

WEEK - 6

1. Implement navigation using react router.

Components in React Router:

There are two types of router components:

- **<BrowserRouter>**: It is used for handling the dynamic URL.
- **<HashRouter>**: It is used for handling the static request.

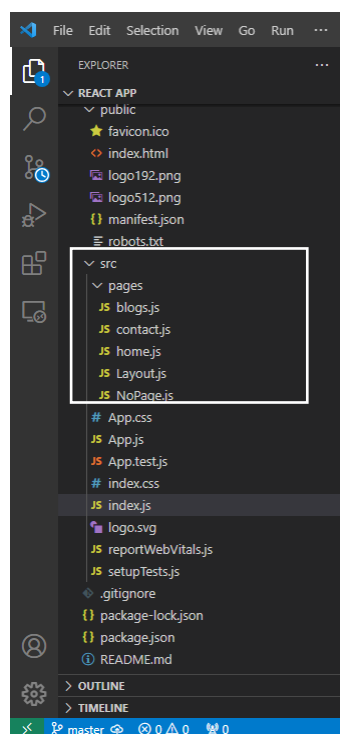
Step 1: Open the VS Code and launch the React on the local server.

• Add React Router:

To add React Router in your application, run this command '**npm i -D react-router-dom**' in the terminal from the root directory of the application. (Go to Terminal ☐ New Terminal ☐ Command Prompt).

Step 2: Open the folder in the VS Code and in the '**src**' folder, create a new folder named '**Pages**'.

Step 3: Now within the '**Pages**' folder, create the new files named '**blogs.js**', '**contact.js**', '**home.js**', '**Layout.js**' & '**NoPage.js**' as shown below.



Step 4: Then add code inside the '**index.js**' file as shown here.

```
JS index.js
router > src > JS index.js > ...
1 import ReactDOM from "react-dom/client";
2 import { BrowserRouter, Routes, Route } from "react-router-dom";
3 import Layout from "../pages/Layout";
4 import Home from "../pages/home";
5 import Blogs from "../pages/blogs";
6 import Contact from "../pages/contact";
7 import NoPage from "../pages/NoPage";
8 export default function App() {
9   return (
10     <BrowserRouter>
11       <Routes>
12         <Route path="/" element={<Layout />} />
13         <Route index element={<Home />} />
14         <Route path="blogs" element={<Blogs />} />
15         <Route path="contact" element={<Contact />} />
16         <Route path="*" element={<NoPage />} />
17       </Routes>
18     </BrowserRouter>
19   );
20 }
21
22 const root = ReactDOM.createRoot(document.getElementById('root'));
23 root.render(<App />);
24
```

Step 5: Now add code to ‘App.css’ file to style your application.

```
JS index.js # App.css X
router > src > # App.css > li a: hover: not(.active)
1 ul {
2   list-style-type: none;
3   margin: 0;
4   padding: 0;
5   overflow: hidden;
6   background-color: #04AA6D;
7 }
8
9 li {
10  float: left;
11  border-right: 1px solid #bbb;
12 }
13
14 li a {
15  display: block;
16  color: white;
17  text-align: center;
18  padding: 14px 16px;
19  text-decoration: none;
20 }
21
22 li a: hover: not(.active) {
23  background-color: #111;
24 }
25
```

Step 6: Write the below code to ‘blogs.js’ file inside the ‘Pages’ folder.

```
JS index.js JS blogs.js X
router > src > pages > JS blogs.js > default
1 const Blogs = () => {
2   return <h1>Blog Articles</h1>
3 };
4 export default Blogs;
```

Step 7: Write the below code to ‘contact.js’ file inside the ‘Pages’ folder.

```
JS index.js JS contact.js X
router > src > pages > JS contact.js > default
1 const contact = () => {
2   return <h1>contact me</h1>
3 };
4 export default contact;
```

Step 8: Write the below code to **'home.js'** file inside the **'Pages'** folder.

```
JS index.js • JS home.js X
router > src > pages > JS home.js > [0] default
1  const Home = () => {
2    |   return <h1>Home</h1>
3    | };
4  export default Home;
```

Step 9: Write the below code to **'Layout.js'** file inside the **'Pages'** folder.

```
JS index.js • JS Layout.js X
router > src > pages > JS Layout.js > [0] default
1  import { Outlet, Link } from "react-router-dom";
2  const Layout = () => {
3    |   return (
4    |     <div>
5    |       <nav>
6    |         <ul>
7    |           <li>
8    |             <Link to="/">Home</Link>
9    |           </li>
10          <li>
11            <Link to="/blogs">Blogs</Link>
12          </li>
13          <li>
14            <Link to="/contact">contact</Link>
15          </li>
16        </ul>
17      </nav>
18      <Outlet />
19    </div>
20    | );
21    | };
22  export default Layout;
```

Step 10: Write the below code to **'NoPages.js'** file inside the **'Pages'** folder.

```
JS index.js • JS NoPage.js X
router > src > pages > JS NoPage.js > [0] default
1  const NoPage = () => {
2    |   return <h1>404</h1>;
3    | };
4  export default NoPage;
```

Now run the **'index.js'** file to view the final output.

Output:





2. React Hooks-demonstrating useState, useEffect, useContext, useReducer.

React Hooks: Hooks (Receive or Steal) are the new feature introduced in the React 16.8 version. It allows you to use state and other React features without writing a class.

- **Types of Hooks in React:**

- 1) **useState** - The React useState Hook allows us to track state in a function component. State generally refers to data or properties that need to be tracking in an application. To use the useState Hook, we first need to import it into our component.

Syntax: import { useState } from "react";

- 2) **useEffect** - The useEffect Hook allows you to perform side effects in your components. Some examples of side effects are: fetching data, directly updating the DOM, and timers. useEffect accepts two arguments. The second argument is optional.

Syntax: useEffect(<function>, <dependency>)

- 3) **useContext** - React Context is a way to manage state globally. It can be used together with the useState Hook to share state between deeply nested components more easily than with useState alone. To create context, you must Import createContext and initialize it.

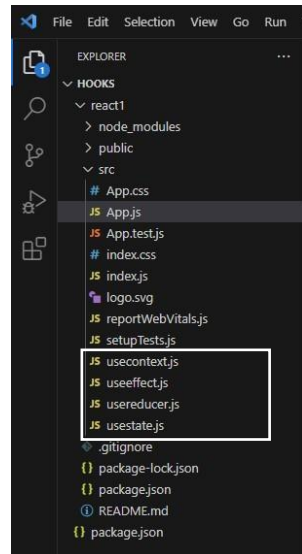
Syntax: const myValue = useContext(MyContext);

- 4) **useReducer** - The useReducer Hook is similar to the useState Hook. It allows for custom state logic. The useReducer Hook accepts two arguments.

Syntax: useReducer(<reducer>, <initialState>)

Step 1: Open the VS Code and launch the React on the local server.

Step 2: To run the React Hooks, create the files (**useState.js**, **useeffect.js**, **usecontext.js**, **usereducer.js**) as shown below in the ‘src’ folder.



- **Example of useState:**

Step 3: Write the below code in the ‘App.js’ file inside the ‘src’ folder.

```
react1 > src > JS App.js > ...
1  import Name from './useState';
2
3  function App() {
4    return (
5      <div>
6
7        <Name/>
8      </div>
9    );
10 }
11
12 export default App;
13
14 |
```

Step 4: Then add code in ‘useState.js’ file within the ‘src’ folder.

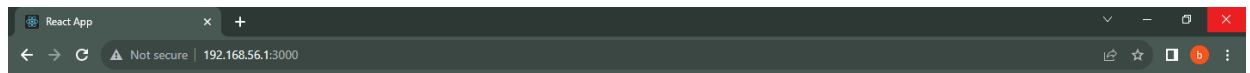
```
react1 > src > JS usestate.js > ...
1  import { useState } from "react";
2
3  function Name() {
4    const [name, setName] = useState("VVP_CS_FSD");
5    const changeName = () => {
6      setName("Full stack development");
7    };
8
9    return (
10     <div>
11       <p>My name is {name}</p>
12       <button onClick={changeName}> Click me </button>
13     </div>
14   );
15 }
16
17 export default Name;
18
19
```

Output:



My name is VVP_CS_FSD

Click me



My name is Full stack development

Click me

- **Example of UseEffect:**

Step 5: Write the below code in the 'App.js' file inside the 'src' folder.

```
react1 > src > JS App.js > ...
1  import CounterExample from './useeffect';
2
3  function App() {
4    return (
5      <div>
6
7        <CounterExample/>
8      </div>
9    );
10 }
11
12 export default App;
13
```

Step 6: Then add code in 'useeffect.js' file within the 'src' folder.

```
react1 > src > JS useeffect.js > ...
1  import React, { useState, useEffect } from 'react';
2
3  function CounterExample() {
4    const [count, setCount] = useState(0);
5
6    // Similar to componentDidMount and componentDidUpdate:
7    useEffect(() => {
8      // Update the document title using the browser API
9      document.title = `You clicked ${count} times`;
10    });
11
12    return (
13      <div>
14        <p>You clicked {count} times</p>
15        <button onClick={() => setCount(count + 1)}>
16          Click me
17        </button>
18      </div>
19    );
20  }
21  export default CounterExample;
22
```

Output:



You clicked 0 times

Click me



You clicked 6 times

Click me

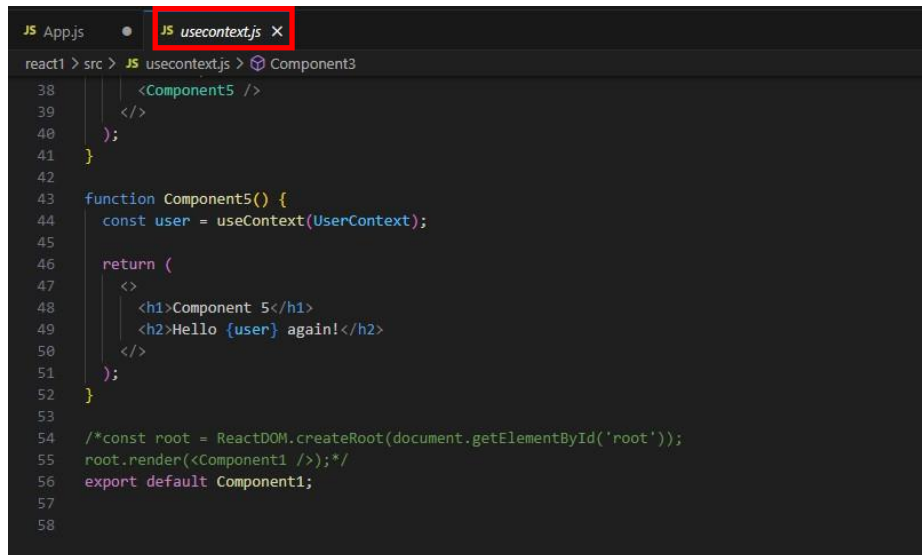
- **Example of useContext:**

Step 7: Write the below code in the 'App.js' file inside the 'src' folder.

```
react1 > src > JS App.js > ...
1  import Component1 from './usecontext';
2
3  function App() {
4    return (
5      <div>
6        <Component1/>
7      </div>
8    );
9  }
10
11 export default App
12
```

Step 8: Then add code in the 'usecontext.js' file within the 'src' folder.

```
react1 > src > JS usecontext.js > Component3
1  import { useState, createContext, useContext } from "react";
2
3  const UserContext = createContext();
4
5  function Component1() {
6    const [user, setUser] = useState("FSD");
7
8    return (
9      <UserContext.Provider value={user}>
10        <h1>Hello {user}!</h1>
11        <Component2 />
12      </UserContext.Provider>
13    );
14  }
15
16  function Component2() {
17    return (
18      <>
19        <h1>Component 2</h1>
20        <Component3 />
21      </>
22    );
23  }
24
25  function Component3() {
26    return (
27      <>
28        <h1>Component 3</h1>
29        <Component4 />
30      </>
31    );
32  }
33
34  function Component4() {
35    return (
36      <>
37        <h1>Component 4</h1>
38        <Component5 />
39      </>
40    );
41  }
42
```

A screenshot of a VS Code editor window. The top bar shows two tabs: 'JS App.js' and 'JS usecontext.js', with the latter being the active tab and highlighted with a red box. The breadcrumb navigation at the top reads 'react1 > src > JS usecontext.js > Component3'. The code in the editor is as follows:

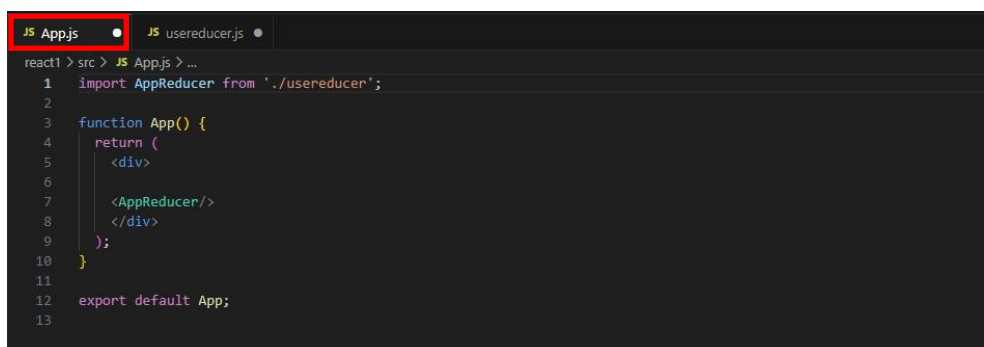
```
38 |         <Component5 />
39 |       </>
40 |     );
41 |   }
42 |
43 |   function Component5() {
44 |     const user = useContext(UserContext);
45 |
46 |     return (
47 |       <>
48 |         <h1>Component 5</h1>
49 |         <h2>Hello {user} again!</h2>
50 |       </>
51 |     );
52 |   }
53 |
54 |   /*const root = ReactDOM.createRoot(document.getElementById('root'));
55 |   root.render(<Component1 />);*/
56 |   export default Component1;
57 |
58 |
```

Output:



• Example of UseReducer:

Step 9: Write the below code in the 'App.js' file inside the 'src' folder.

A screenshot of a VS Code editor window. The top bar shows two tabs: 'JS App.js' and 'JS usereducer.js', with the former being the active tab and highlighted with a red box. The breadcrumb navigation at the top reads 'react1 > src > JS App.js > ...'. The code in the editor is as follows:

```
1 | import AppReducer from './usereducer';
2 |
3 | function App() {
4 |   return (
5 |     <div>
6 |
7 |       <AppReducer/>
8 |     </div>
9 |   );
10 | }
11 |
12 | export default App;
13 |
```

Step 10: Then add code in the 'usereducer.js' file within the 'src' folder.


```
JS App.js • JS usereducer.js •
react1 > src > JS usereducer.js > ...
1  import React, { useReducer } from "react";
2
3  // Defining the initial state and the reducer
4  const initialState = 0;
5  const reducer = (state, action) => {
6    switch (action) {
7      case "add":
8        return state + 1;
9      case "subtract":
10       return state - 1;
11      case "reset":
12        return 0;
13      default:
14        throw new Error("Unexpected action");
15    }
16  };
17
18  const AppReducer = () => {
19    // Initialising useReducer hook
20    const [count, dispatch] = useReducer(reducer, initialState);
21    return (
22      <div>
23        <h2>{count}</h2>
24        <button onClick={() => dispatch("add")}>
25          add
26        </button>
27        <button onClick={() => dispatch("subtract")}>
28          subtract
29        </button>
30        <button onClick={() => dispatch("reset")}>
31          reset
32        </button>
33      </div>
34    );
35  };
36  export default AppReducer;
37
```

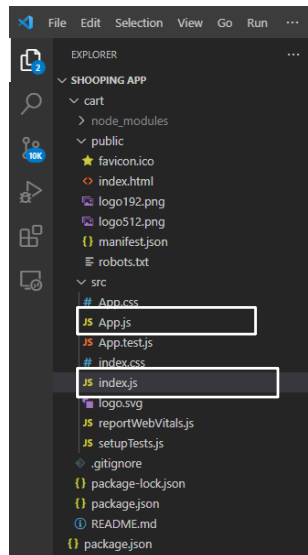
Output:



3. Build single page application - Shopping cart.

Step 1: Open the VS Code and launch the React on the local server.

- React folder structure:



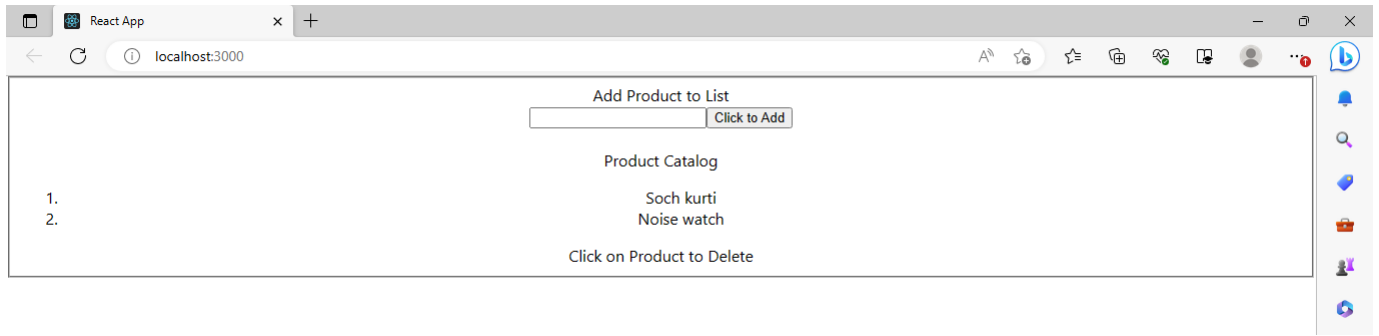
Step 2: Add code to 'index.js' file in the 'src' folder.

```
JS App.js • JS index.js •
cart > src > JS index.js > ...
1  import React from 'react';
2  import ReactDOM from 'react-dom/client';
3  import './index.css';
4  import App from './App';
5  import reportWebVitals from './reportWebVitals';
6  const root = ReactDOM.createRoot(document.getElementById('root'));
7  root.render(
8    <React.StrictMode>
9      <App/>
10   </React.StrictMode>
11 );
12
13
```

Step 3: Then write the below code in the 'App.js' within the 'src' folder.

```
JS App.js • JS index.js
cart > src > JS App.js > ...
1  import { useState } from "react";
2  function App() {
3    const [list, setList] = useState([]);
4    const [value, setValue] = useState("");
5    const addToList = () => {
6      let tempArr = list;
7      tempArr.push(value);
8      setList(tempArr);
9      setValue("");
10   };
11   const deleteItem = (index) => {
12     let temp = list.filter((item, i) => i !== index);
13     setList(temp);
14   };
15   return (
16     <div className="App">
17       <fieldset>
18         <h>Add Product to List</h><br></br>
19         <input type="text" value={value} onChange={(e) => setValue(e.target.value)} />
20         <button onClick={addToList}> Click to Add </button><br></br></br>
21         <h>Product Catalog</h><br></br>
22         <ol>
23           {list.map((item, i) => <li onClick={() => deleteItem(i)}>{item} </li>)}
24         </ol>
25         <h>Click on Product to Delete</h><br></br>
26       </fieldset></div>
27     );
28   } export default App;
```

Output:

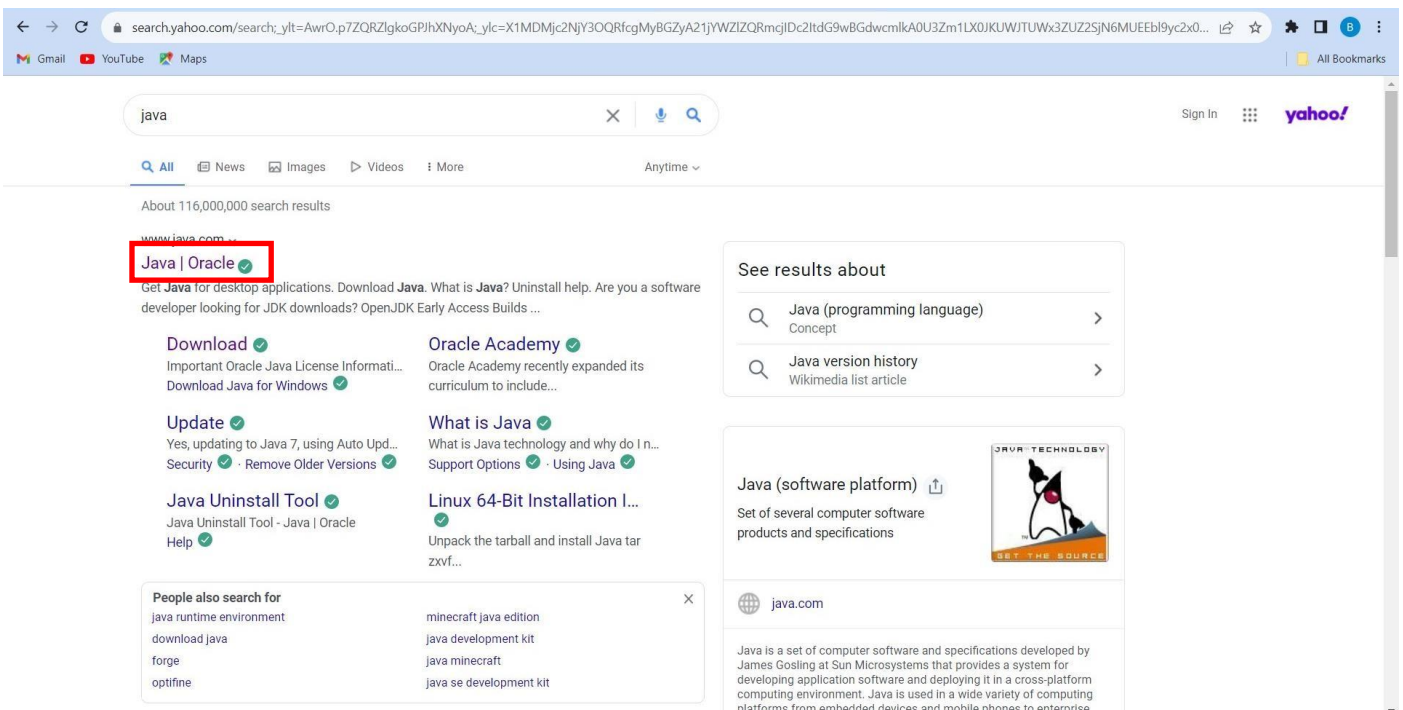


4. Setting up the environment and tools to install Java (latest stable version) and add environment variable to it.

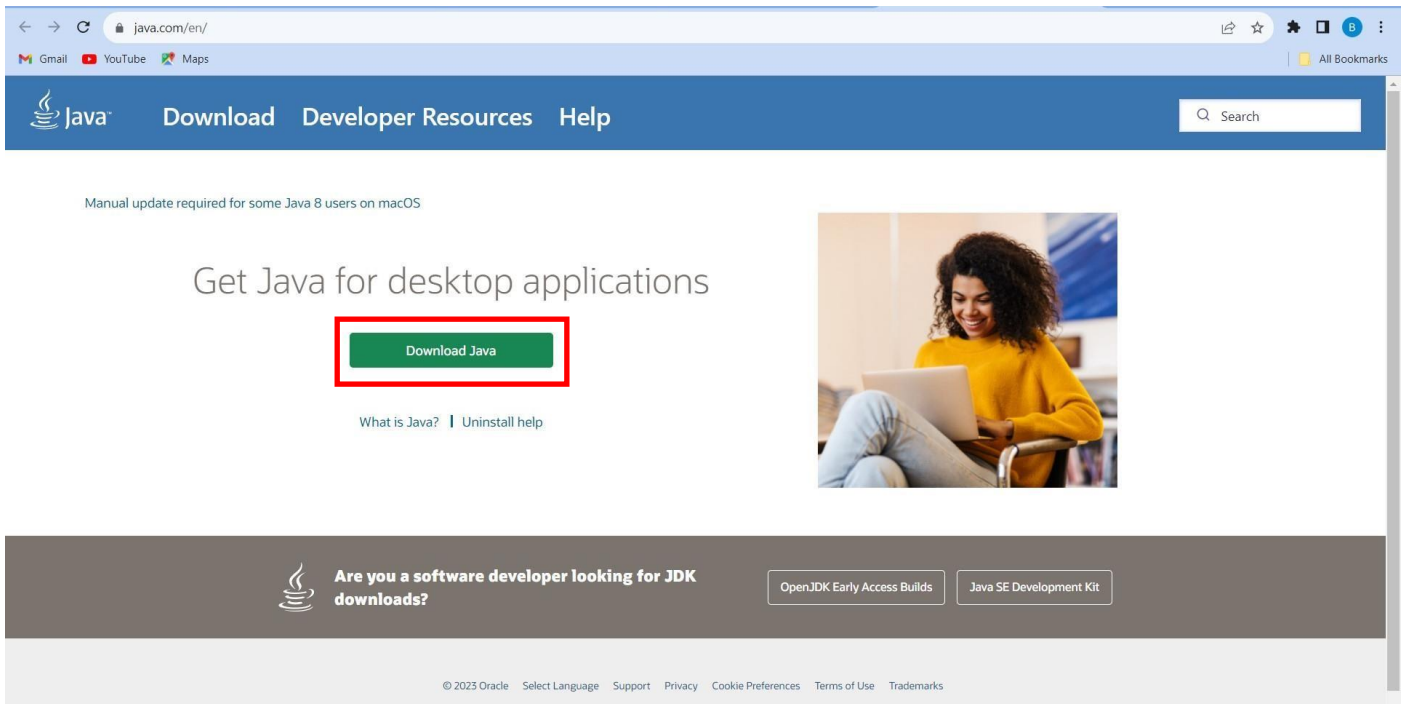
JAVA: Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers *write once, run anywhere* (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to recompile.

Installation of Java on Windows:

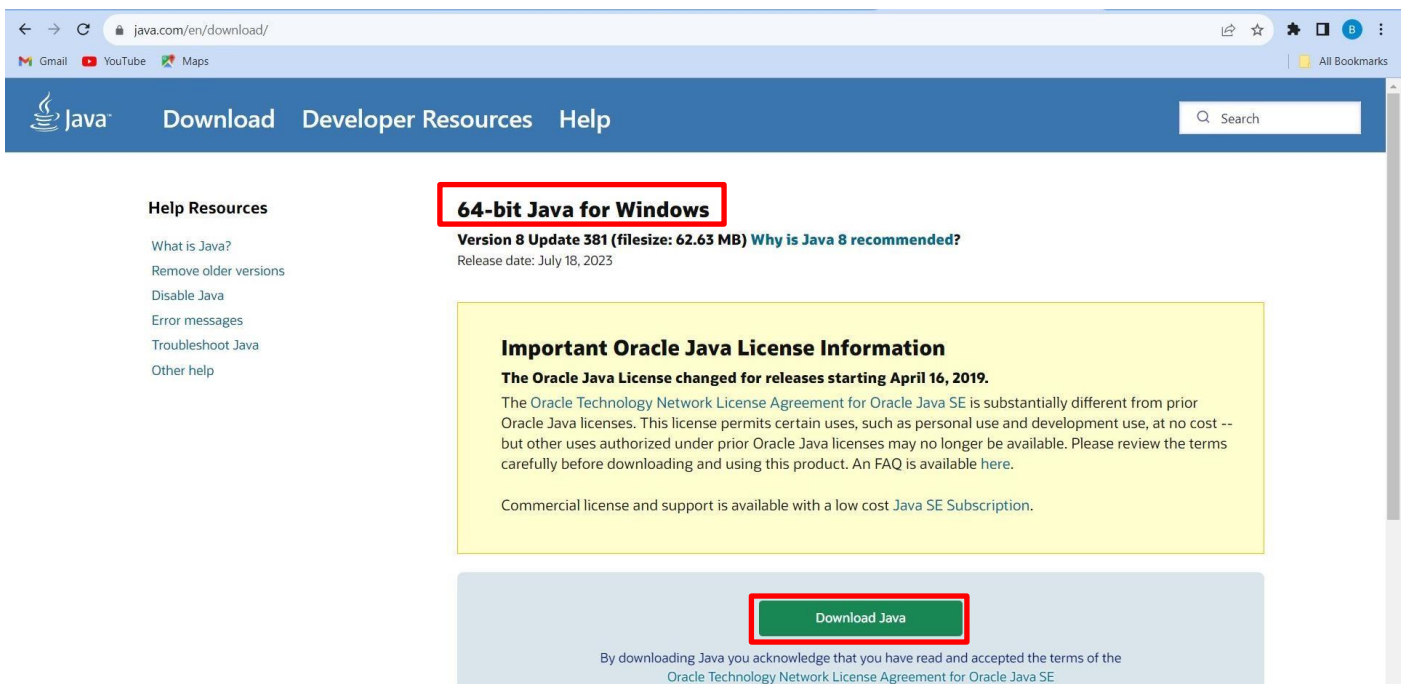
Step 1: Go to Web Browser → Search for Java → Click on the first link here.



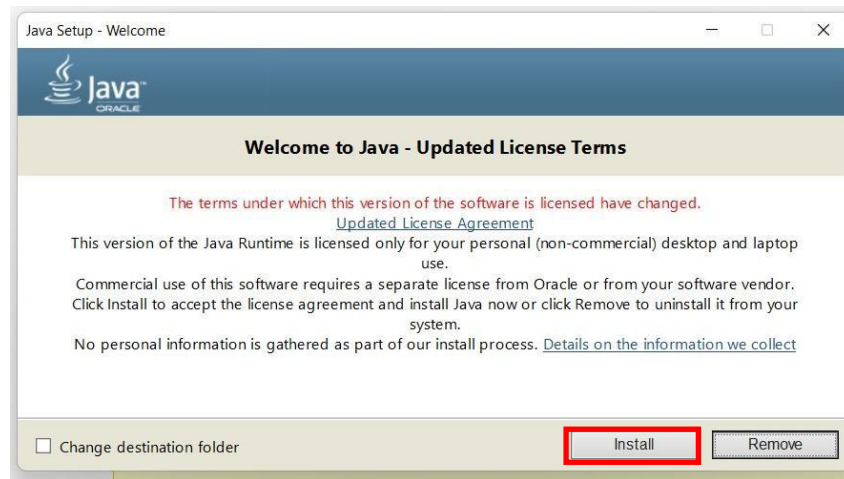
Step 2: Now click on 'Download Java'.



Step 3: Make sure to download the stable version for Windows. Here **64-bit** system type is recommended for download. Now click on ‘**Download Java**’.



Step 4: Open the downloaded file & click on ‘**Install**’.



Step 5: Wait until the installation is completed



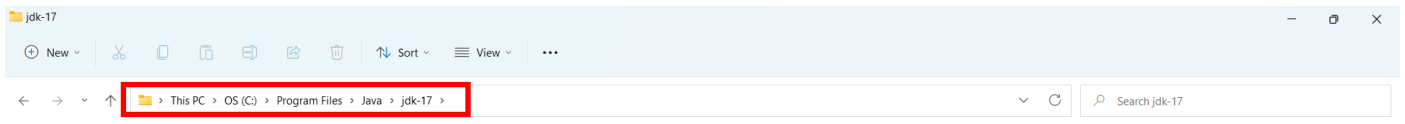
Step 6: Click on 'close'.



- **Setting up of Environment Variables for Java.**

Note: To run Java program, make sure to setup the Environment variables for Java on your windows system.

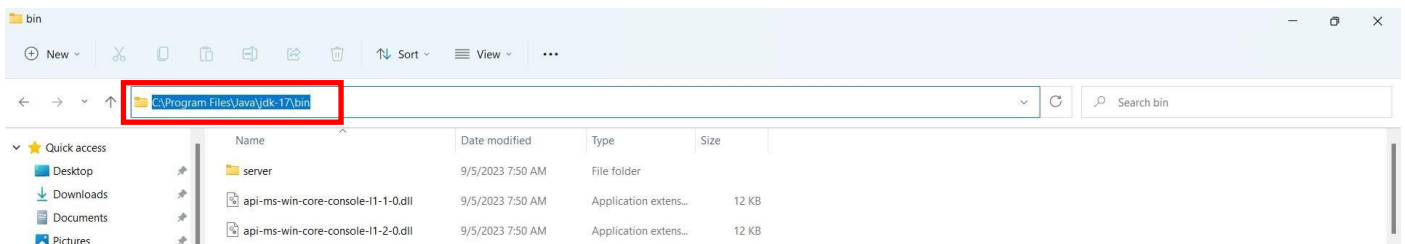
Step 1: After the installation of Java, go to This PC → C drive → Program files → Java → jdk17 as shown below.



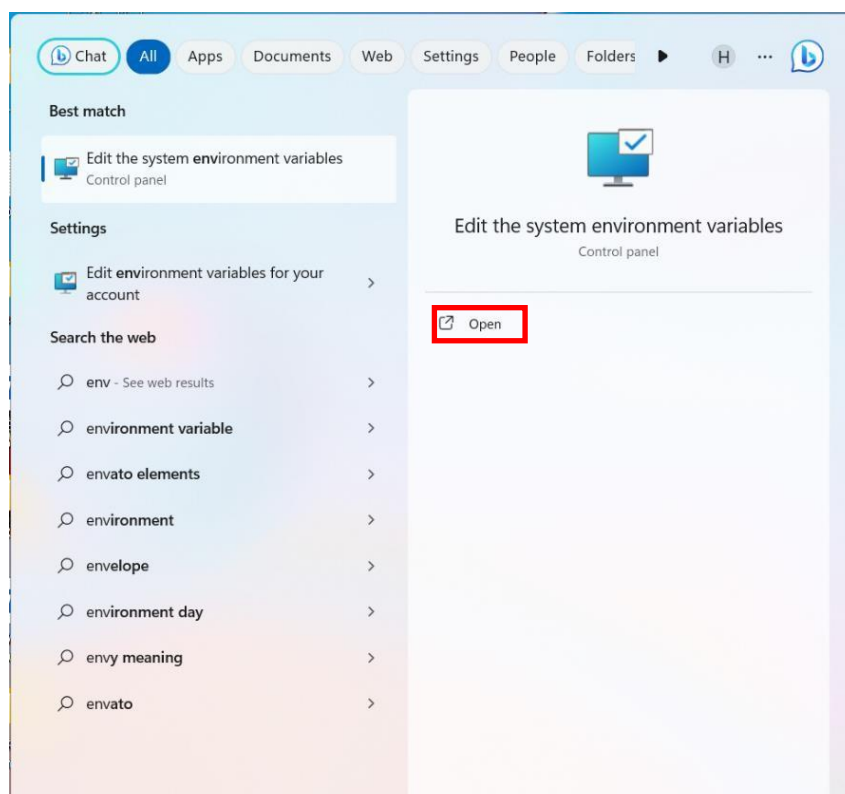
Step 2: Now click on 'bin' folder.

Name	Date modified	Type	Size
bin	9/5/2023 7:50 AM	File folder	
conf	9/5/2023 7:50 AM	File folder	
include	9/5/2023 7:50 AM	File folder	
jmods	9/5/2023 7:50 AM	File folder	
legal	9/5/2023 7:50 AM	File folder	
lib	9/5/2023 7:50 AM	File folder	
jenkins.err	9/29/2023 8:26 AM	Text Document	169 KB
jenkins	8/23/2023 7:45 AM	Application	606 KB
jenkins.exe	8/23/2023 1:34 PM	Configuration Sou...	1 KB
jenkins.out	9/6/2023 9:41 PM	Text Document	1 KB
Jenkins.war	8/23/2023 1:26 PM	WAR File	87,435 KB
jenkins.wrapper	9/6/2023 9:41 PM	Text Document	4 KB
jenkins	9/5/2023 7:52 AM	XML Document	3 KB
LICENSE	9/5/2023 7:50 AM	File	7 KB
README	9/5/2023 7:50 AM	File	1 KB
release	9/5/2023 7:50 AM	File	2 KB

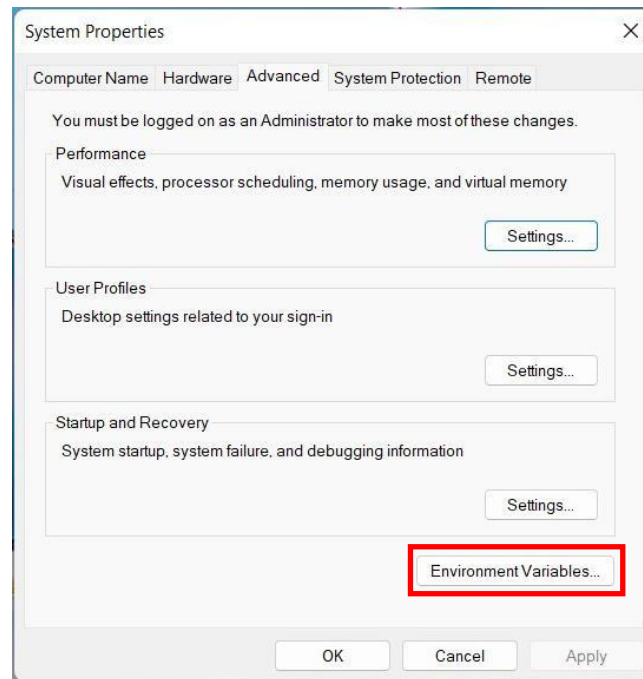
Step 3: Then copy the copy the bin location here.



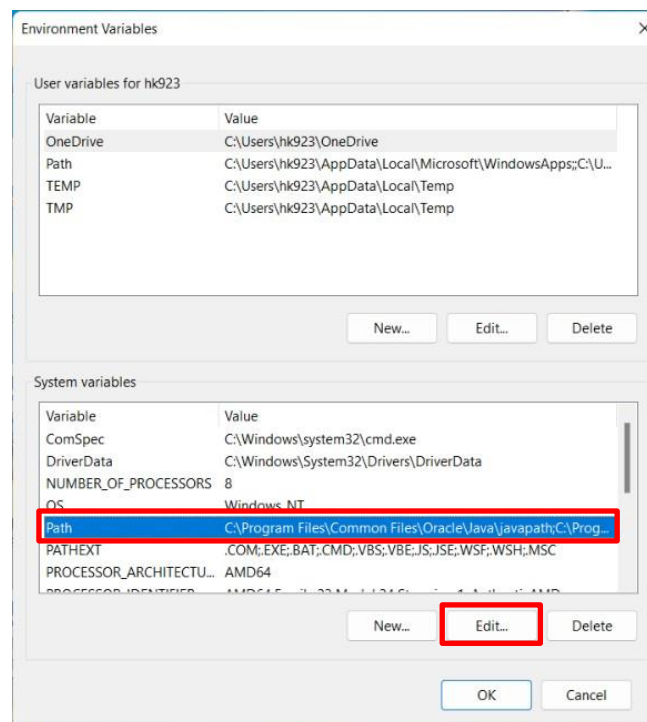
Step 4: Go to windows search bar ☐ Search for 'Environment Variables' & click on **Open**



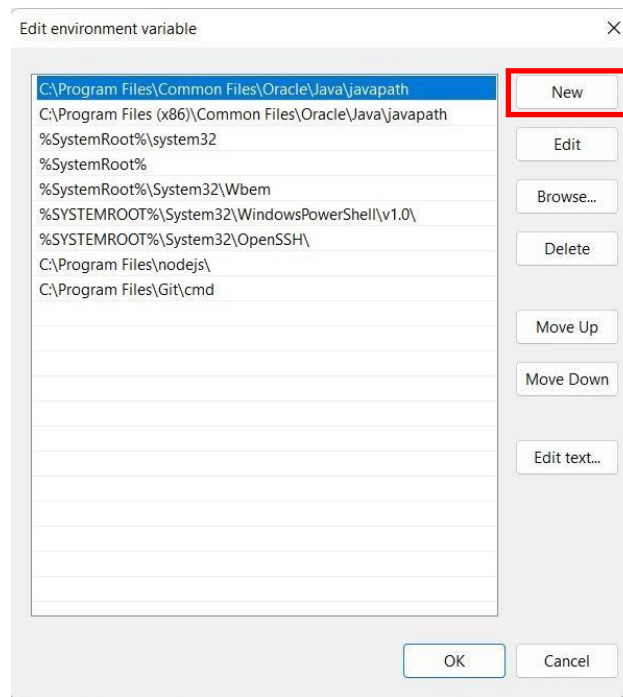
Step 5: Now click on **‘Environment Variables’** here.



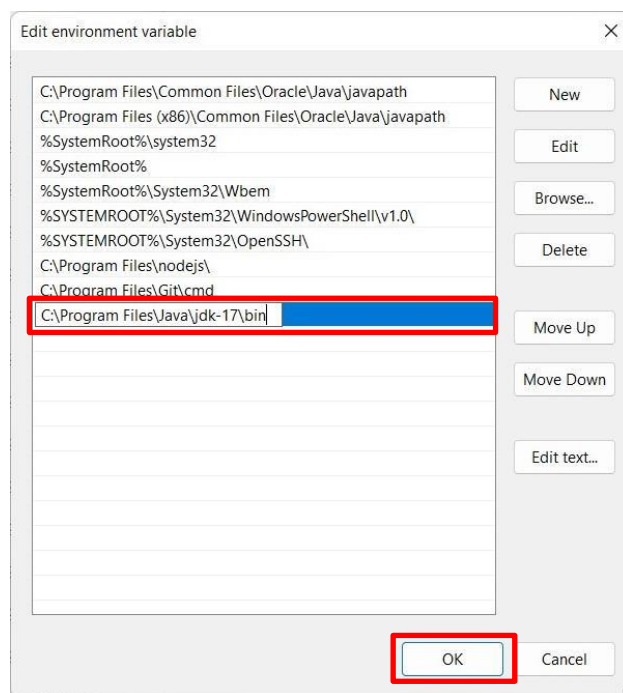
Step 6: Under System Variables, select Path location & click on **‘Edit’**.



Step 7: Click on **‘New’**.



Step 8: Now paste the bin location which was copied in step 3.



Step 9: Use Command Prompt to check whether the Java Program is working or not.

Give the command as '**java -version**' to check the Version and other details as shown below.

```
Microsoft Windows [Version 10.0.22000.2295]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Hk923>java -version
java version "17.0.8" 2023-07-18 LTS
Java(TM) SE Runtime Environment (build 17.0.8+9-LTS-211)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.8+9-LTS-211, mixed mode, sharing)
```

Step 10: Now again enter the command as '**javac**' to check the working of Java compiler on your Windows.


```
Command Prompt
Microsoft Windows [Version 10.0.22000.2295]
(c) Microsoft Corporation. All rights reserved.

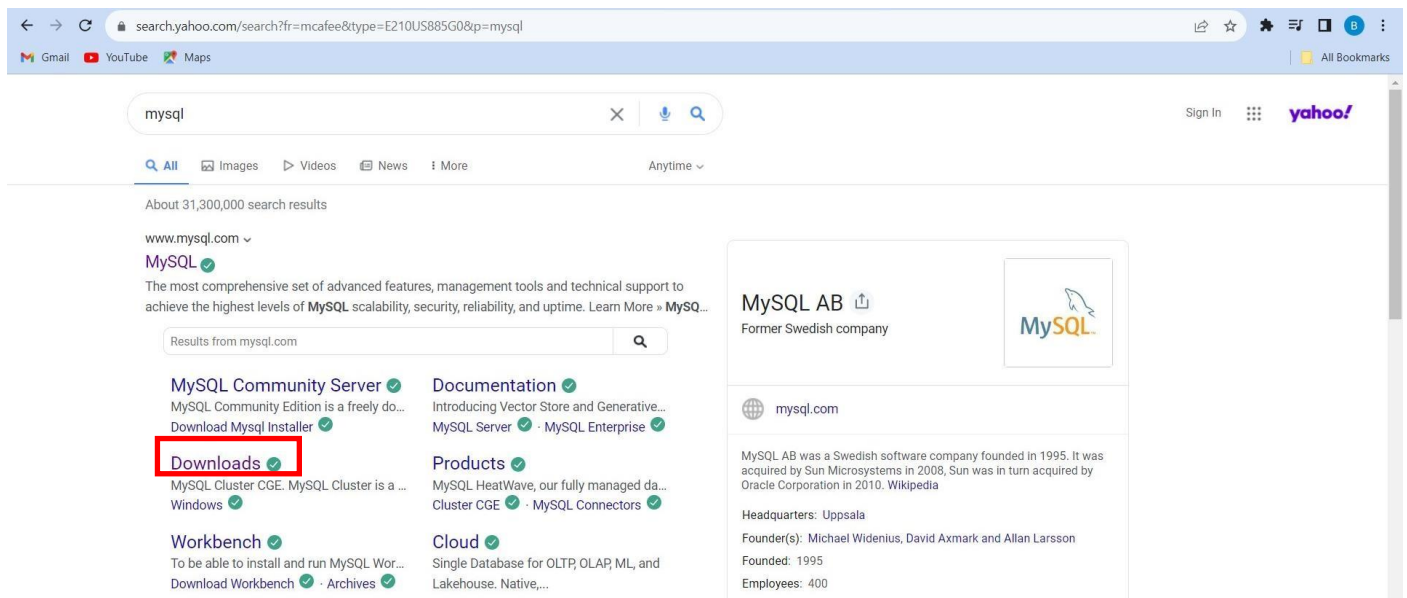
C:\Users\hk923>javac
Usage: javac <options> <source files>
where possible options include:
  -E<filename>           Read options and filenames from file
  -Akey[=value]          Options to pass to annotation processors
  --add-modules <module>[,<module>]*
                        Root modules to resolve in addition to the initial modules, or all modules
                        on the module path if <module> is ALL-MODULE-PATH.
  --boot-class-path <path>, -bootclasspath <path>
                        Override location of bootstrap class files
  --class-path <path>, -classpath <path>, -cp <path>
                        Specify where to find user class files and annotation processors
  -d <directory>         Specify where to place generated class files
  -deprecation           Output source locations where deprecated APIs are used
  --enable-preview       Enable preview language features. To be used in conjunction with either --source or --release.
  -encoding <encoding>   Specify character encoding used by source files
  -endorseddirs <dirs>   Override location of endorsed standards path
  -extdirs <dirs>        Override location of installed extensions
  -g                    Generate all debugging info
  -g: {lines,vars,source} Generate only some debugging info
  -g:none               Generate no debugging info
  -h <directory>        Specify where to place generated native header files
  -help, -help, -?      Print this help message
  --help-extra, -X      Print help on extra options
  -implicit:{none,class} Specify whether or not to generate class files for implicitly referenced files
```

5. Install DBMS (MySQL, PostgreSQL or any other) tools.

MySQL: MySQL is an open-source relational database management system (RDBMS). SQL is a language that programmers use to create, modify and extract data from the relational database, as well as control user access to the database. It manages the users, allows for network access and facilitates testing database integrity and creation of backups.

- Installation procedures for MySQL:

Step 1: Go to Web browser ☐ Search for MySQL ☐ Click on ‘Downloads’.



Step 2: Click on ‘MySQL Community (GPL) Downloads’.

MySQL Newsletter

[Subscribe »](#)

[Archive »](#)

Free Webinars

What's New with MySQL and MySQL HeatWave (Announcements at CloudWorld)
Thursday, October 05, 2023

Virtual Conference: Machine Learning for Beginners - From Data to Insights
Thursday, October 05, 2023

MySQL Security from Data Protection to Regulation Compliance
Thursday, October 19, 2023

[More »](#)

MySQL Enterprise Edition

MySQL Enterprise Edition includes the most comprehensive set of advanced features, management tools and technical support for MySQL.

[Learn More »](#)

[Customer Download »](#)

[Trial Download »](#)

MySQL Cluster CGE

MySQL Cluster is a real-time open source transactional database designed for fast, always-on access to data under high throughput conditions.

- MySQL Cluster
- MySQL Cluster Manager
- Plus, everything in MySQL Enterprise Edition

[Learn More »](#)

[Customer Download »](#) (Select Patches & Updates Tab, Product Search)

[Trial Download »](#)

[MySQL Community \(GPL\) Downloads »](#)

Step 3: Now click on ‘MySQL Installer for Windows’.

MySQL Community Downloads

- MySQL Yum Repository
- MySQL APT Repository
- MySQL SUSE Repository
- MySQL Community Server
- MySQL Cluster
- MySQL Router
- MySQL Shell
- MySQL Operator
- MySQL NDB Operator
- MySQL Workbench
- C API (libmysqlclient)
- Connector/C++
- Connector/J
- Connector/NET
- Connector/Node.js
- Connector/ODBC
- Connector/Python
- MySQL Native Driver for PHP
- MySQL Benchmark Tool
- Time zone description tables
- Download Archives
- MySQL Installer for Windows

ORACLE © 2023 Oracle

[Privacy](#) / [Do Not Sell My Info](#) | [Terms of Use](#) | [Trademark Policy](#) | [Cookie Preferences](#)

Step 4: Then select the second option & click on ‘Download’.

MySQL Community Downloads

◀ MySQL Installer

[General Availability \(GA\) Releases](#) [Archives](#) [Help](#)

MySQL Installer 8.0.34

Note: MySQL 8.0 is the final series with MySQL Installer. As of MySQL 8.1, use a MySQL product's MSI or Zip archive for installation. MySQL Server 8.1 and higher also bundle MySQL Configurator, a tool that helps configure MySQL Server.

Select Version:
8.0.34

Select Operating System:
Microsoft Windows

Windows (x86, 32-bit), MSI Installer	8.0.34	2.4M	Download
(mysql-installer-web-community-8.0.34.0.msi)			
Windows (x86, 32-bit), MSI Installer	8.0.34	331.3M	Download
(mysql-installer-community-8.0.34.0.msi)			

Step 5: Now you can Login or Sign up with your Oracle account or simply click on ‘No thanks, just start my download’.

MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

Login »

using my Oracle Web account

Sign Up »

for an Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can sign up for a free account by clicking the Sign Up link and following the instructions.

No thanks, just start my download.

ORACLE © 2023 Oracle

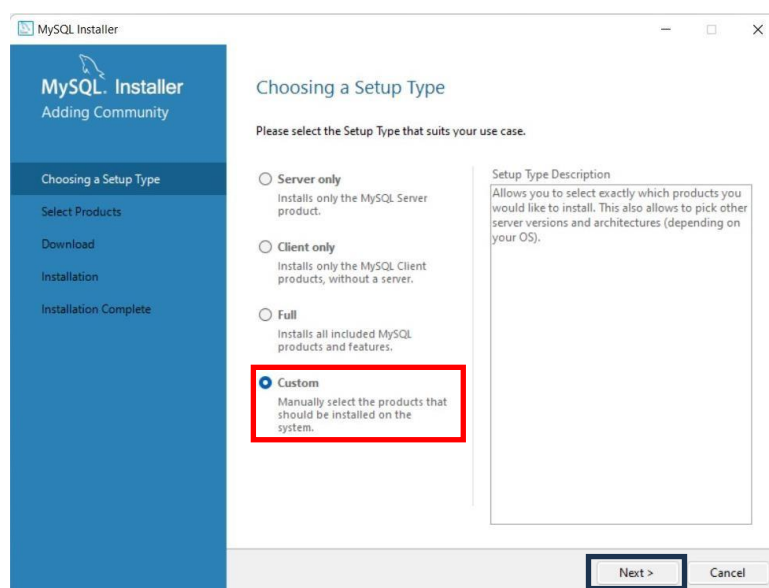
[Privacy / Do Not Sell My Info](#) | [Terms of Use](#) | [Trademark Policy](#) | [Cookie Preferences](#)

Step 6: Validating the internet connections.

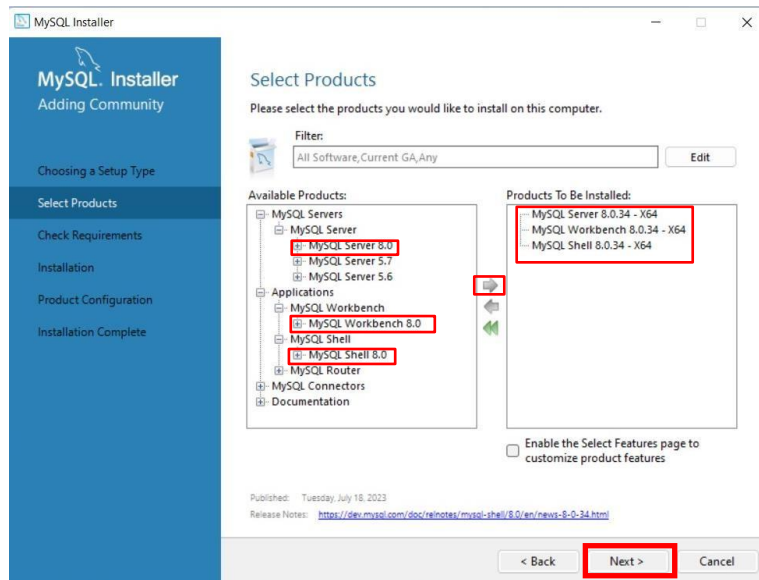


Installing the MySQL on both client & server side interfaces.

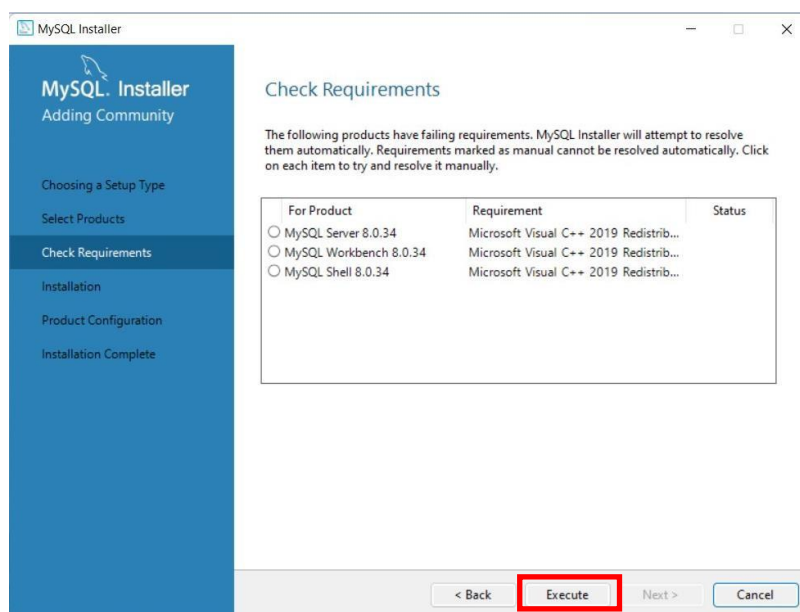
Step 7: Select 'Custom' & click on 'Next'.



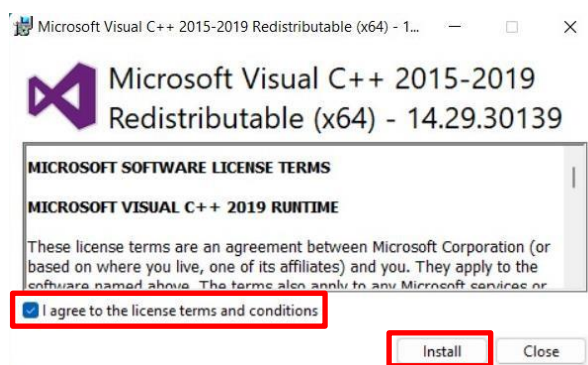
Step 8: Select the MySQL products from **Available products** to **Products to be installed** stage & click on **'Next'**.



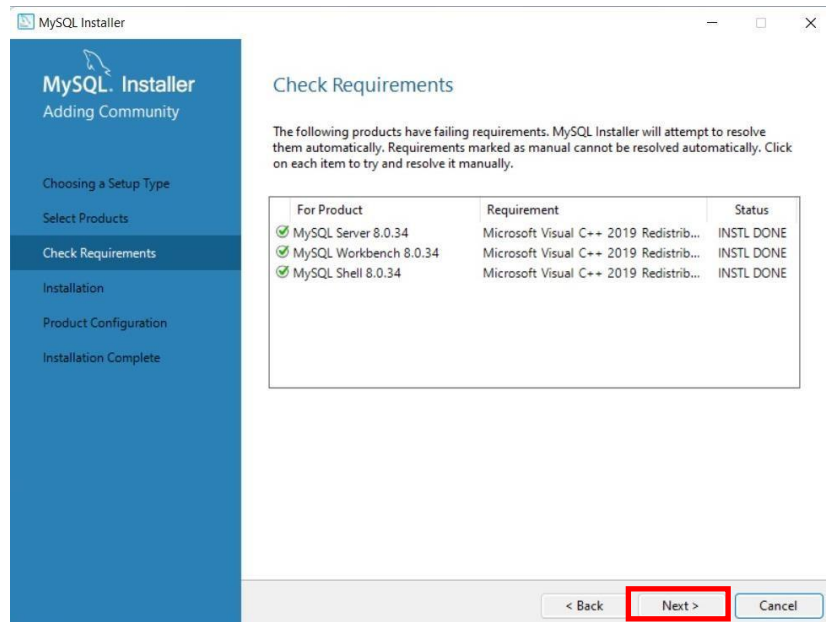
Step 9: Click on **'Execute'**.



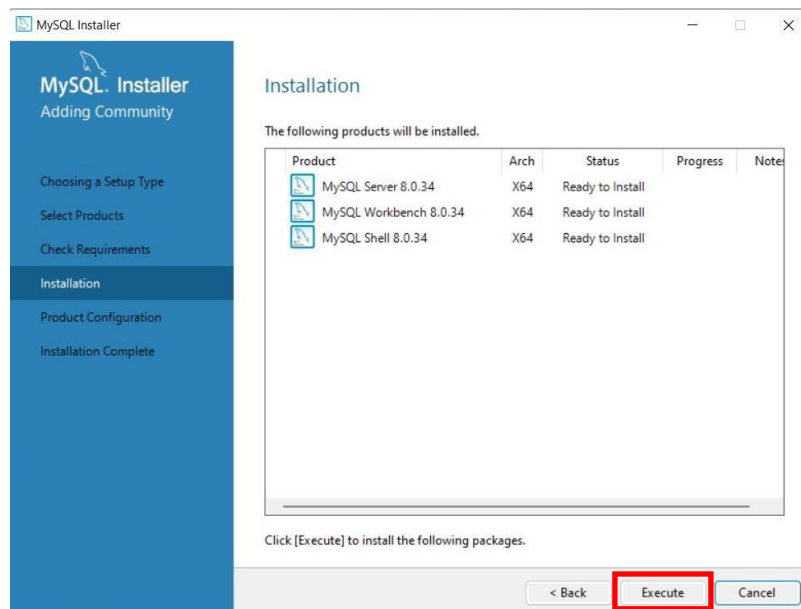
Step 10: Enable the option below & click on **'Install'**.



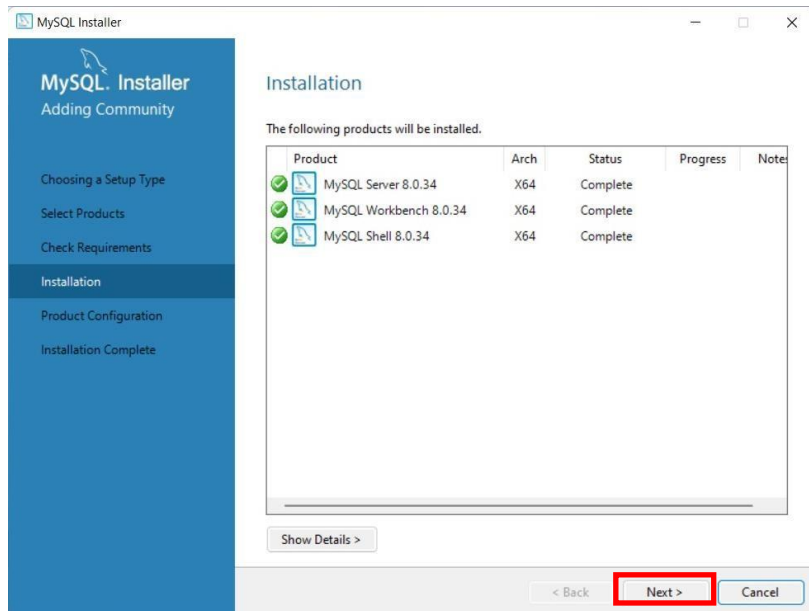
Step 11: After the installation of the products are complete, click on **'Next'**.



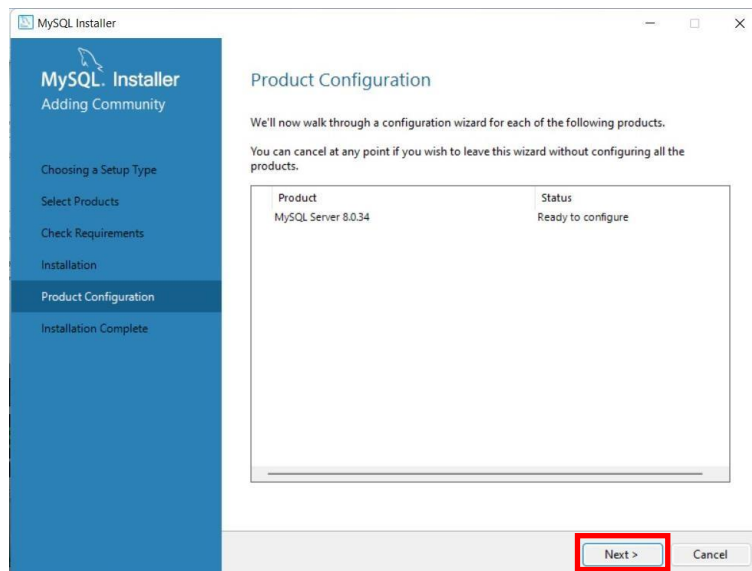
Step 12: Click on 'Execute'.



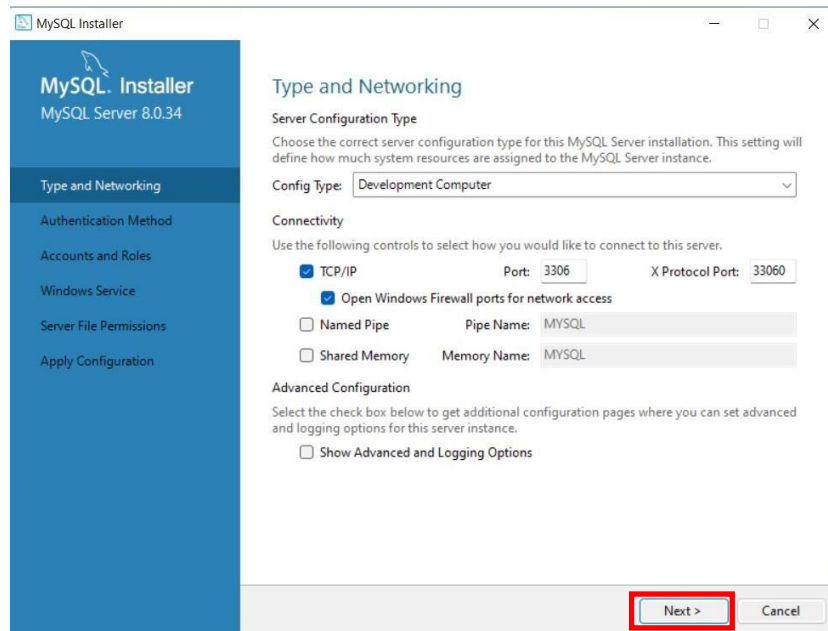
Step 13: After the complete execution, click on 'Next'.



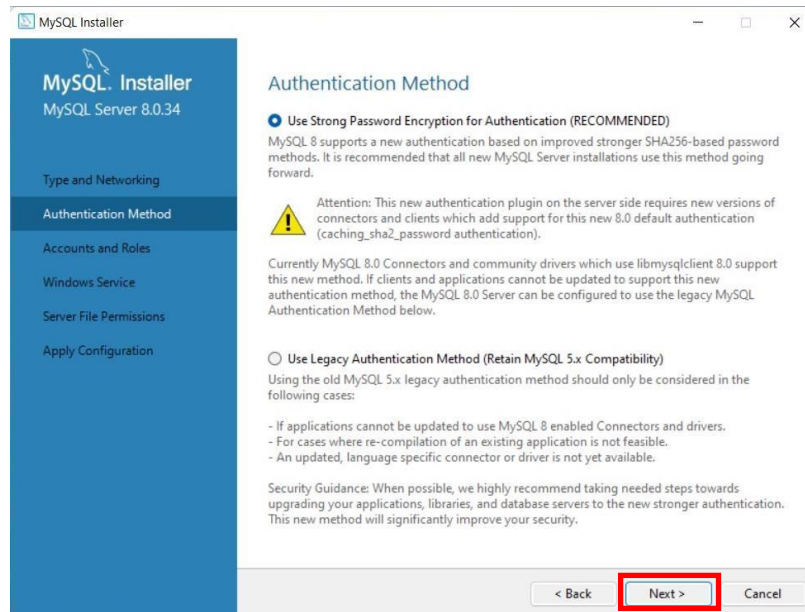
Step 14: Click on 'Next'.



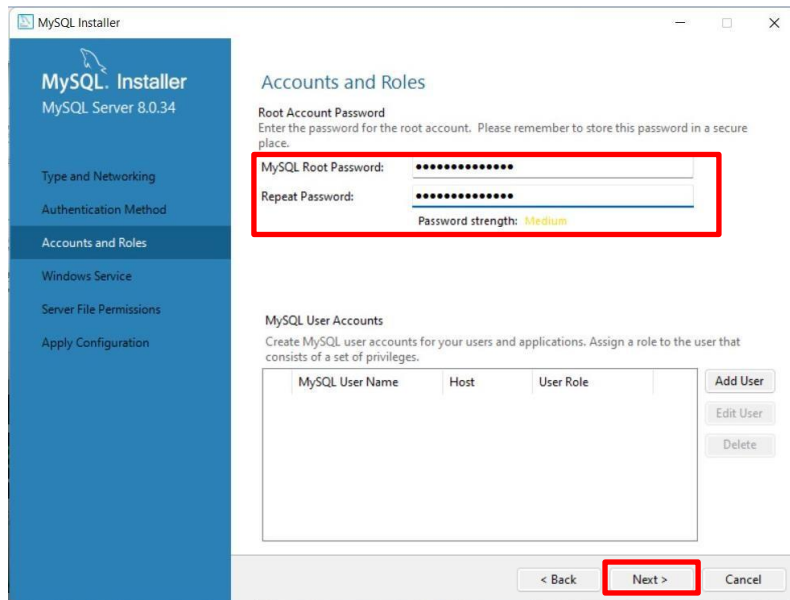
Step 15: Don't change the default selections, simply click on 'Next'.



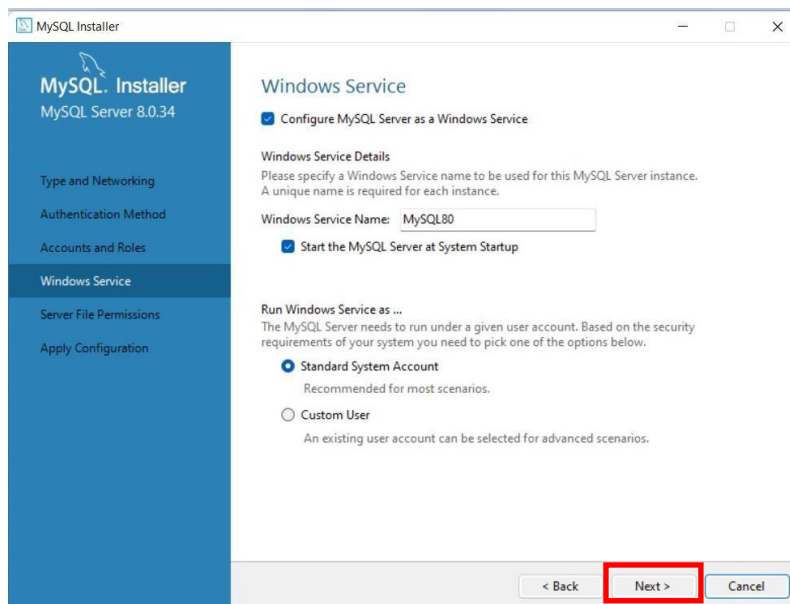
Step 16: Click on 'Next'.



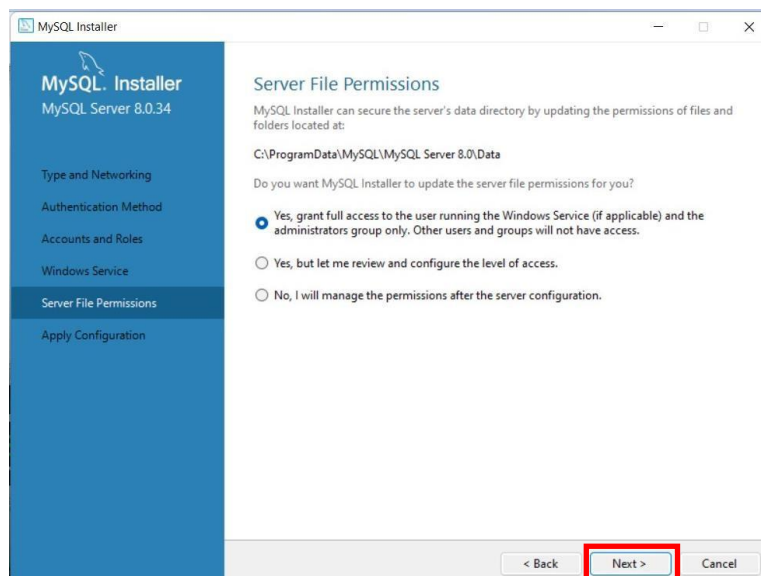
Step 17: Create a strong password for MySQL server & click on 'Next'.



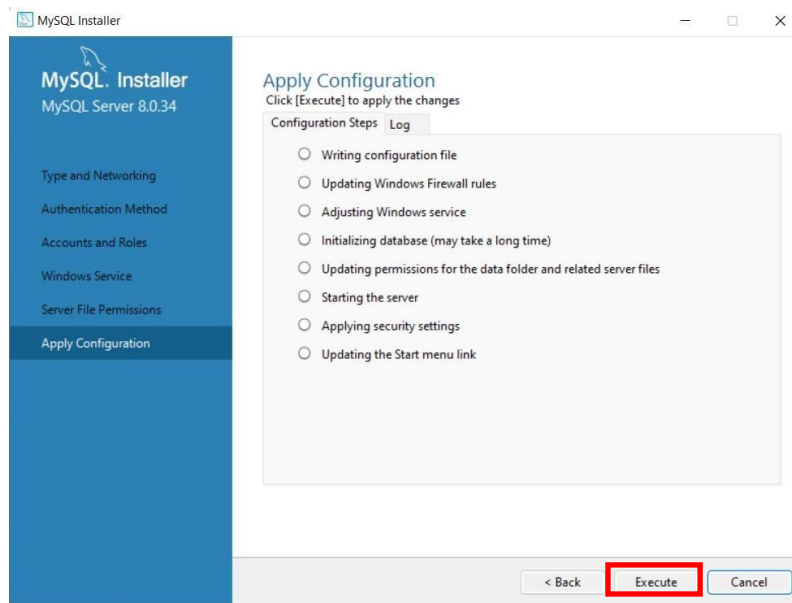
Step 18: Then click on 'Next'.



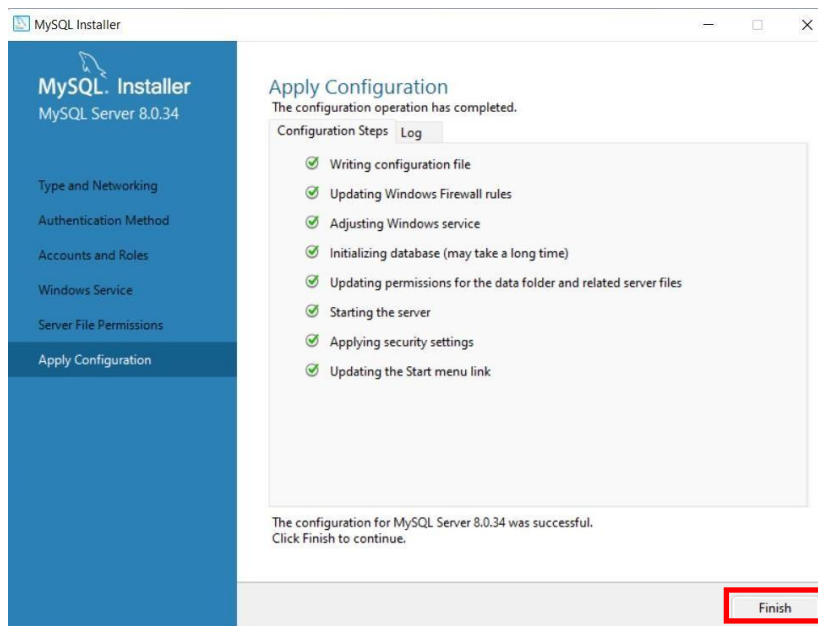
Step 19: Now click on 'Next'.



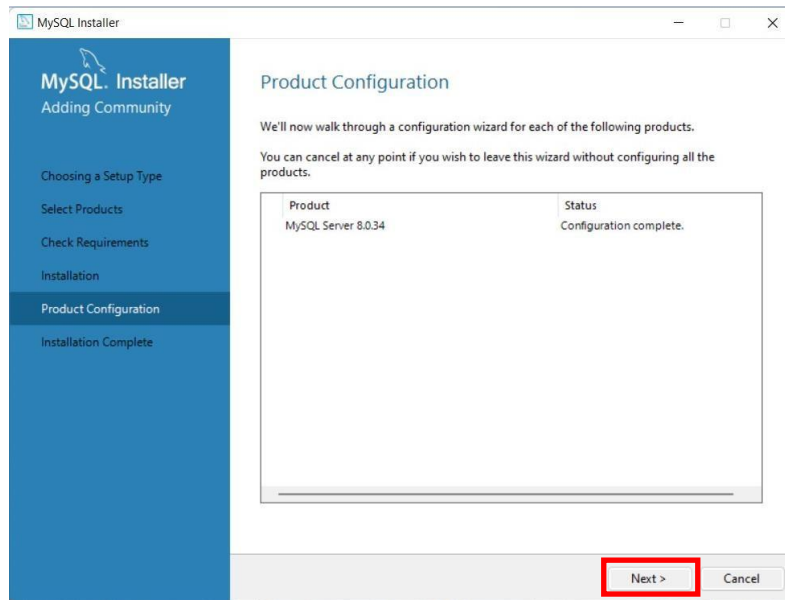
Step 20: Click on ‘Execute’.



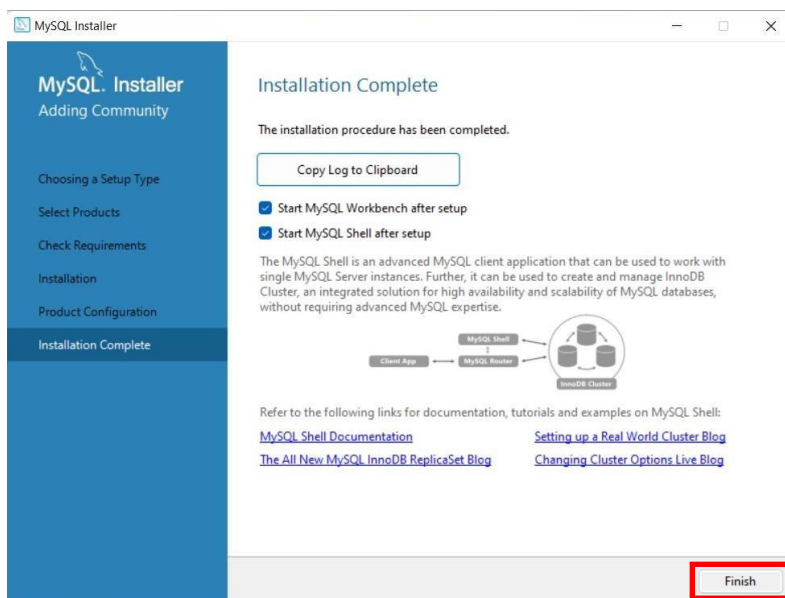
Step 21: Simply click on ‘Finish’.



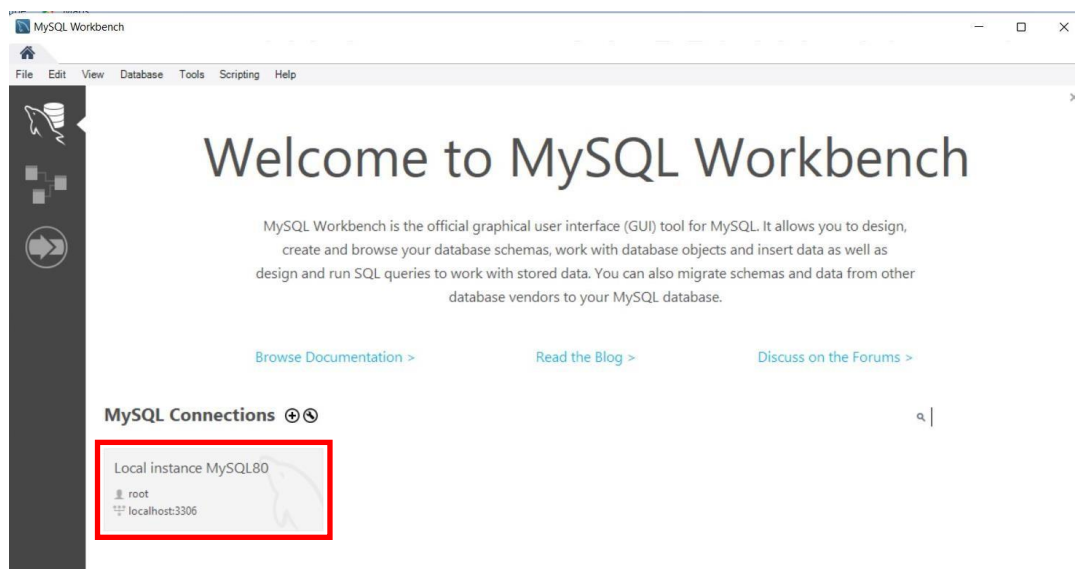
Step 22: Click on ‘Next’.



Step 23: Click on 'Finish'.



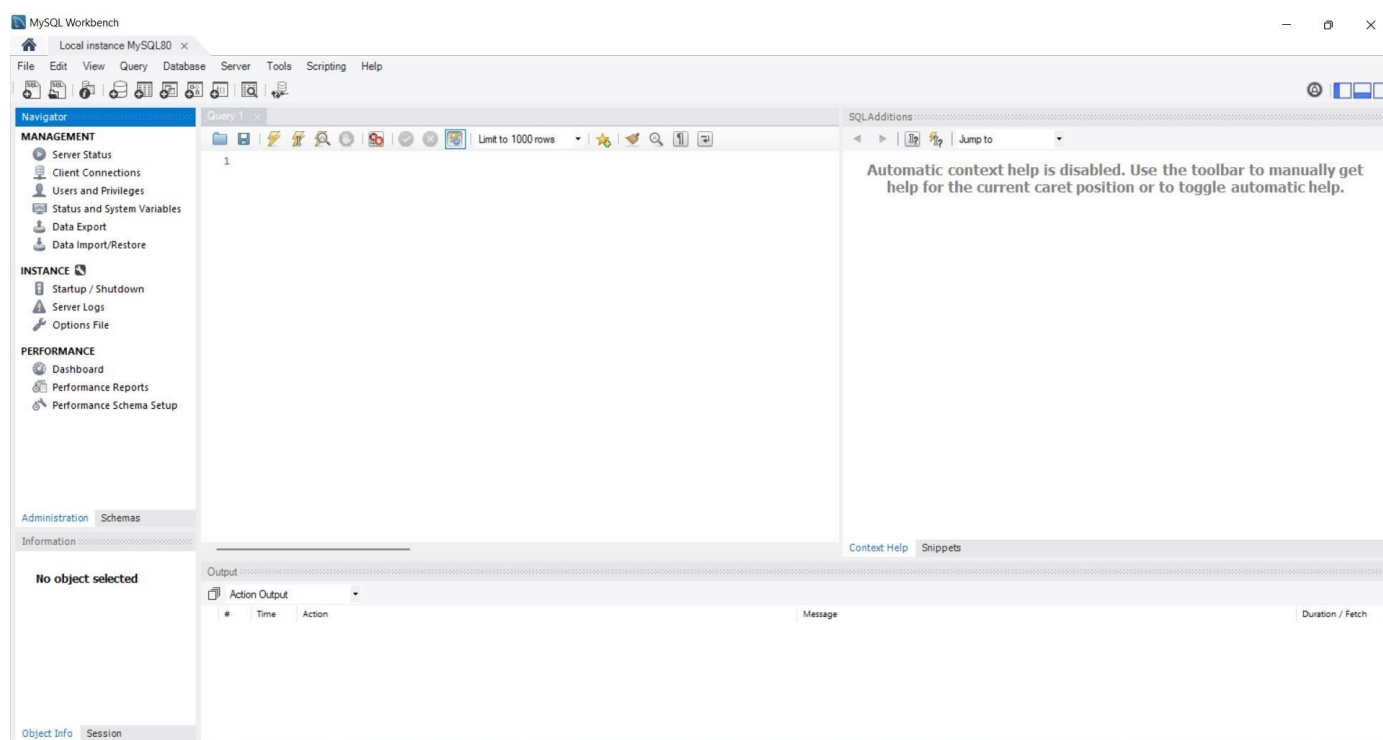
Step 24: Now click on 'Local instance MySQL80' here.



Step 25: Now enter your MySQL Server password here & click on 'OK'.



Step 26: Here is your MySQL Workbench is ready to use.



To run MySQL server:

Setup the Environment Variables for MySQL server & enter the command as '**mysql -u root -p**'. Also enter your password to use the MySQL Monitor.

```
Command Prompt - mysql -u root -p
Microsoft Windows [Version 10.0.22000.2295]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hik923>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 8.0.34 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Step 1: Go to command Prompt → give the command as '**mysql -version**' → to check the current version of MySQL.

Command Prompt - mysql -u root -p

```
Microsoft Windows [Version 10.0.22000.2295]  
(c) Microsoft Corporation. All rights reserved.
```

```
C:\Users\hk923>mysql --version  
mysql Ver 8.0.34 for Win64 on x86_64 (MySQL Community Server - GPL)
```

Step 2: Now give the command as '**mysql -u root -p**' & enter the password to enter into the MySQL server.

```
C:\Users\hk923>mysql -u root -p  
Enter password: *****  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 21  
Server version: 8.0.34 MySQL Community Server - GPL  
  
Copyright (c) 2000, 2023, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Step 3: Then enter '**show databases;**' to view the embedded databases.

Command Prompt - mysql -u root -p

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
+-----+  
4 rows in set (0.09 sec)
```

Step 4: Enter the command '**create database Employee**' to create a new database.

```
mysql> create database Employee;  
Query OK, 1 row affected (0.08 sec)
```

Step 5: Again enter the command 'show databases' to view the created databases as shown below.

```
mysql> show databases;
+-----+
| Database |
+-----+
| employee |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)
```