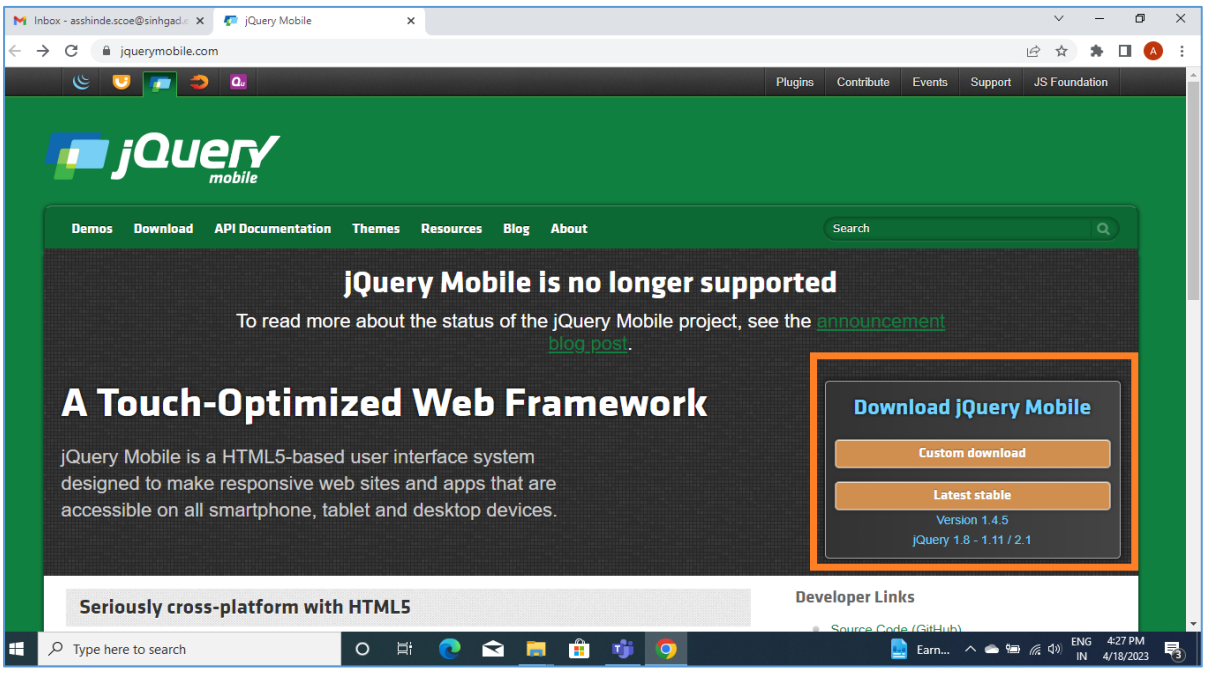
```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content= "width = device-width, initial-scale = 1">
```

```
    <link rel="stylesheet" href="https://code.jquery.com/mobile/1.4.5/
     jquery.mobile-1.4.5.min.css">
    <script src= "https://code.jquery.com/jquery-1.11.3.min.js">  </script>
    <script src= "https://code.jquery.com/mobile/1.4.5/jquery.mobile-
     1.4.5.min.js">     </script>
</head>
<body>
   <div data-role="page" id="page one">
      <div data-role="header">
         <h1>Hello!</h1>
      </div>
      <div data-role="main" class="ui-content">
         <h2> Welcome to SCOE</h2>
      </div>
   <div data-role="footer">
      <p>Hope you have a great day!</p>
      </div>
   </div>
</body>
</html
```

How to install jQuery on Windows:

Step 1: Download jQuery Mobile

Open Google chrome and open jQuery official link to download jquery Mobile setup. https://jquerymobile.com/



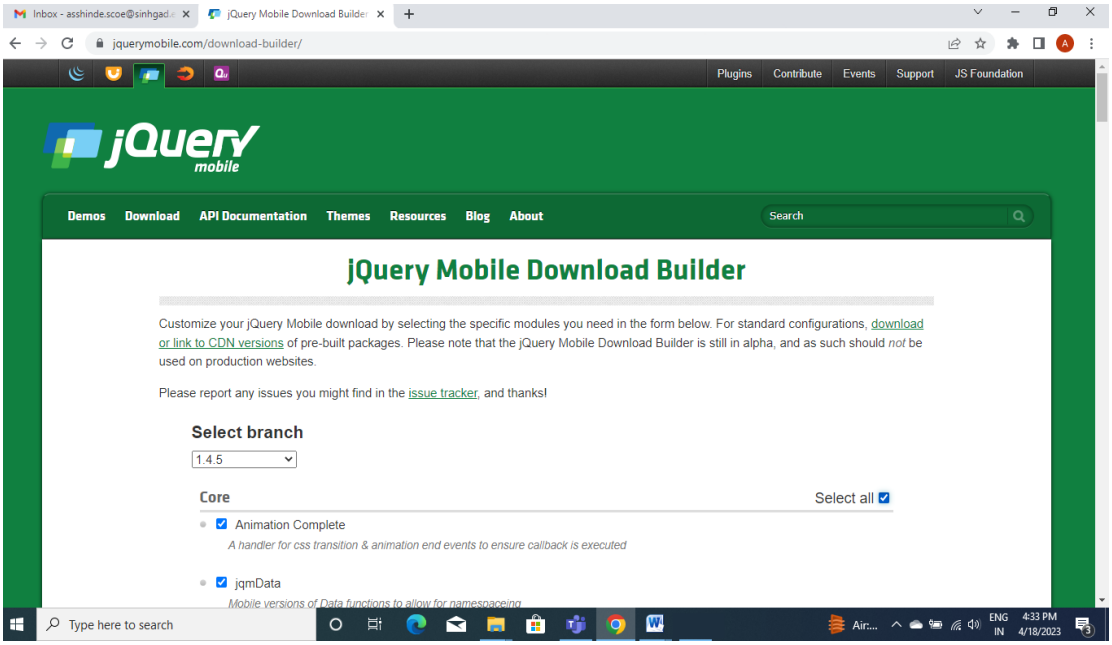
Step 2: There are two options to download jQuery mobile library.

2.1 Custom Download: Click this button to download a customized version of library.

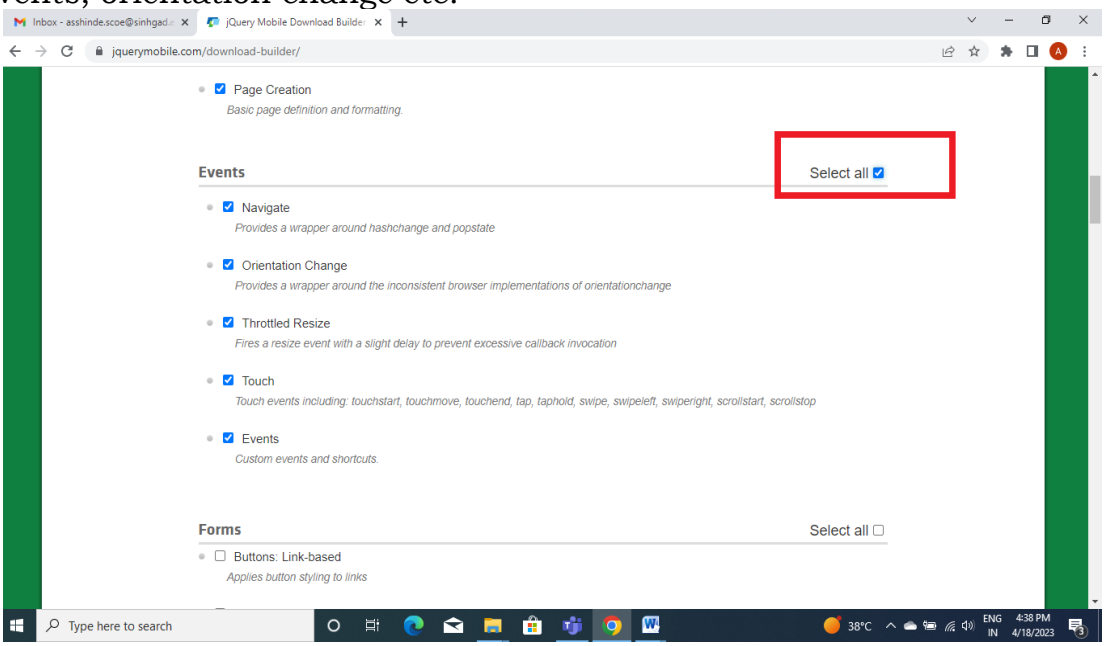
2.2 Latest Stable: Click this button to get the stable and latest version of jQuery mobile library.

2.1 Custom Download with Download Builder:

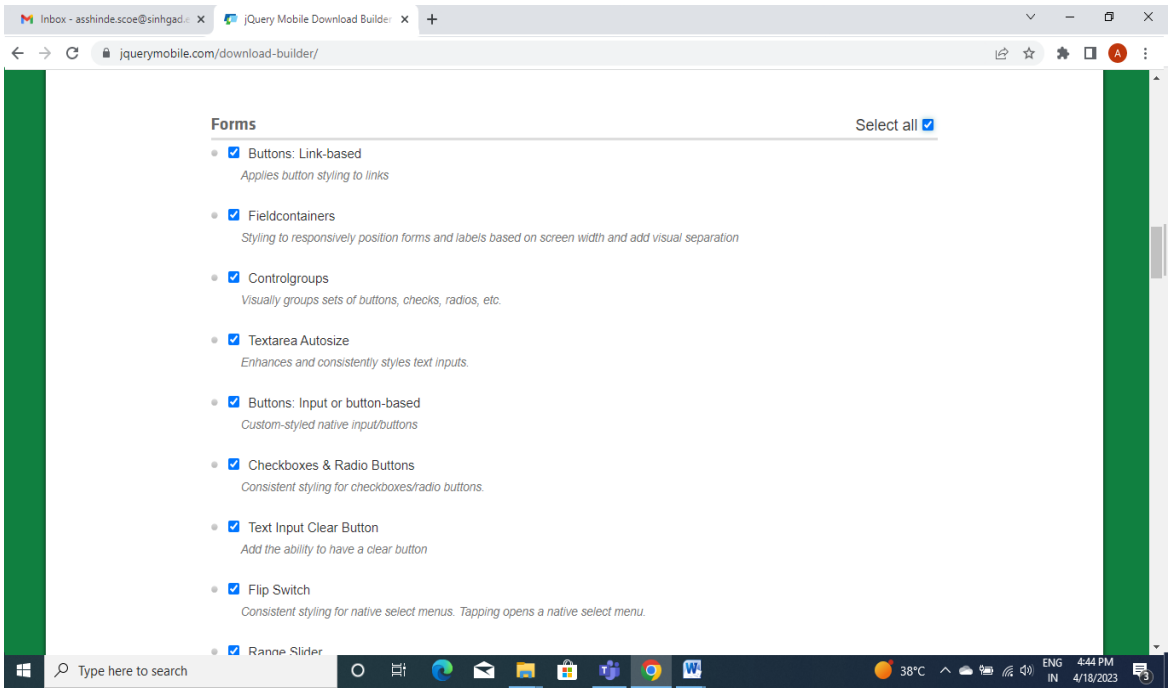
a. Click on "**Select Branch**" and select branch from dropdown list and click on "**Select all**"



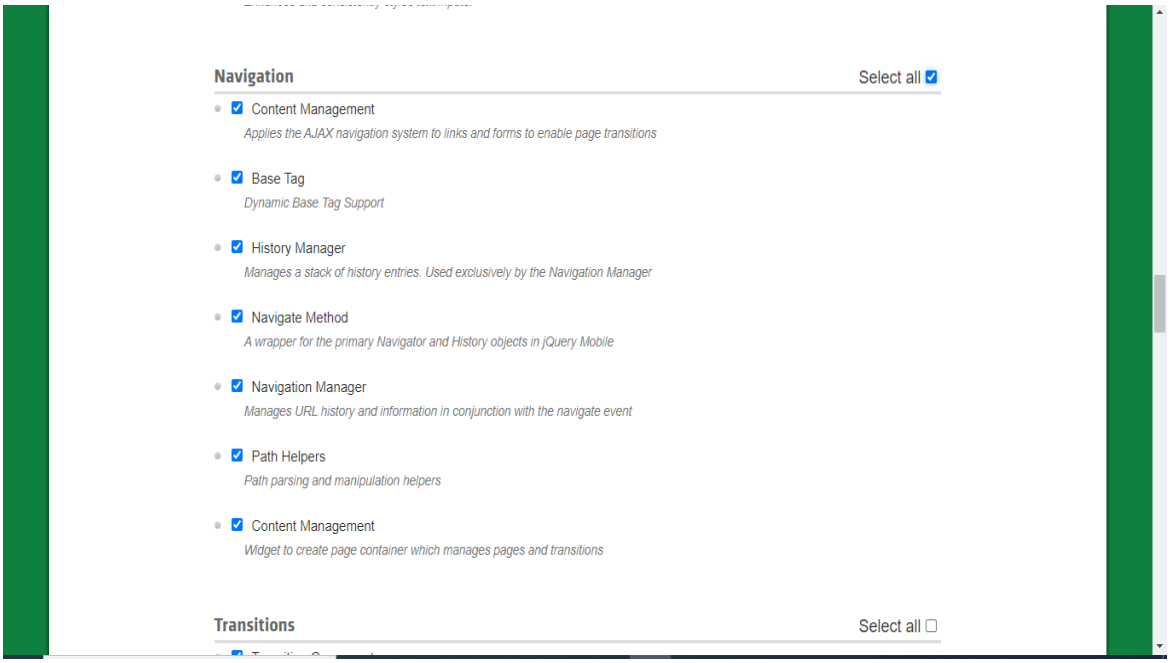
b. Click on "**Select all**". By this we get all events like Navigate, Touch, events, orientation change etc.



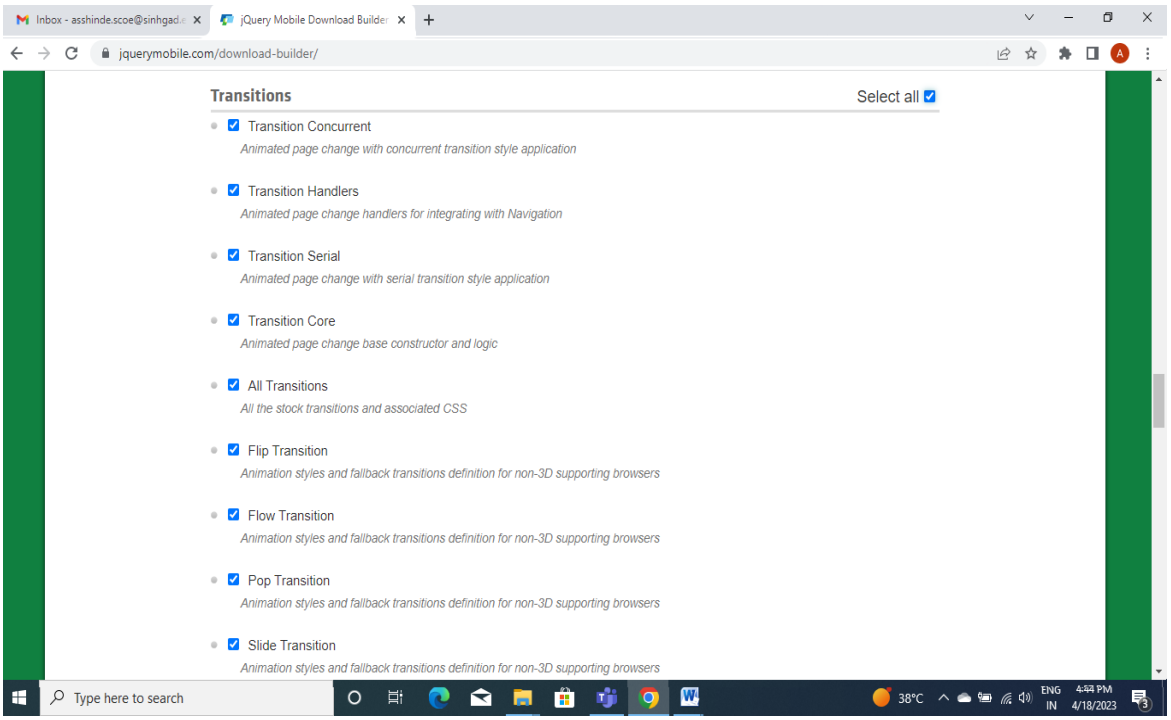
c. Click on “**Select all**”. By this we get all Forms like Buttons: Link-based, Control groups, Range Slider, Form Reset etc.



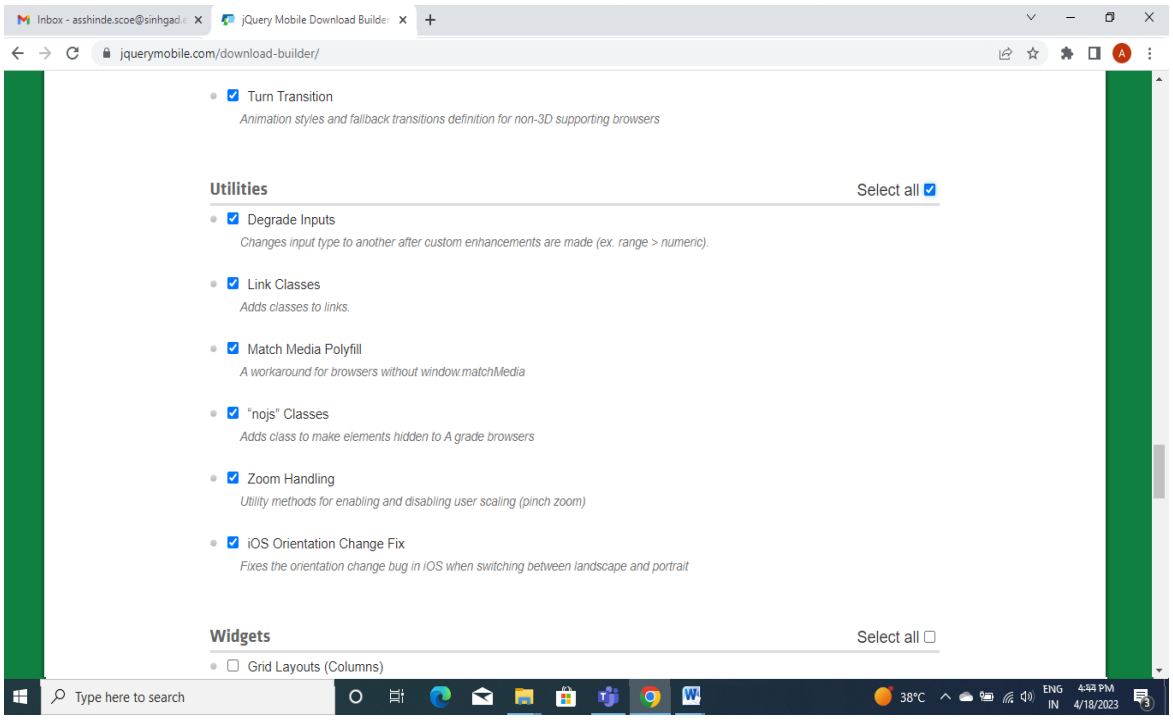
d. Click on “Select all”. By this we get all Navigation like Content Management, Base Tag, History Manager, Path Helpers etc.



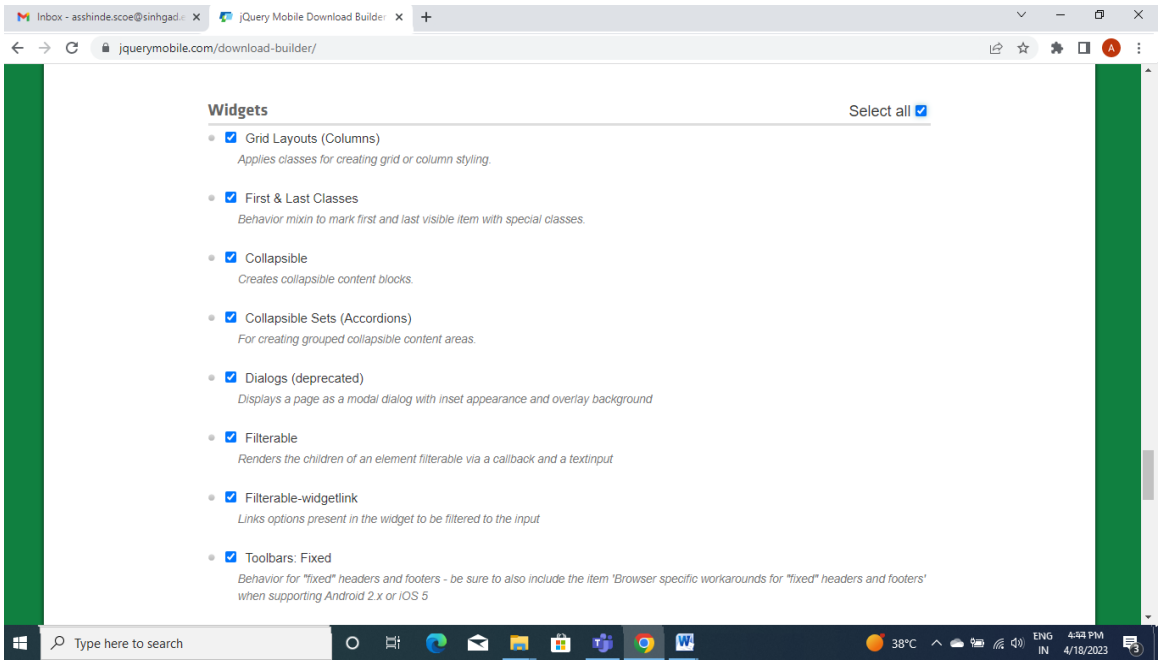
e. Click on "**Select all**". By this we get all Transitions like Transition Concurrent, Transition Handlers, All Transitions, Flip Transition, Management, Base Tag, History Manager, Path Helpers etc.



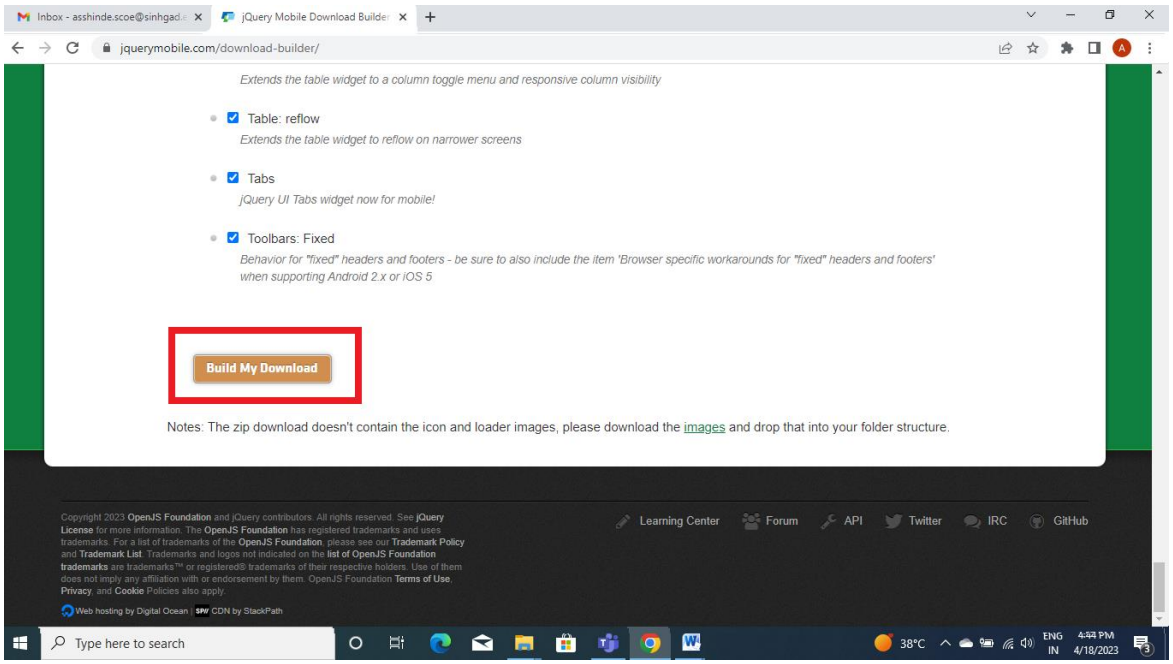
f. Click on "**Select all**". By this we get all Utilities like Degrade Inputs, Link Classes, Zoom Handling, "nojs" Classes, Match Media Polyfill etc.



g. Click on “**Select all**”. By this we get all Widgets like Grid Layouts (Columns), First & Last Classes, Dialogs (deprecated), Filterable-widgetlink, Toolbars: Fixed, List view, Panel, Table etc.



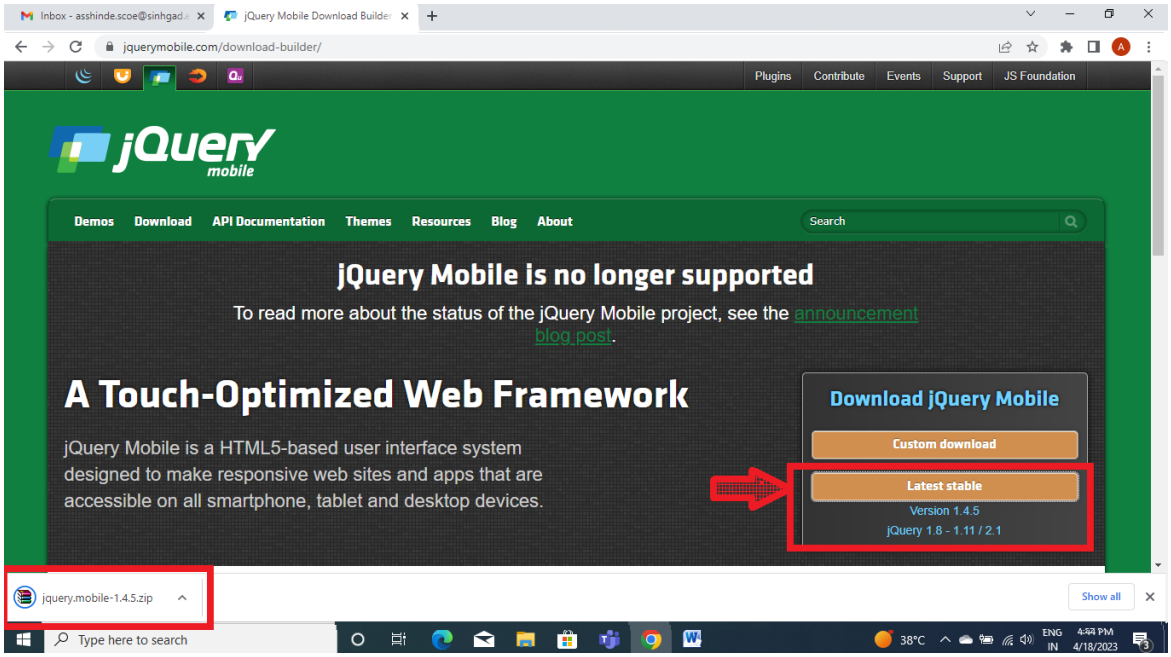
h. You can select the libraries according to your need and click the Build My Download button.



2.2 Stable download:

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:

OUTPUT:

(Attach Screenshots of your output in sequence)

ASSIGNMENT NO. : 04(B)

AIM:

Deploy/Host your web application on AWS VPC or AWS Elastic Beanstalk

PRE-REQUISITE:

1. Knowledge of HTML5 & CSS.
2. Knowledge of JQuery.
3. Knowledge of JavaScript.
4. Knowledge of cloud.

OBJECTIVE:

1. To study Amazon web Services.
2. To study AWS Virtual Private Cloud.
3. Host web application using AWS

THEORY:

What is AWS?

Amazon Web Services:

Amazon Web Services (AWS) is Amazon's cloud web hosting platform that offers flexible, reliable, scalable, easy-to-use, and cost-effective solutions. This tutorial covers various important topics illustrating how AWS works and how it is beneficial to run your website on Amazon Web Services. n 2006, Amazon Web Services (AWS) started to offer IT services to the market in the form of web services, which is nowadays known as cloud computing. With this cloud, we need not plan for servers and other IT infrastructure which takes up much of time in advance. Instead, these services can instantly spin up hundreds or thousands of servers in minutes and deliver results faster. We pay only for what we use with no up-front expenses and no long-term commitments, which makes AWS cost efficient.

Today, AWS provides a highly reliable, scalable, low-cost infrastructure platform in the cloud that powers multitude of businesses in 190 countries around the world.

What is Cloud Computing?

Cloud computing is an internet-based computing service in which large groups of remote servers are networked to allow centralized data storage, and online access to computer services or resources.

Using cloud computing, organizations can use shared computing and storage resources rather than building, operating, and improving infrastructure on their own.

Cloud computing is a model that enables the following features.

- Users can provision and release resources on-demand.
- Resources can be scaled up or down automatically, depending on the load.
- Resources are accessible over a network with proper security.
- Cloud service providers can enable a pay-as-you-go model, where customers are charged based on the type of resources and per usage.

Types of Clouds:

There are three types of clouds – Public, Private, and Hybrid cloud.

Public Cloud	Private Cloud	Hybrid Cloud
In public cloud, the third-party service providers make resources and services available to their customers via Internet. Customer's data and related security is with the service providers' owned infrastructure.	A private cloud also provides almost similar features as public cloud, but the data and services are managed by the organization or by the third party only for the customer's organization. In this type of cloud, major control is over the infrastructure so security related issues are minimized.	A hybrid cloud is the combination of both private and public cloud. The decision to run on private or public cloud usually depends on various parameters like sensitivity of data and applications, industry certifications and required standards, regulations, etc.

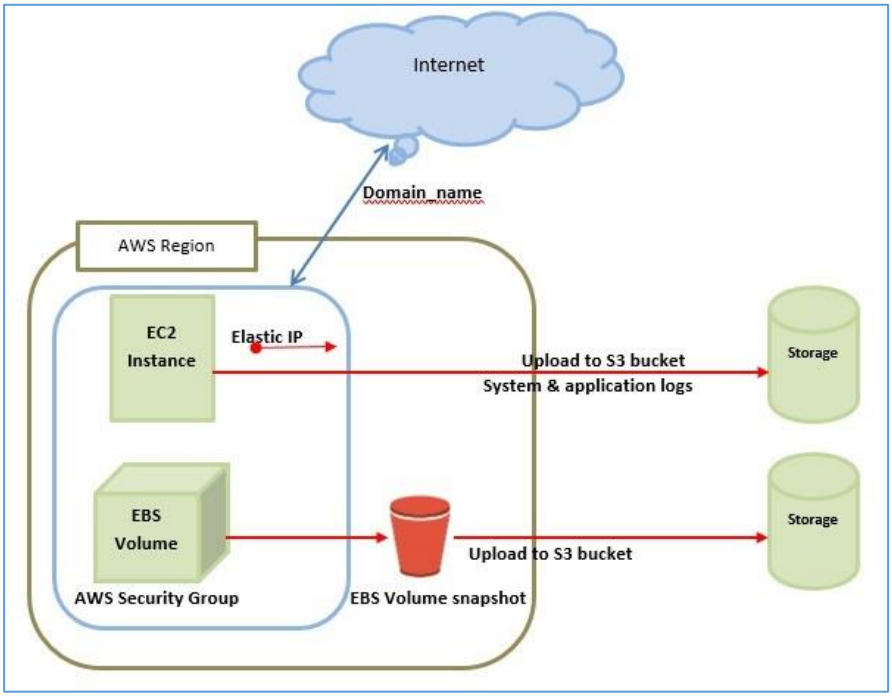
Cloud Service Models

There are three types of service models in cloud – IaaS, PaaS, and SaaS.

IaaS	PaaS	SaaS
IaaS stands for Infrastructure as a Service. It provides users with the capability to provision processing, storage, and network connectivity on demand. Using this service model, the customers can develop their own applications on these resources	PaaS stands for Platform as a Service. Here, the service provider provides various services like databases, queues, workflow engines, e-mails, etc. to their customers. The customer can then use these components for building their own applications. The services, availability of resources and data backup are handled by the service provider that helps the customers to focus more on their application's functionality.	SaaS stands for Software as a Service. As the name suggests, here the third-party providers provide end-user applications to their customers with some administrative capability at the application level, such as the ability to create and manage their users. Also some level of customizability is possible such as the customers can use their own corporate logos, colors, etc.

Amazon Web Services - Basic Architecture:

This is the basic structure of AWS EC2, where EC2 stands for Elastic Compute Cloud. EC2 allow users to use virtual machines of different configurations as per their requirement. It allows various configuration options, mapping of individual server, various pricing options, etc. We will discuss these in detail in AWS Products section. Following is the diagrammatic representation of the architecture.



Amazon Web Services - Virtual Private Cloud:

Amazon Virtual Private Cloud (VPC) allows the users to use AWS resources in a virtual network. The users can customize their virtual networking environment as they like, such as selecting own IP address range, creating subnets, and configuring route tables and network gateways.

The list of AWS services that can be used with Amazon VPC are −

- Amazon EC2
- Amazon Route 53
- Amazon WorkSpaces
- Auto Scaling
- Elastic Load Balancing
- AWS Data Pipeline
- Elastic Beanstalk
- Amazon Elastic Cache
- Amazon EMR
- Amazon OpsWorks
- Amazon RDS
- Amazon Redshift

How to Use Amazon VPC?

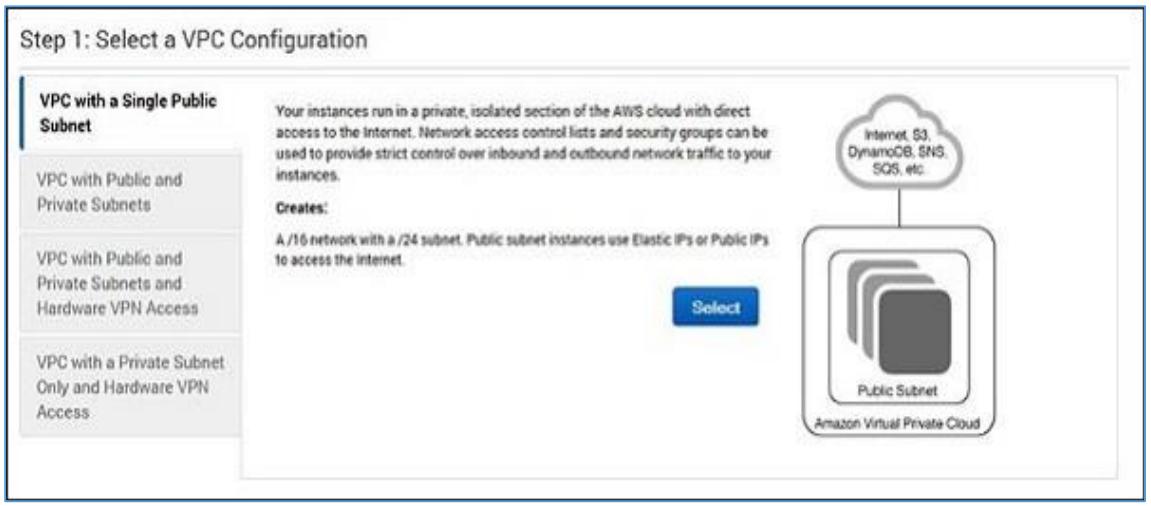
Following are the steps to create VPC.

A. Create VPC:

Step 1: Open the Amazon VPC console by using the following link https://console.aws.amazon.com/vpc/

Step 2: Select creating the VPC option on the right side of the navigation bar. Make sure that the same region is selected as for other services.

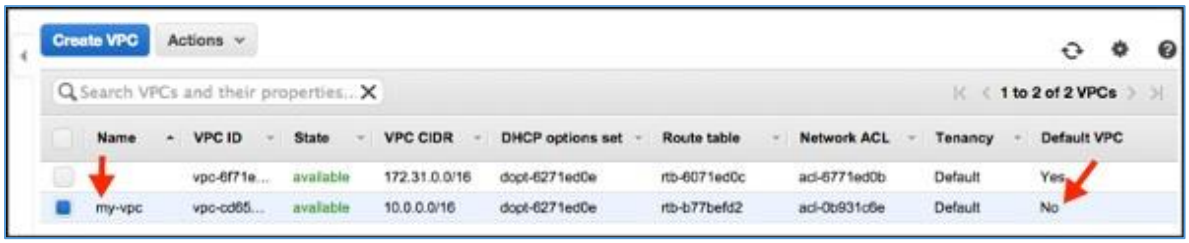
Step 3: Click the start VPC wizard option, then click VPC with single public subnet option on the left side.



Step 4: A configuration page will open. Fill in the details like VPC name, subnet name and leave the other fields as default. Click the Create VPC button.

Step 5: A dialog box will open, showing the work in progress. When it is completed, select the OK button.

The Your VPCs page opens which shows a list of available VPCs. The setting of VPC can be changed here.

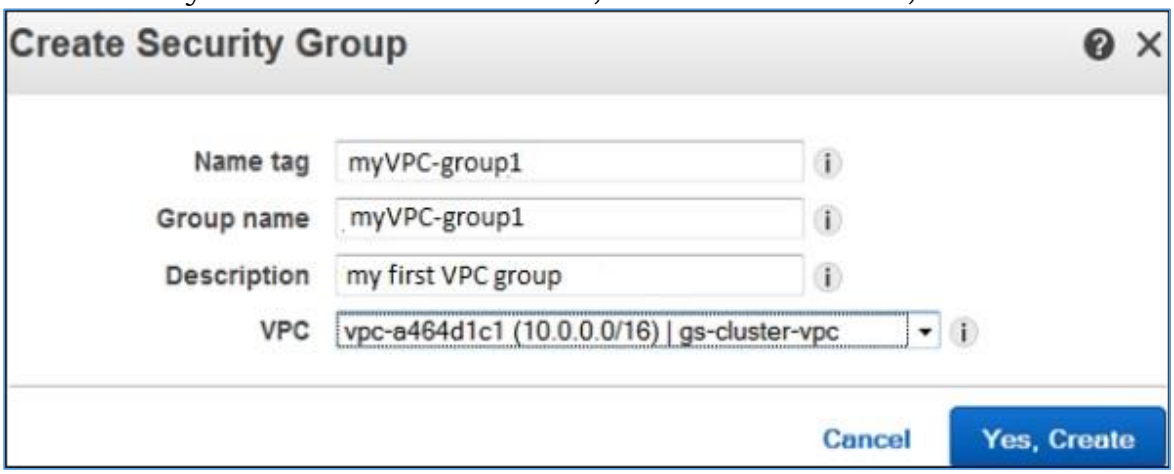


B. Select/Create VPC Group:

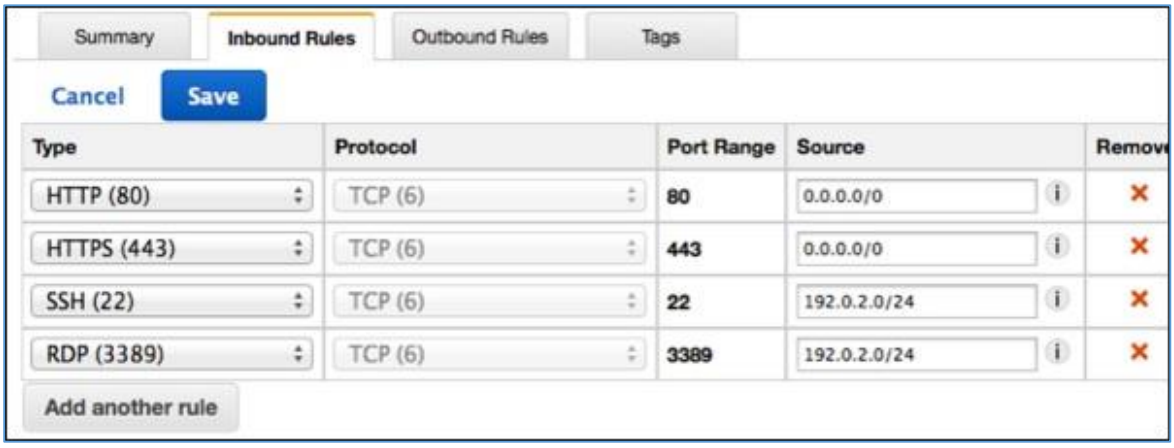
Step 1: Open the Amazon VPC console by using the following link https://console.aws.amazon.com/vpc/

Step 2: Select the security groups option in the navigation bar, then choose create security group option.

Step 3: A form will open, enter the details like group name, name tag, etc. Select ID of your VPC from VPC menu, then select the Yes, create button.



Step 4: The list of groups opens. Select the group name from the list and set rules. Then click the Save button.



C. Launch Instance into VPC:

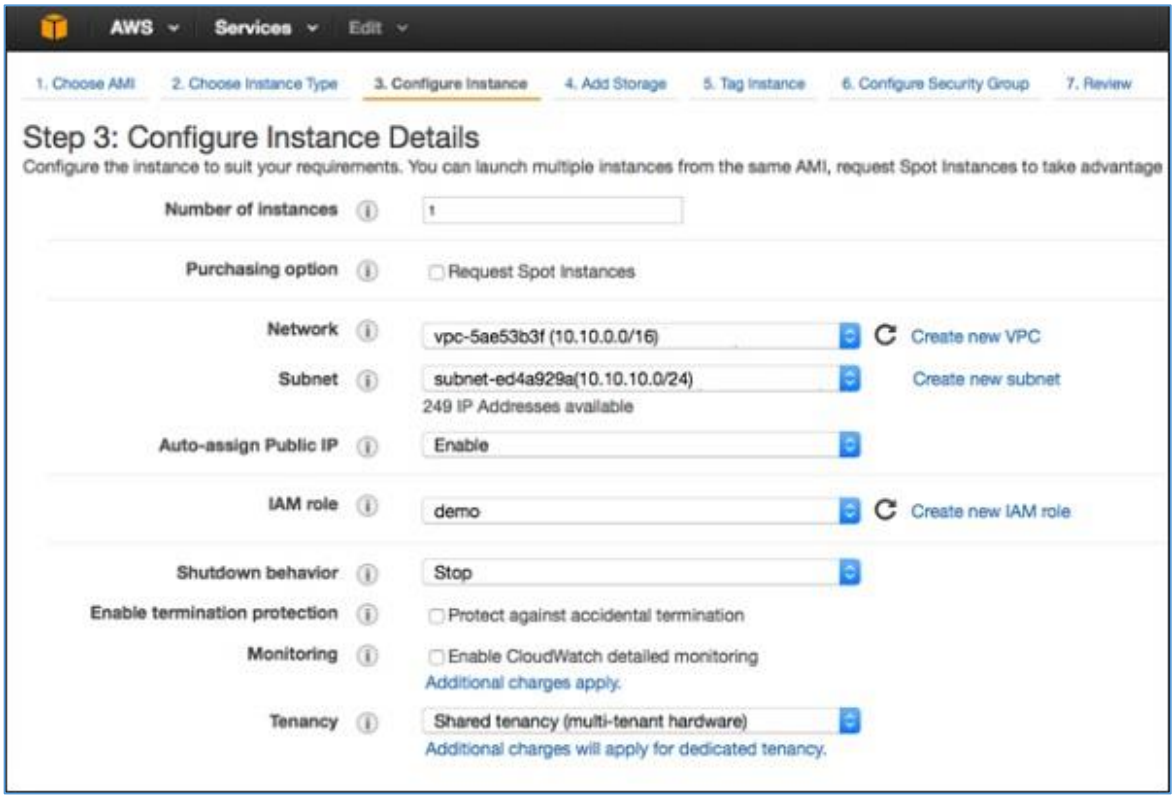
Step 1: Open the Amazon VPC console using the following link https://console.aws.amazon.com/vpc/

Step 2: Select the same region as while creating VPC and security group.

Step 3: Now select the Launch Instance option in the navigation bar.

Step 4: A page opens. Choose the AMI which is to be used.

Step 5: A new page opens. Choose an Instance Type and select the hardware configuration. Then select Next: Configure Instance Details.



Step 6: Select the recently created VPC from the Network list, and the subnet from the Subnet list. Leave the other settings as default and click Next till the Tag Instance page.

Step 7: On the Tag Instance page, tag the instance with the Name tag. This helps to identify your instance from the list of multiple instances. Click Next: Configure Security Group.

Step 8: On the Configure Security Group page, select the recently created group from the list. Then, select Review and Launch button.

Step 9: On the Review Instance Launch page, check your instance details, then select Launch.

Step 10: A dialog box appears. Choose the option Select an existing key pair or create a new key pair, then click the Launch Instances button.

Step 11: The confirmation page open which shows all the details related to instances.

D. Assign Elastic IP Address to VPC Instances:

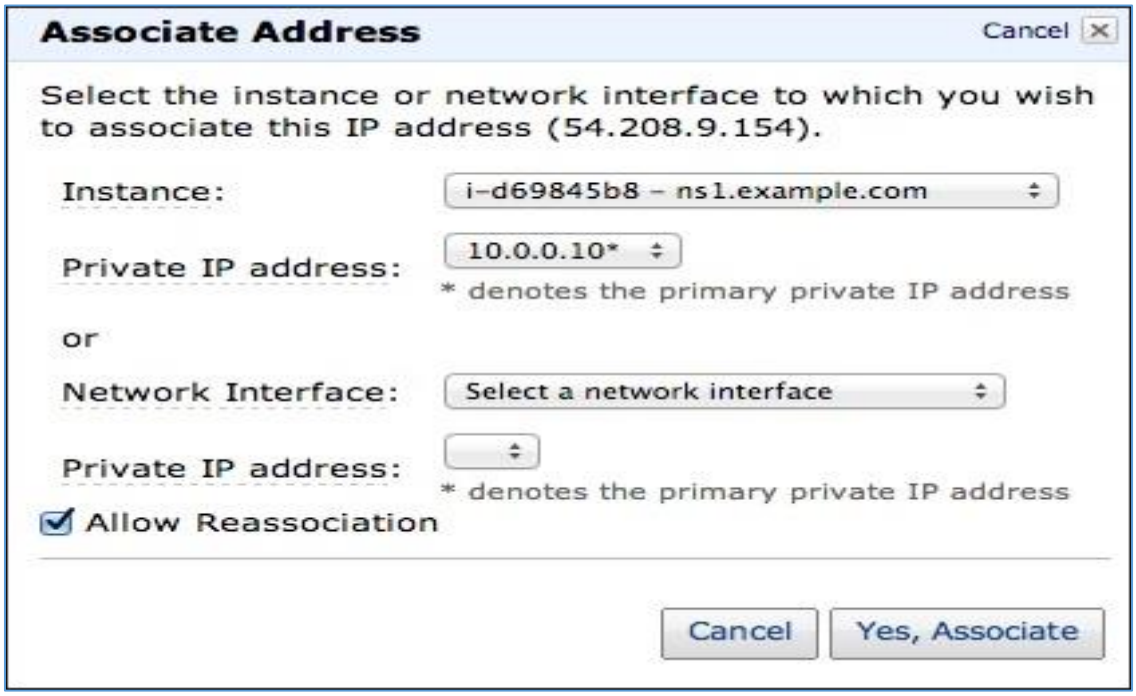
Step 1: Open the Amazon VPC console using the following link https://console.aws.amazon.com/vpc/

Step 2: Select Elastic IP's option in the navigation bar.

Step 3: Select Allocate New Address. Then select Yes, Allocate button.

Step 4: Select your Elastic IP address from the list, then select Actions, and then click the Associate Address button.

Step 5: A dialog box will open. First select the Instance from the Associate with list. Then select your instance from the Instance list. Finally click the Yes, Associate button.



E. Delete a VPC

There are several steps to delete VPC without losing any resources associated with it. Following are the steps to delete a VPC.

Step 1: Open the Amazon VPC console using the following link https://console.aws.amazon.com/vpc/

Step 2: Select Instances option in the navigation bar.

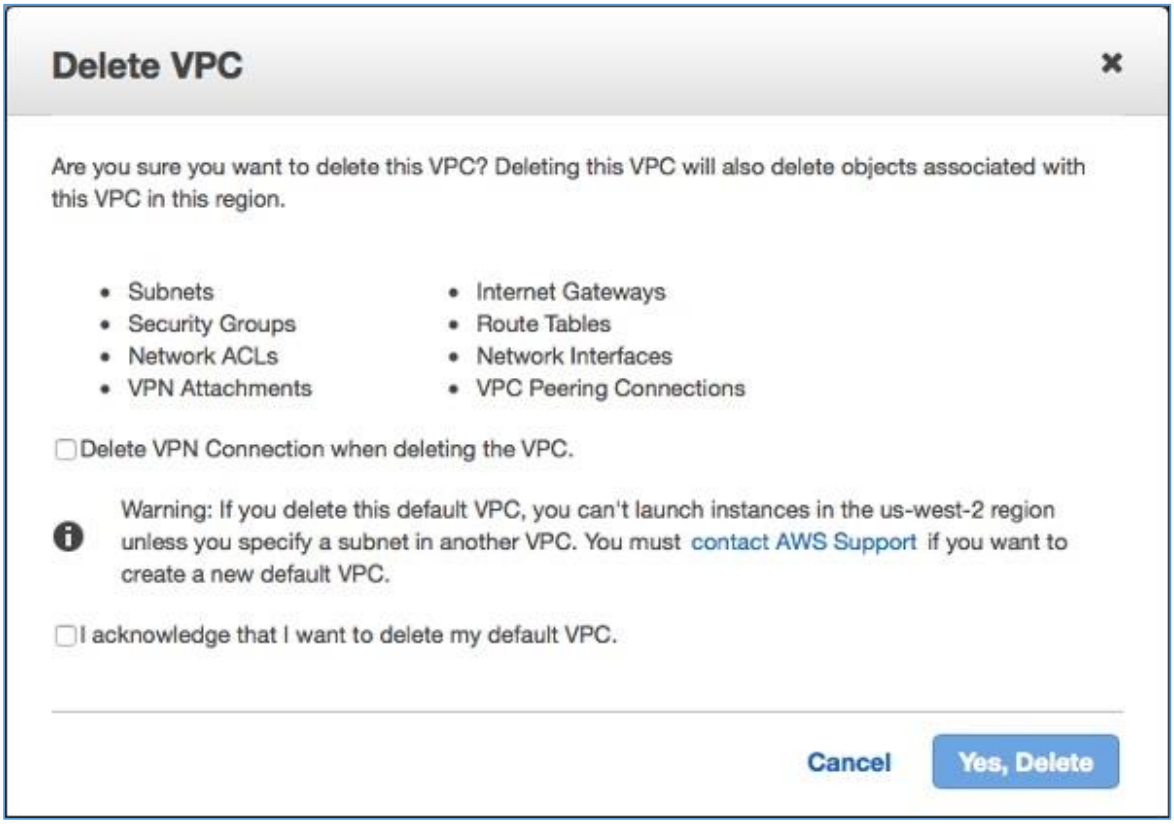
Step 3: Select the Instance from the list, then select the Actions → Instance State → Terminate button.

Step 4: A new dialog box opens. Expand the Release attached Elastic IPs section, and select the checkbox next to the Elastic IP address. Click the Yes, Terminate button.

Step 5: Again open the Amazon VPC console using the following link https://console.aws.amazon.com/vpc/

Step 6: Select the VPC from the navigation bar. Then select Actions & finally click the Delete VPC button.

Step 7: A confirmation message appears. Click the Yes, Delete button.



CONCLUSION:

_____

_____

_____

_____

_____

OUTPUT:

(Attach Screenshots of your output in sequence)