**DOP: / /2023 DOS: / /2023**

**Experiment No: 01**

**Title**: Study web analytics using open source tools like Matomo, Open Web Analytics, AWStats, Countly, Plausible.

**Theory**:

* **Web Analytics:**

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

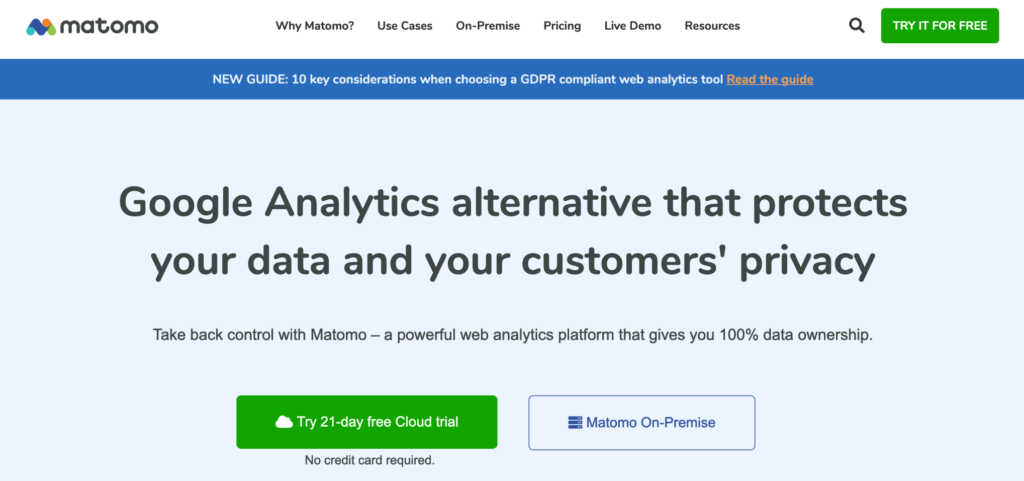
* **Benefits of Web Analytics**
  + - Measure online traffic
    - Tracking Bounce Rate
    - Optimizing and Tracking of Marketing Campaigns
    - Finding the Right Target Audience and its Capitalization
    - Conversion Rate Optimization (CRO)
    - Tracking business goals online

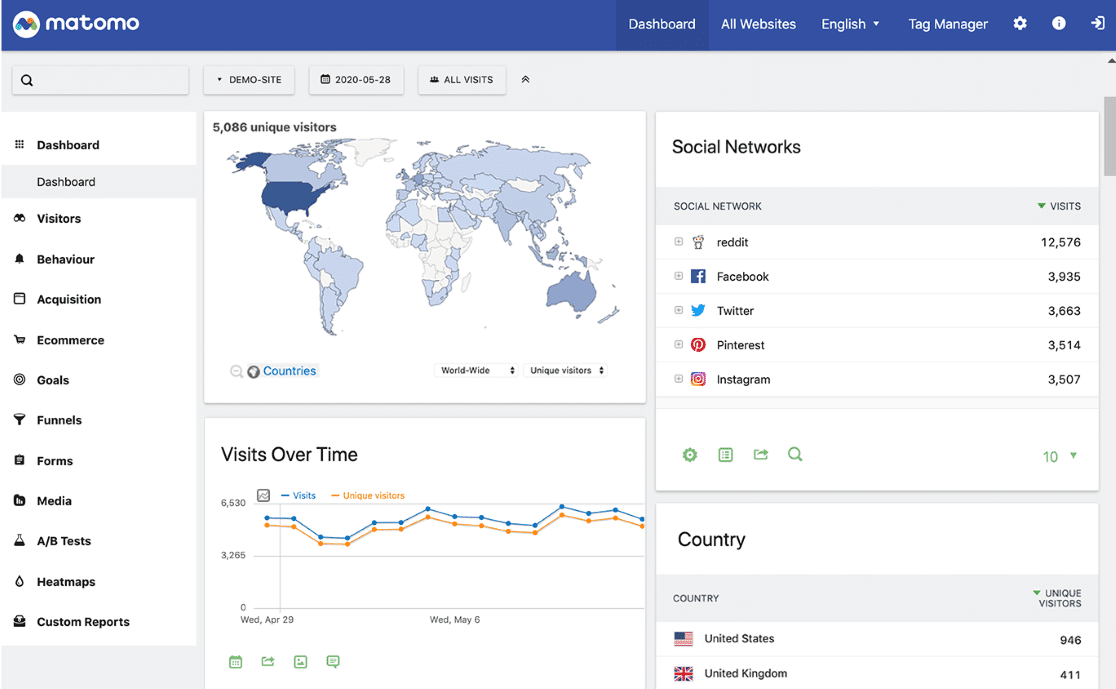
**Tools of Web Analytics:**

In web analytics, there is a multitude of different tools with complex purposes for tracking anything online. There are free and paid tools for tracking general traffic and even more specific goals. here are few webs’ analytics tools:

**🔸Matomo:**

Matomo, formerly known as Piwik, is a downloadable, Free (GPL licensed) web analytics software platform. It provides detailed reports on your website and its visitors, including the search engines and keywords they used, the language they speak, which pages they like, the files they download and so much more.





The installation of Matomo is simple and can be done by non-technical personnel, since it is enough to decompress a file, upload it by FTP and connect it with a database.

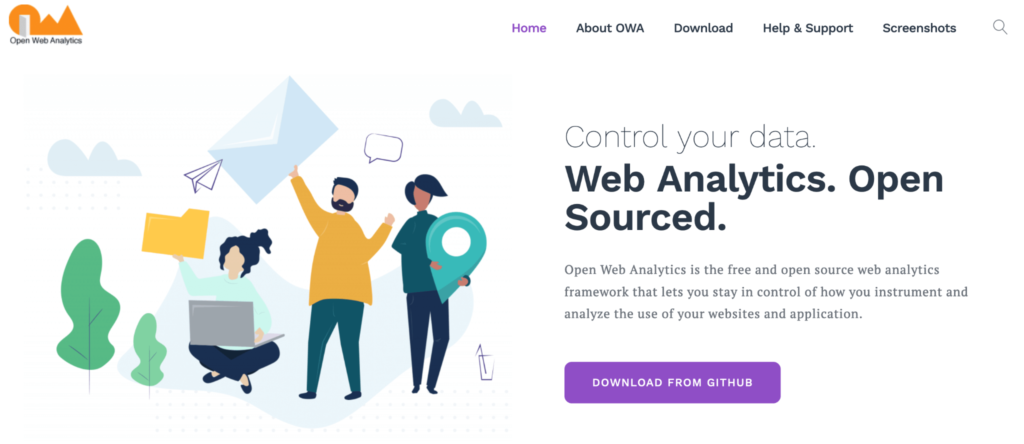
As far as features go, Matomo has a lot of the features that Google Analytics does, as well as heatmaps and A/B testing.Your pricing depends on how many visitors you get per month, plus whether you’re hosting the tool on your own servers or Matomo’s. For easy setup on Matomo’s servers, the monthly price starts at $23.

**🔸Open Web Analytics**

Open Web Analytics (OWA) helps you track and analyze the way visitors interact with your websites and applications. This web analytics tool comes with inbuilt support for tracking and monitoring WordPress sites too.You can watch your visitors’ mouse trails and heatmaps to analyze where they spend their time on your site.

OWA tells you about the number of unique visitors, their session duration, bounce rate. It also allows you to integrate goals.

However, you can’t monitor the stats for all your websites in one place, which makes some marketers avoid using it. Lastly, it’s hosted on-site but is completely free to use.



**🔸AWStats:**

AWStats is short for Advanced Web Statistics. AWStats is powerful log analyzer which creates advanced web, ftp, mail and streaming server statistics reports based on the rich data contained in server logs. Data is graphically presented in easy-to-read web pages.

AWStats statistics database files are saved in directory defined by the DirData parameter in configuration file.

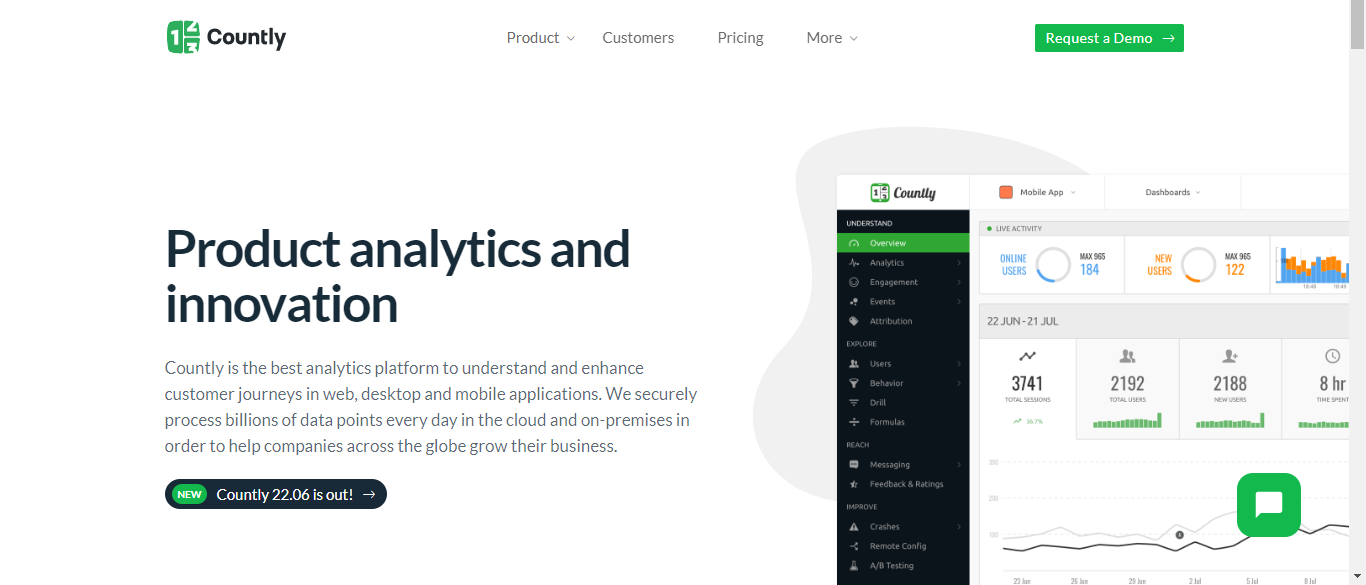
The Awstats interface displays traffic statistics from the Advanced Web Statistics (AWStats) software, which compiles information about how users access your website



**🔸Countly:**

Web Analytics is an integrated analytics platform which covers everything you need for data analytics like user analytics, heatmaps, customer feedback, extensive segmentation, JavaScript error crash reports, and more.

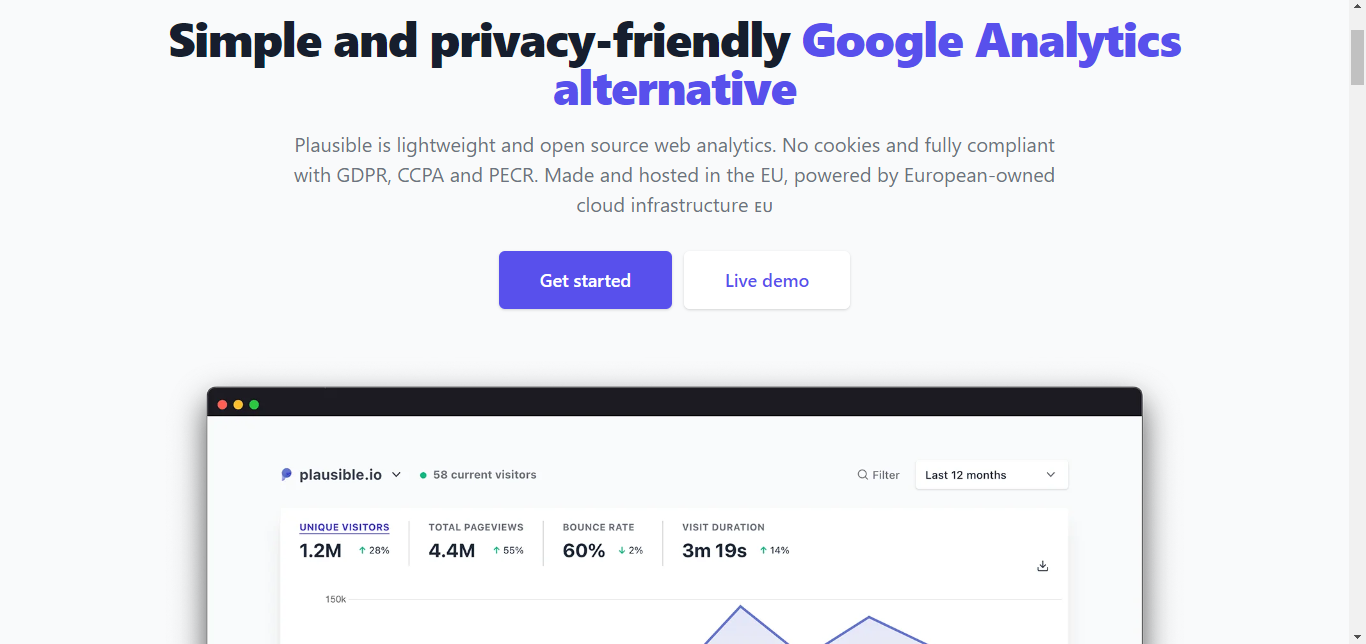
Countly can be deployed onto your own infrastructure, or in cloud servers based in the EU, so that data isn't stored in servers outside of GDPR jurisdiction. It doesn't offer a cookie-less tracking option, but it does have consent systems built in.



**🔸Plausible:**

Plausible Analytics is built for privacy-conscious site owners. You get valuable and actionable stats to help you improve your efforts while your visitors keep having a nice and enjoyable experience.

Plausible is lightweight analytics. Our script is 45 times smaller than Google Analytics. Your page weight will be cut down, your site will load faster and you'll reduce your carbon footprint for a greener and more sustainable web. A site with 10,000 monthly visitors can save 4.5 kg of CO2 emissions per year by switching.



**Conclusion: -**

Thus, we have studied different Web Analytics tools like Smart Look, Matomo, Open Web Analytics, AWStats, Countly and Plausible.

**DOP: / /2023 DOS: / /2023**

**Experiment No:2**

**Title:** 2.1 Installation of Typescript in windows, 2.2 Small code snippets for programs like Hello World, 2.3. Access Modifiers example using TypeScript,2.4 Inheritance example using TypeScript.

**2.1**Installation of Typescript in windows.

**Theory:**

* **Typescript:**

TypeScript is an open-source pure object-oriented programming language. It is a strongly typed superset of JavaScript which compiles to plain JavaScript. It contains all elements of the JavaScript. It is a language designed for large-scale JavaScript application development, which can be executed on any browser, any Host, and any Operating System. The TypeScript is a language as well as a set of tools. TypeScript is the ES6 version of JavaScript with some additional features.



TypeScript because of the following **benefits**.

* TypeScript supports Static typing, Strongly type, Modules, Optional Parameters, etc.
* TypeScript supports object-oriented programming features such as classes, interfaces, inheritance, generics, etc.
* TypeScript is fast, simple, and most importantly, easy to learn.
* TypeScript provides the error-checking feature at compilation time.
* TypeScript supports all JavaScript libraries because it is the superset of JavaScript.
* TypeScript support reusability because of the inheritance.
* TypeScript supports the latest JavaScript features, including ECMAScript 2015.
* TypeScript gives all the benefits of ES6 plus more productivity.
* Developers can save a lot of time with TypeScript.

**🔹TypeScript Installation:**

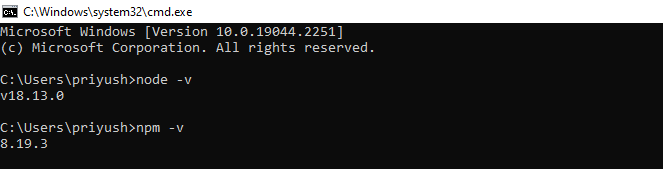
Pre-requisite to install TypeScript

* Text Editor or IDE
* Node.js Package Manager (npm)
* The TypeScript compiler

**Step**-**1**: Install Node.js. It is used to setup TypeScript on our local computer.

To install Node.js on Windows, go to the following link <https://nodejs.org/en/download/>

**Step**-**2**: To verify the installation was successful, enter the following command in the Terminal Window.



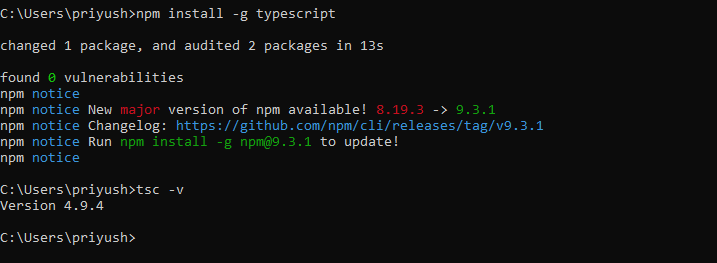
**Step-3** Install TypeScript. To install TypeScript, enter the following command in the Terminal Window.

$ npm install typescript --save-dev //As dev dependency

$ npm install typescript -g //Install as a global module

$ npm install typescript@latest -g //Install latest if you have an older version

**Step-4** To verify the installation was successful, enter the command $ tsc -v in the Terminal Window.



**2.2**

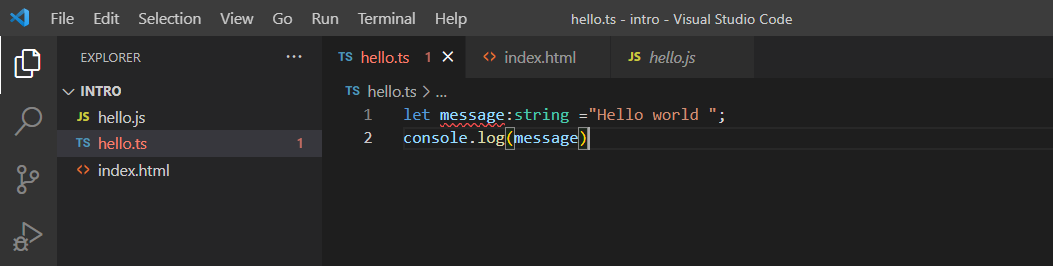
**Title**: Hello World Program in Typescript.

**Theory:**

We are going to learn how we can write a program in TypeScript, how to compile it, and how to run it. Also, we will see how to compiles the program and shows the error, if any.

Let us write a program in the text editor, save it, compile it, run it, and display the output to the console. To do this, we need to perform the following steps.

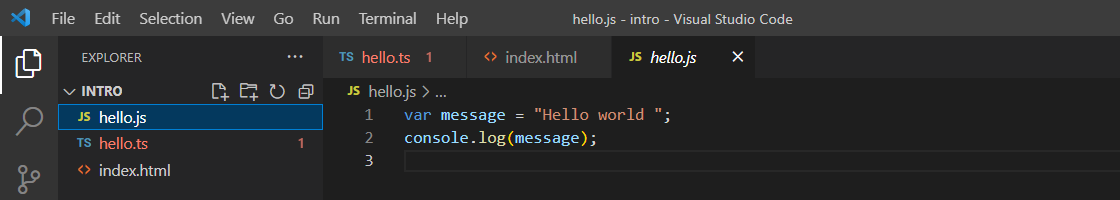
**Step-1** Open the Text Editor and write/copy the following code.



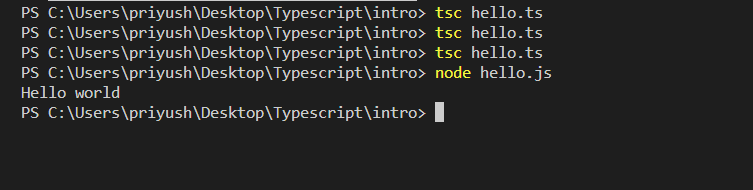
**Step-2** Save the above file as "**.ts**" extension.

**Step-3** Compile the TypeScript code. To compile the source code, open the **command prompt**, and then goes to the file directory location where we saved the above file. For example, if we save the file on the desktop, go to the terminal window and type: - **cd Desktop/folder name**. Now, type the following command tsc **fileName. Ts** for compilation and press **Enter**.

It will generate JavaScript file with ".js" extension at the same location where the TypeScript source file exists. The below ".js" file is the output of TypeScript (.ts) file.



**Step-4** Now, to run the above JavaScript file, type the following command in the terminal window: node filename.js and press Enter. It gives us the final output as:



**2.3**

**Title**: Access Modifiers example using TypeScript.

**Theory:**

* **Access Modifiers:**

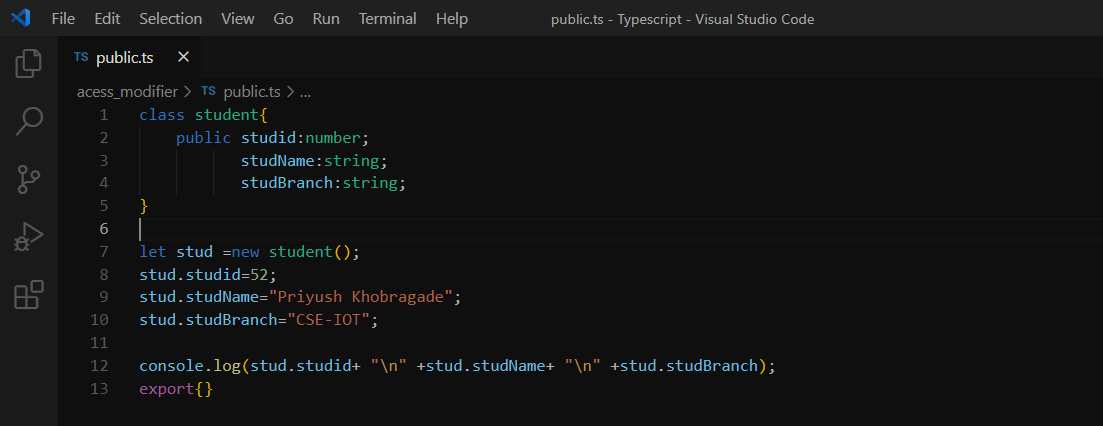
Typescript allows us to use access modifiers at the class level. It gives direct access control to the class member. These class members are functions and properties. We can use class members inside its own class, anywhere outside the class, or within its child or derived class.

The access modifier increases the security of the class members and prevents them from invalid use. We can also use it to control the visibility of data members of a class. If the class does not have to be set any access modifier, TypeScript automatically sets public access modifier to all class members.

The TypeScript access modifiers are of three types. These are:

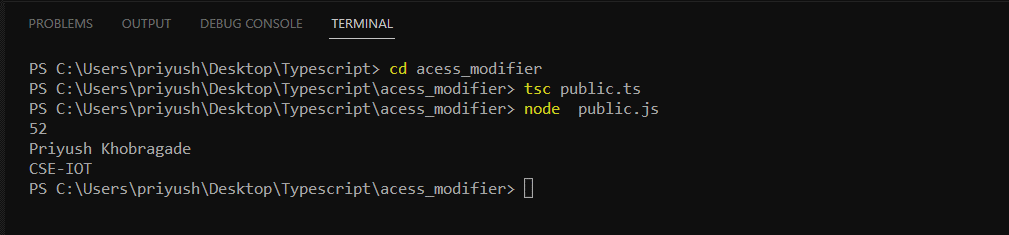
1. Public
2. Private
3. Protected.

1.Public

In TypeScript by default, all the members (properties and methods) of a class are public. So, there is no need to prefix members with this keyword. We can access this data member anywhere without any restriction.

**Input**:

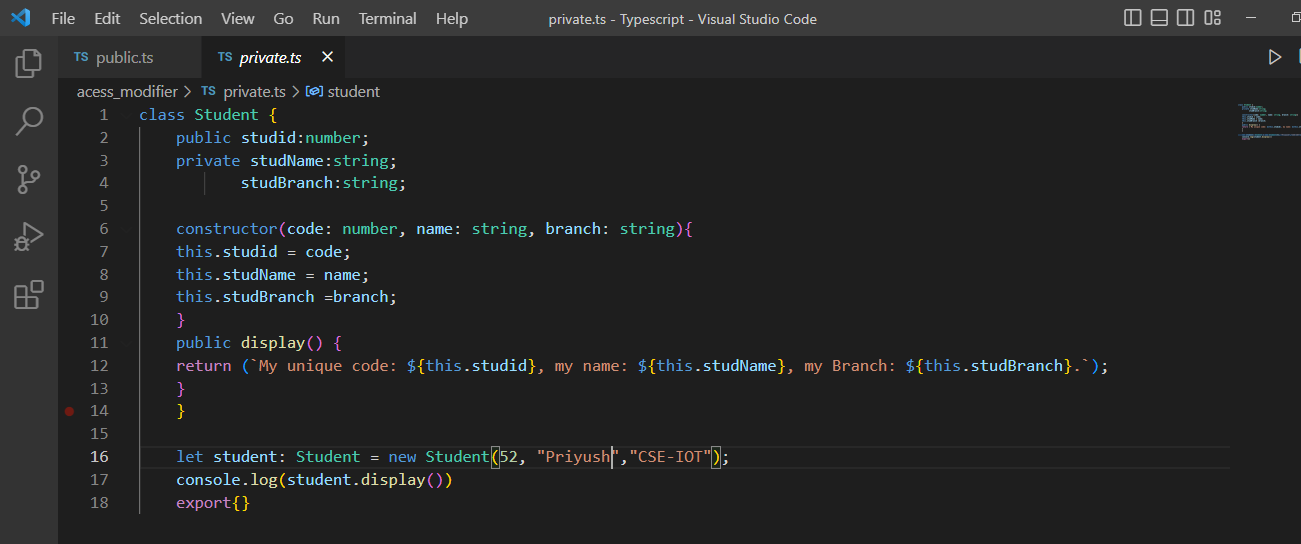
**Output**:



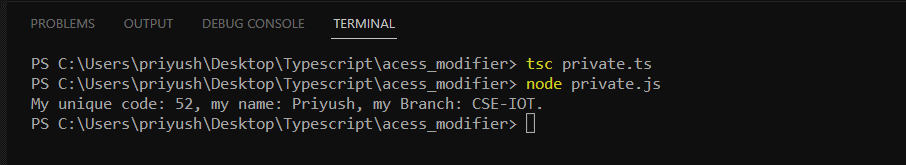
**Private**

The private access modifier cannot be accessible outside of its containing class. It ensures that the class members are visible only to that class in which it is containing.

**Input:**

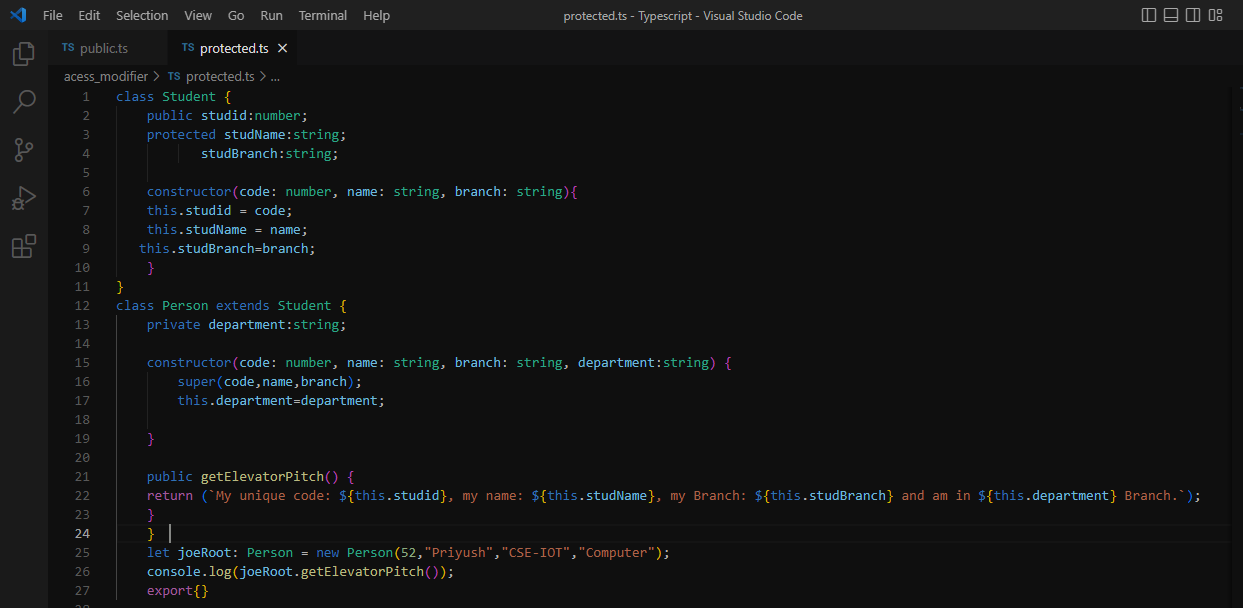


**Output:**

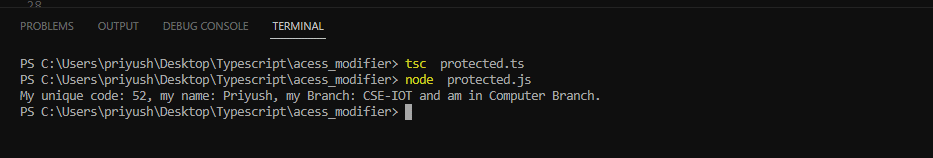


**Protected:**

A Protected access modifier can be accessed only within the class and its subclass. We cannot access it from the outside of a class in which it is containing.

**Input**:

**Output**:



**2.4**

**Title**: Inheritance example using TypeScript.

**Theory:**

* **TypeScript Inheritance:**

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

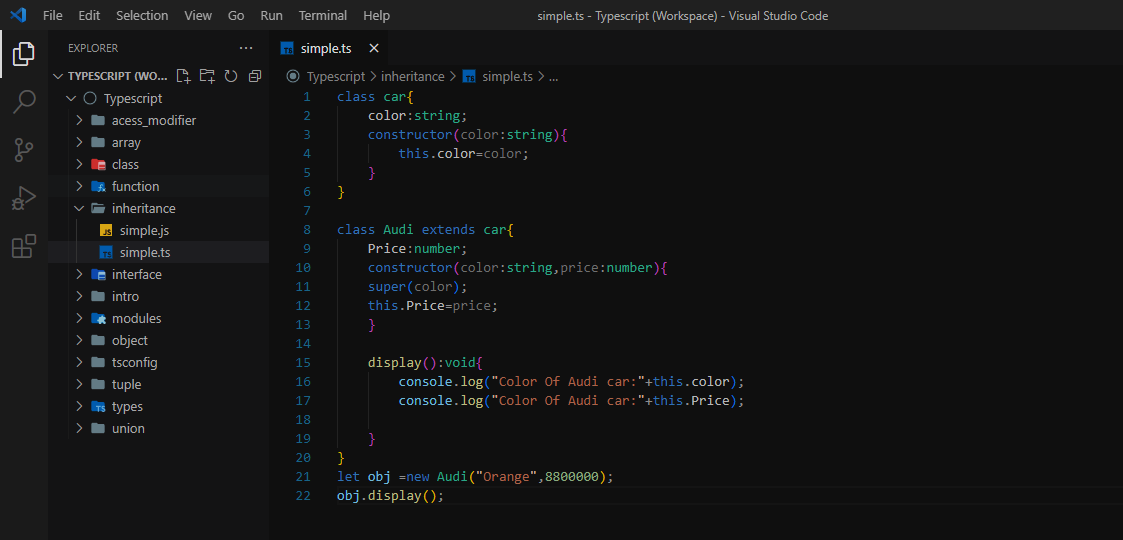
The TypeScript uses class inheritance through the extends keyword. TypeScript supports only **single** inheritance and **multilevel** inheritance. It doesn't support multiple and hybrid inheritance.

**Syntax**

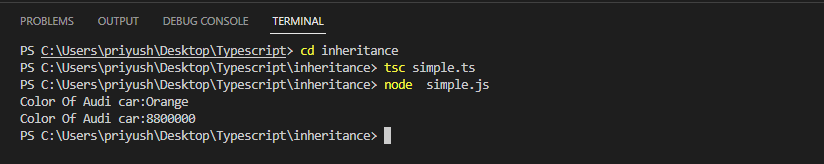
We can declare a class inheritance as below.

1. class sub\_class\_name extends super\_class\_
2. {
3. // methods and fields
4. {

**Simple Inheritance:**



**Output:**



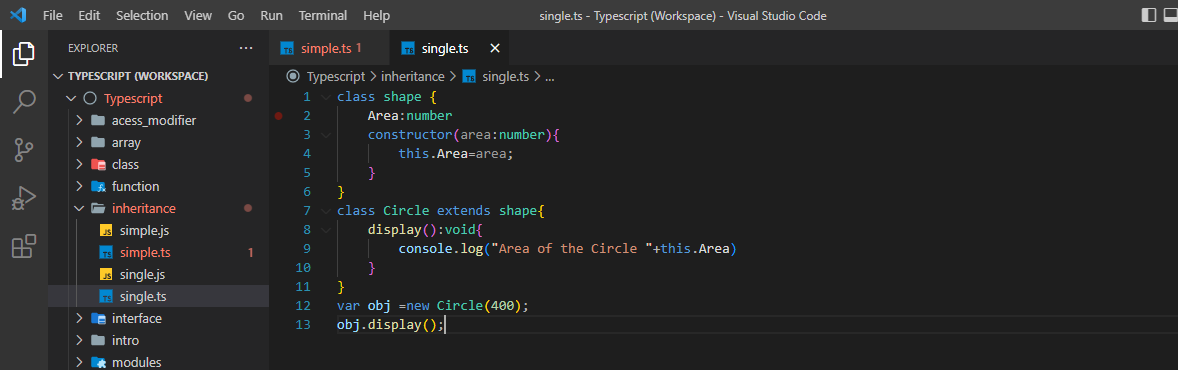
* **Types of Inheritance:**

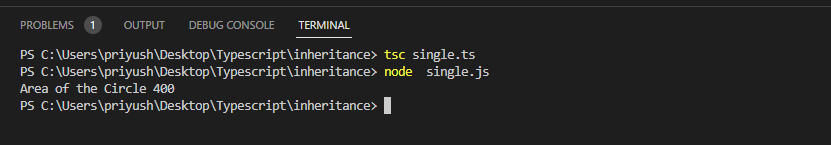
We can classify the inheritance into the five types. These are:

1. Single Inheritance
2. Multilevel Inheritance
3. Multiple Inheritance
4. Hierarchical Inheritance
5. Hybrid Inheritance

**Single Inheritance**

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

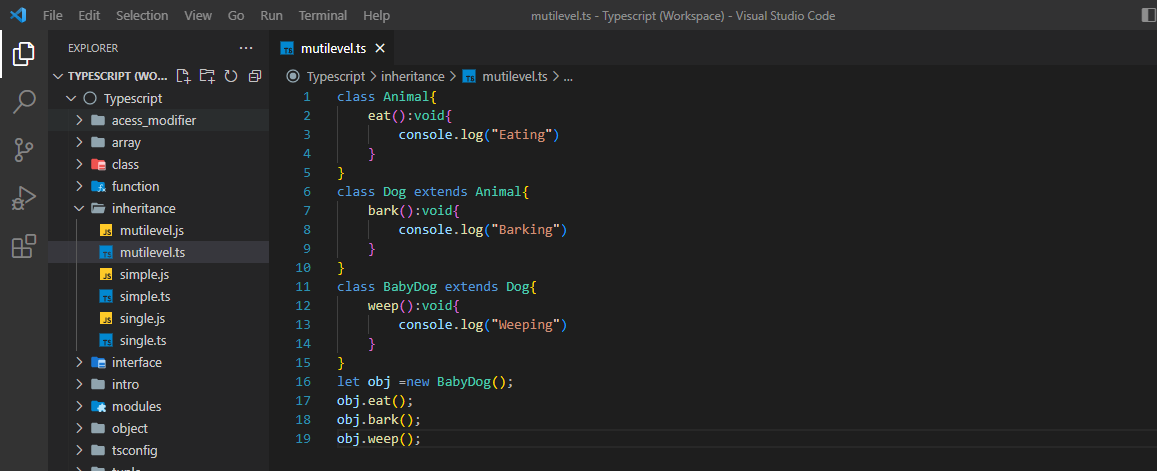
**Input**:

**Output**:

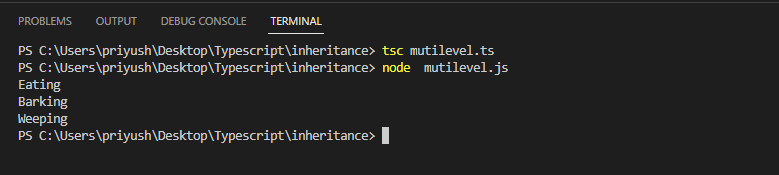
**Multilevel Inheritance**

When a derived class is derived from another derived class, then this type of inheritance is known as multilevel inheritance. Thus, a multilevel inheritance has more than one parent class.

**Input**:



**Output**:



**Multiple Inheritance:**

When an object or class inherits the characteristics and features form more than one parent class, then this type of inheritance is known as multiple inheritance.

**Hierarchical Inheritance:**

When more than one subclass is inherited from a single base class, then this type of inheritance is known as hierarchical inheritance.

**Hybrid Inheritance:**

When a class inherits the characteristics and features from more than one form of inheritance, then this type of inheritance is known as Hybrid inheritance.

**Conclusion: -** We Successfully implementInstallation of Typescript in windows, small code snippets for programs like Hello World, Access Modifiers example using TypeScript, Inheritance example using TypeScript.

**DOP: / /2023 DOS: / /2023**

**Experiment No: 08**

**Title:** Build Hello World App in Node.js

**Theory:**

**🔹What is Node.js:**

Node.js is a cross-platform runtime environment and library for running JavaScript applications outside the browser. It is used for creating server-side and networking web applications. It is open source and free to use. It can be downloaded from this link https://nodejs.org/en/.

Many of the basic modules of Node.js are written in JavaScript. Node.js is mostly used to run real-time server applications.

**🔹Features of Node.js:**

Following is a list of some important features of Node.js that makes it the first choice of software architects.

* **Extremely fast**: Node.js is built on Google Chrome's V8 JavaScript Engine, so its library is very fast in code execution.
* **I/O is Asynchronous and Event Driven:** All APIs of Node.js library are asynchronous i.e. non-blocking. So a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call. It is also a reason that it is very fast.
* **Single threaded**: Node.js follows a single threaded model with event looping.
* **Highly Scalable**: Node.js is highly scalable because event mechanism helps the server to respond in a non-blocking way.
* **No buffering**: Node.js cuts down the overall processing time while uploading audio and video files. Node.js applications never buffer any data. These applications simply output the data in chunks.
* **Open source**: Node.js has an open-source community which has produced many excellent modules to add additional capabilities to Node.js applications.
* **License**: Node.js is released under the MIT license.

**🔹Install Node.js on Windows:**

To install and setup an environment for Node.js, you need the following two softwares available on your computer:

1. Text Editor.
2. Node.js Binary installable

**Text Editor:**

The text editor is used to type your program. For example: Notepad is used in Windows, vim or vi can be used on Windows as well as Linux or UNIX. The name and version of the text editor can be different from operating system to operating system.

The files created with text editor are called source files and contain program source code. The source files for Node.js programs are typically named with the extension ".js".

**The Node.js Runtime:**

The source code written in source file is simply JavaScript. It is interpreted and executed by the Node.js interpreter.

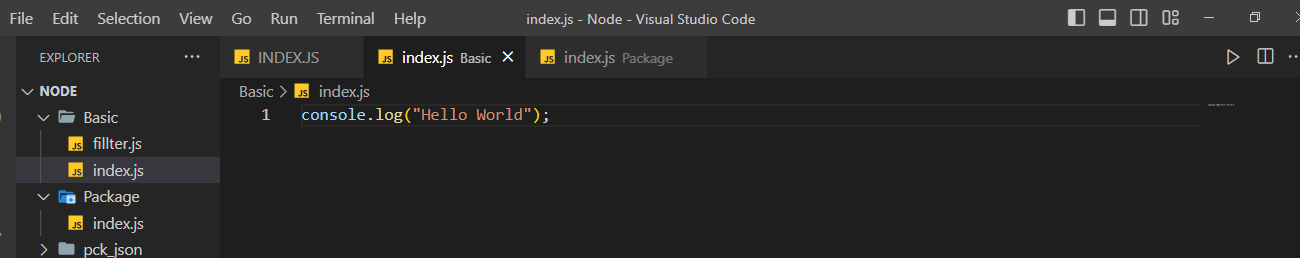
**How to download Node.js:**

You can download the latest version of Node.js installable archive file from <https://nodejs.org/en/>

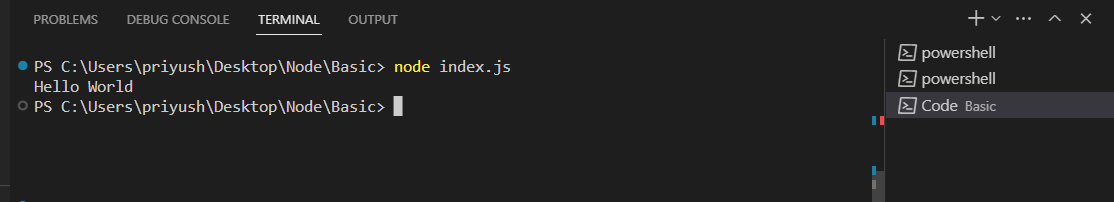
*Hello World App in Node.js*

**Input**:

*Console base:*



**Output**:

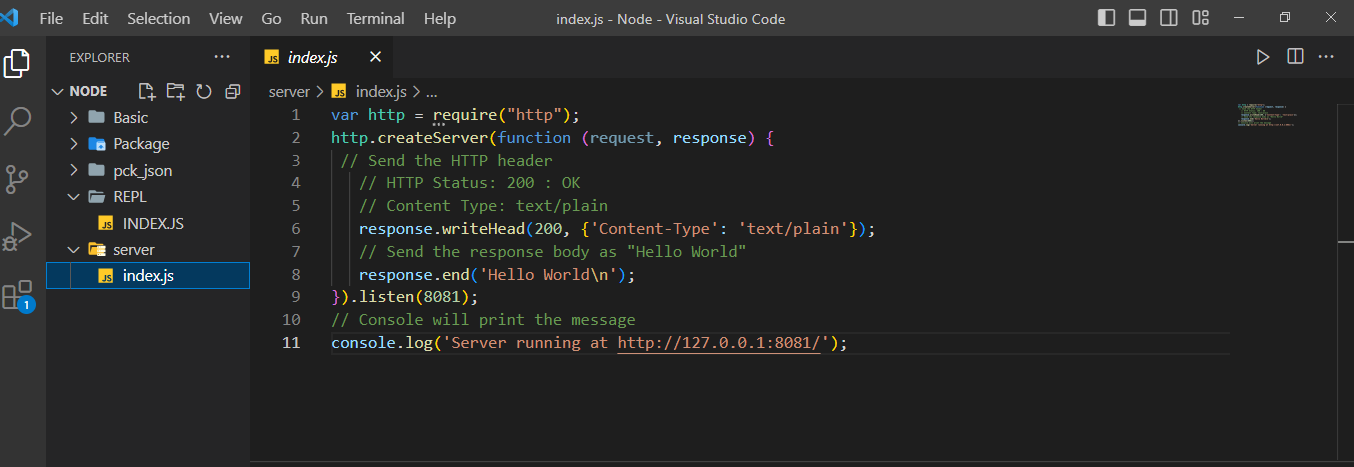


*Node.js web-based Example*

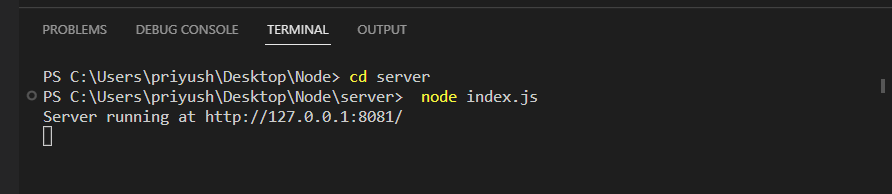
A node.js web application contains the following three parts:

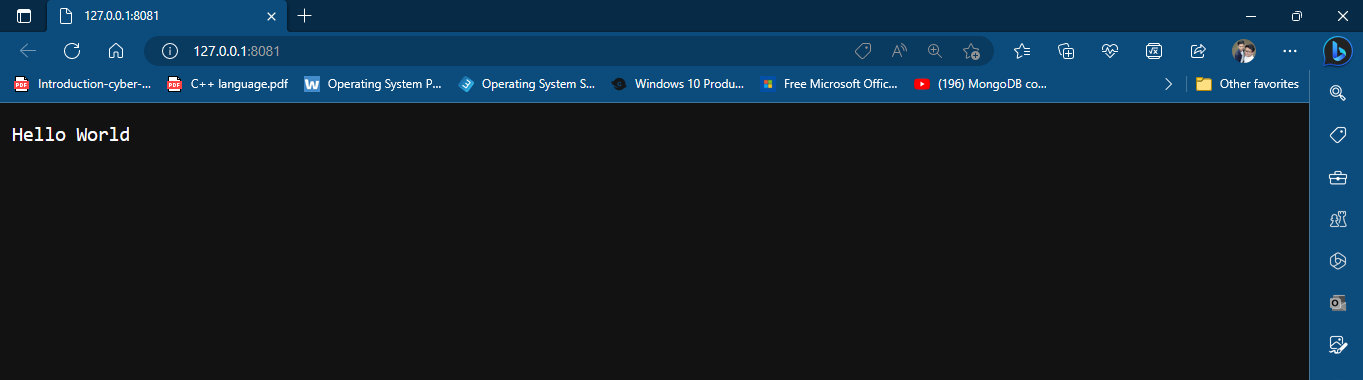
1. **Import required modules**: The "require" directive is used to load a Node.js module.
2. **Create server**: You have to establish a server which will listen to client's request similar to Apache HTTP Server.
3. **Read request and return response:** Server created in the second step will read HTTP request made by client which can be a browser or console and return the response

**Input**:



**Output**:





**Conclusion: -** We Successfully implementInstallation of NodeJS in windows, small code for programs like Hello World.

**DOP: / /2023 DOS: / /2023**

**Experiment No: 09**

**Title:** Modules in Node.js (Networking, File system, Web module).

**Theory:**

**🔹Networking Module:**

**Node.js Net**

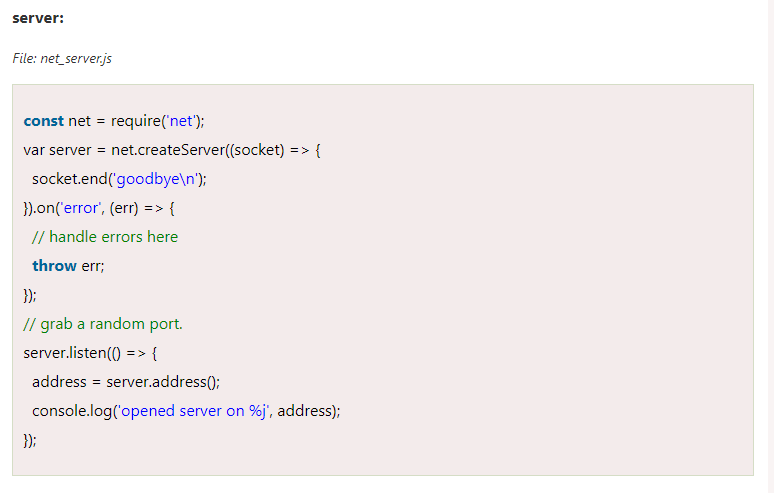
Node.js provides the ability to perform socket programming. We can create chat application or communicate client and server applications using socket programming in Node.js. The Node.js net module contains functions for creating both servers and clients.

Node.js Net Example

In this example, we are using two command prompts:

1. Node.js command prompt for server.
2. Window's default command prompt for client.

**Syntax: var net = require('net');**





**🔹Node.js File System (FS):**

In Node.js, file I/O is provided by simple wrappers around standard POSIX functions. Node File System (fs) module can be imported using following syntax:

Syntax: var fs = require("fs")

**Node.js FS Reading File**

Every method in fs module has synchronous and asynchronous forms.

Asynchronous methods take a last parameter as completion function callback. Asynchronous method is preferred over synchronous method because it never blocks the program execution where as the synchronous method blocks.

**Let's take an example:**

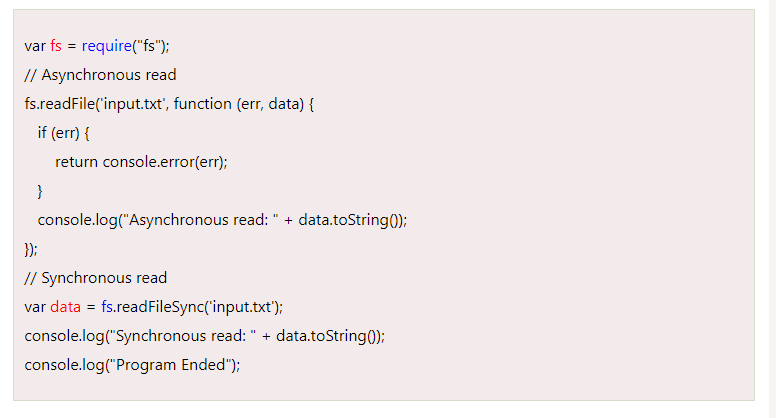
Create a text file named "input.txt" having the following content.

*File: input.txt*

Hello NodeJS

Let's take an example to create a JavaScript file named "main.js" having the following code:

File: main.js



**Node.js Open a file**

Syntax :fs.open(path, flags[, mode], callback)

*Parameter explanation:*

Following is the description of parameters used in the above syntax:

* **path**: This is a string having file name including path.
* **flags**: Flag specifies the behavior of the file to be opened. All possible values have been mentioned below.
* **mode**: This sets the file mode (permission and sticky bits), but only if the file was created. It defaults to 0666, readable and writeable.
* **callback**: This is the callback function which gets two arguments (err, fd).

**🔹Node.js Web Module**:

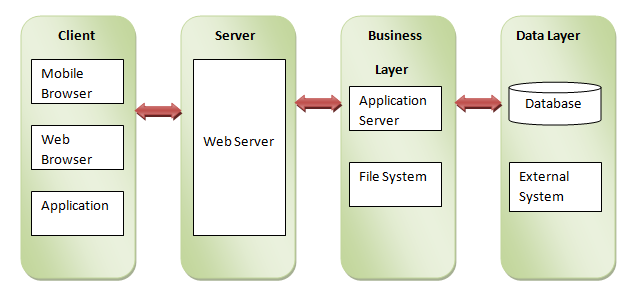
**What is Web Server:**

Web Server is a software program that handles HTTTP requests sent by HTTP clients like web browsers, and returns web pages in response to the clients. Web servers usually respond with html documents along with images, style sheets and scripts.

**Web Application Architecture**

A web application can be divided in 4 layers:

* **Client Layer**: The Client layer contains web browsers, mobile browsers or applications which can make HTTP request to the web server.
* **Server Layer:** The Server layer contains Web server which can intercepts the request made by clients and pass them the response.
* **Business Layer**: The business layer contains application server which is utilized by web server to do required processing. This layer interacts with data layer via data base or some external programs.
* **Data Layer**: The Data layer contains databases or any source of data.



**Creating Web Server using Node.js syntax**:

1. var http = require('http');
2. var fs = require('fs');
3. var url = require('url');
4. // Create a server
5. http.createServer( function (request, response) {
6. }

**Conclusion: -** We Successfully implementand understanding Modules in Node.js

**DOP: / /2023 DOS: / /2023**

**Experiment No: 03**

**Title:** create simple html hello world using angular js framework.

**Theory:**

**🔹What is AngularJS:**

* 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.
* AngularJS is developed by Google.
* It is an excellent framework for building single phase applications and line of business applications.

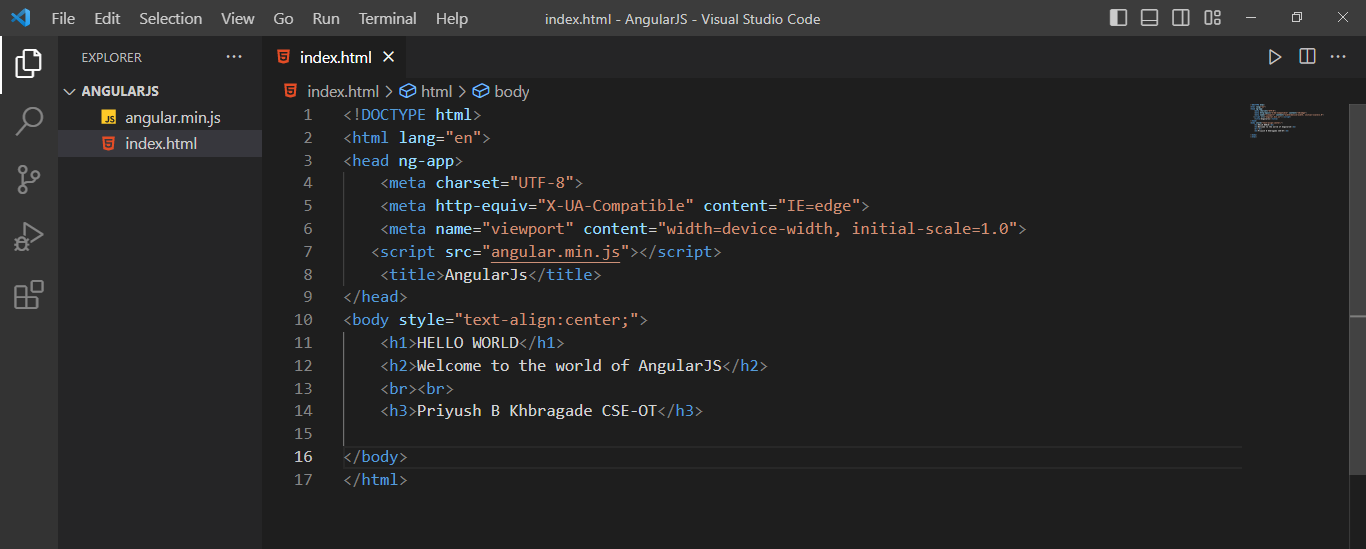
**🔹Advantage of AngularJS:**

There are a lot of JavaScript frameworks for building web applications. So, it is a genuine question, why to use Angular JS.

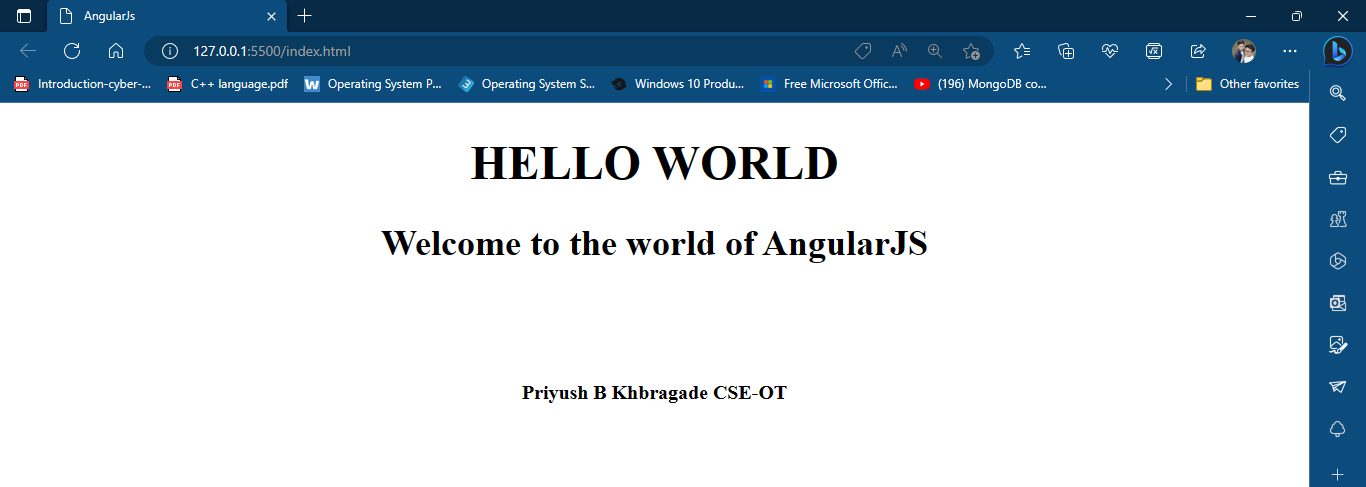
Following are the advantages of AngularJS over other JavaScript frameworks:

* **Dependency Injection:** Dependency Injection specifies a design pattern in which components are given their dependencies instead of hard coding them within the component.
* **Two-way data binding:** AngularJS creates a two way data-binding between the select element and the orderProp model. orderProp is then used as the input for the orderBy filter.
* **Testing**: Angular JS is designed in a way that we can test right from the start. So, it is very easy to test any of its components through unit testing and end-to-end testing.
* **Model View Controller**: In Angular JS, it is very easy to develop application in a clean MVC way. You just have to split your application code into MVC components i.e. Model, View and the Controller.

**Input**:



**Output**:



**Conclusion: -**  Thus we implement create simple html hello world using angular js framework.

**DOP: / /2023 DOS: / /2023**

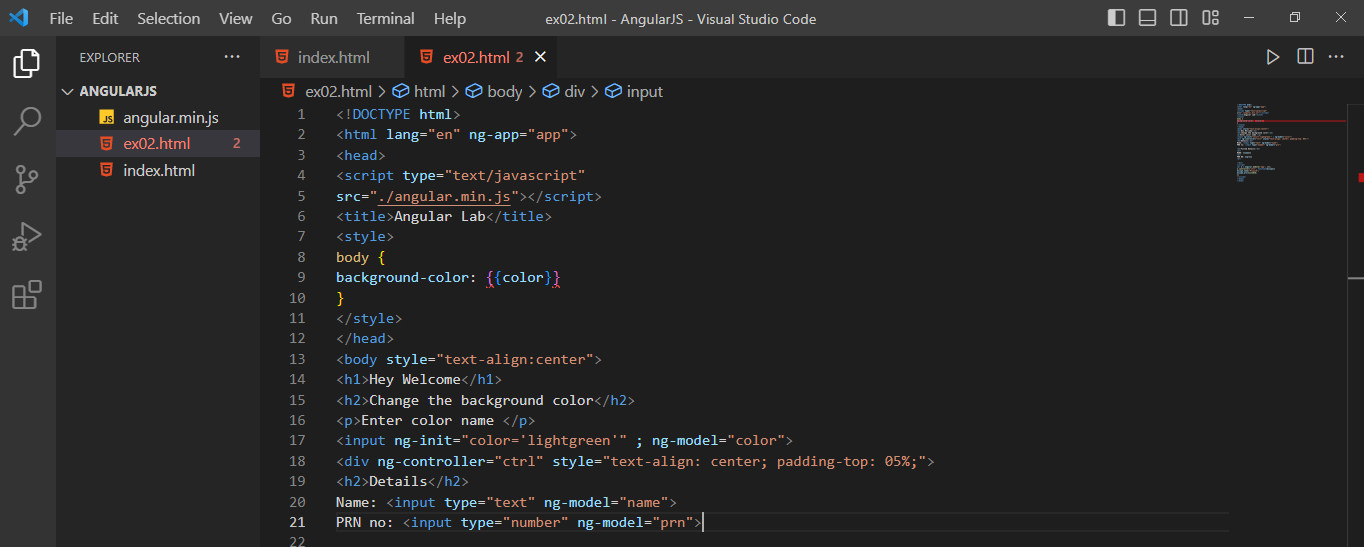
**Experiment No: 04**

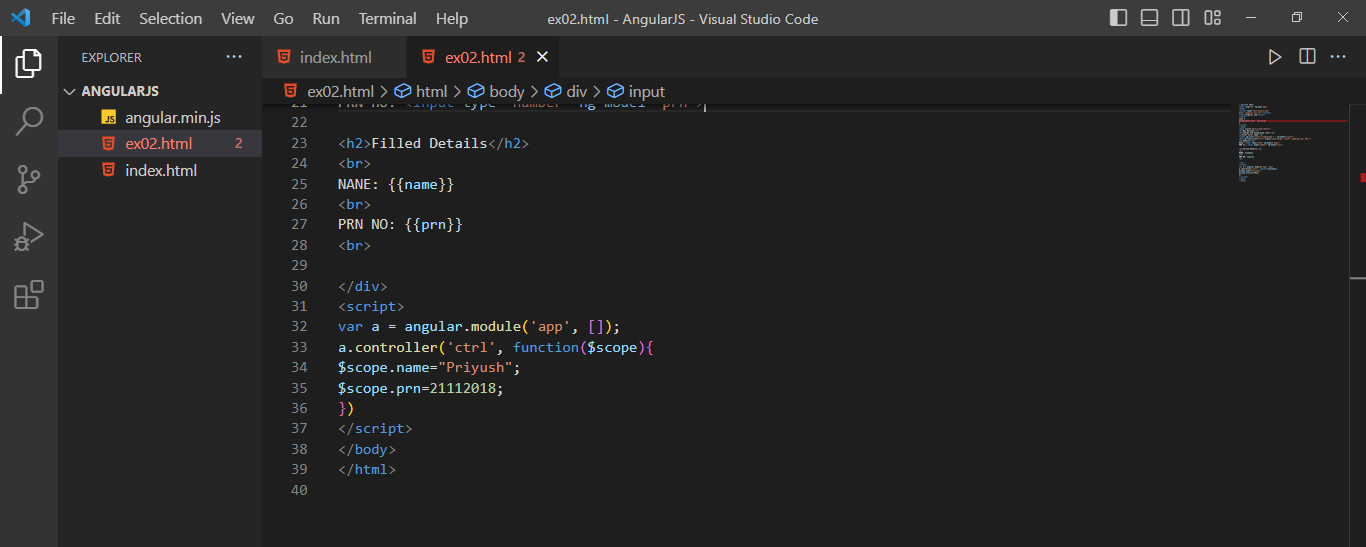
**Title:** To study and create ng-controller, ng-model, and expressions with examples

**Theory:**

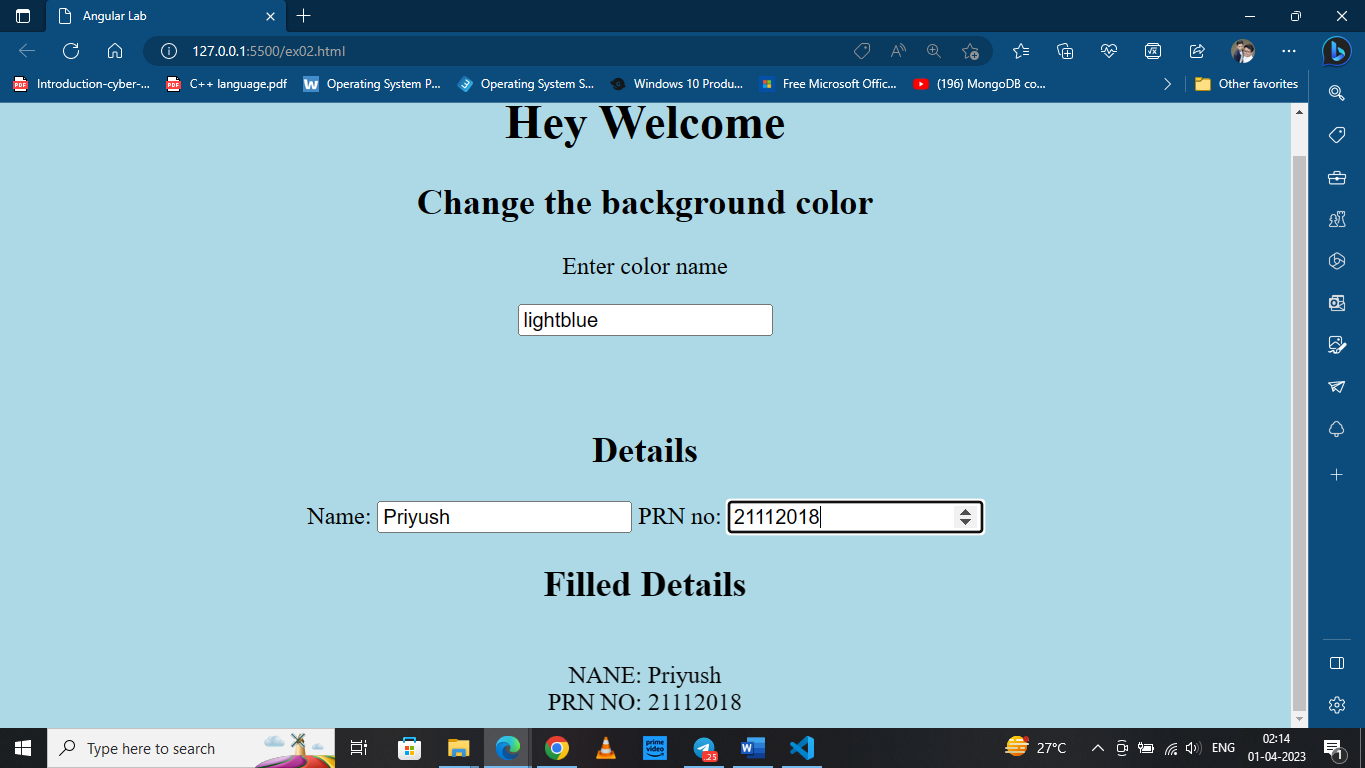
* **NG-CONTROLLER**: The ng-controller directive adds a controller to your application. In the controller, you can write code, and make functions and variables, which will be parts of an object, available inside the current HTML element. In AngularJS this object is called a scope.
* **NG-MODEL**: Ng-model is a directive in Angular. JS represents models and its primary purpose is to bind the “view” to the “model”. The ng-model directive binds the value of HTML controls (input, select, text area) to application data.
* **NG-APP:** The ng-app directive is a starting point for the AngularJS Application. It initializes the AngularJS framework automatically. AngularJS framework will first check for the ng-app directive in an HTML document after the entire document is loaded and if ng-app is found, it bootstraps itself and compiles the HTML template.

**Input:**





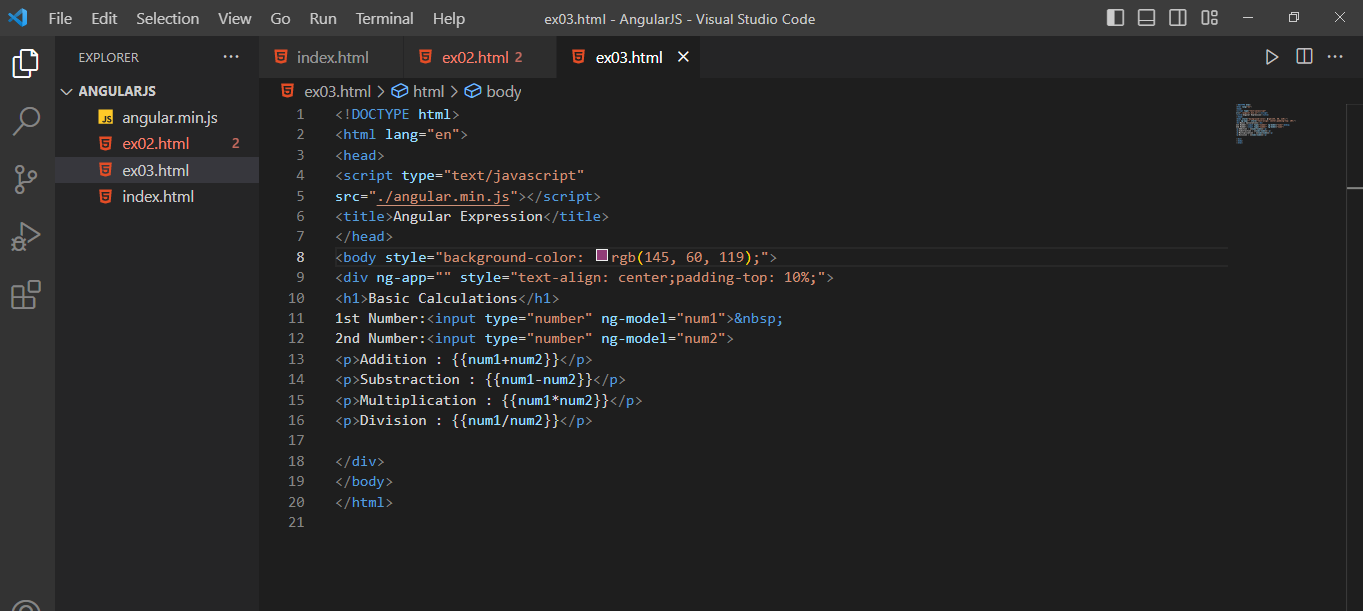
**Output:**



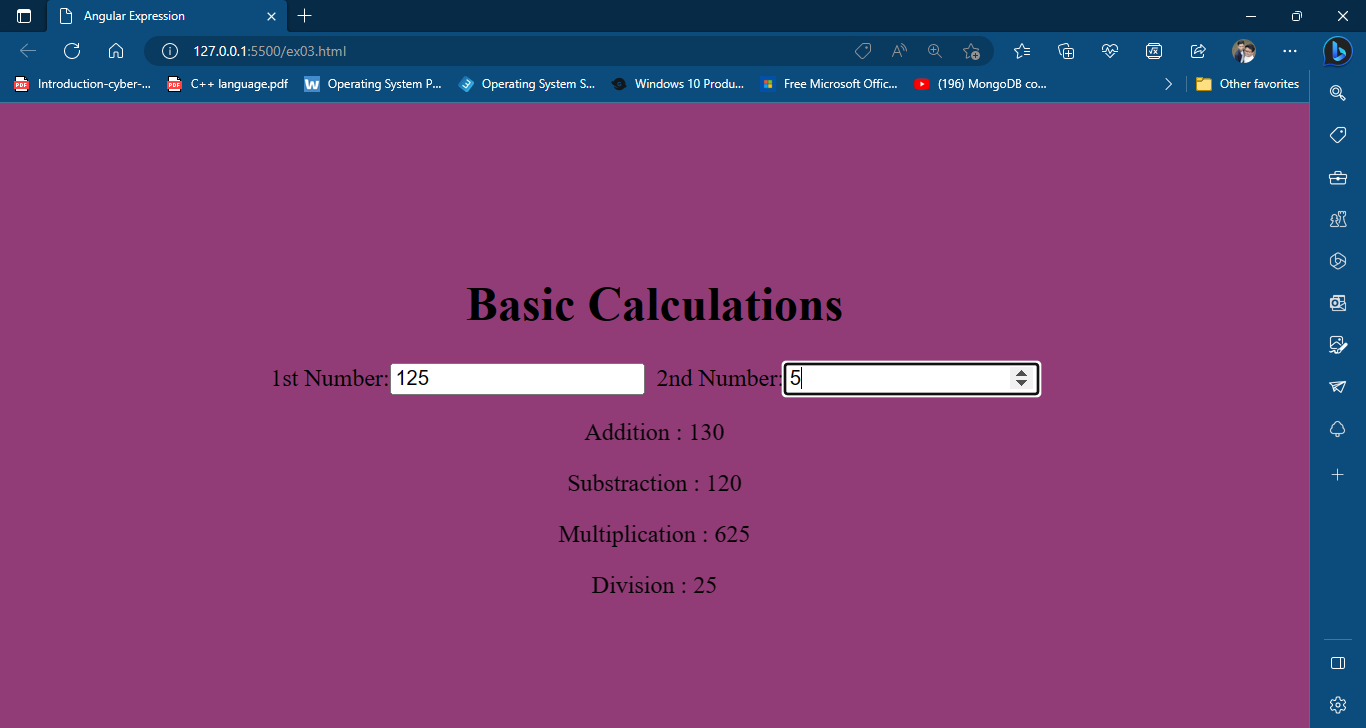
**🔹AngularJS Expressions:**

AngularJS binds data to HTML using Expressions. The expressions are resolved by Angular and the result is returned back to where the expression is written. The expressions in AngularJS are written in double braces: {{ expression }}.

**Input:**



**Output:**



**Conclusion: -** Thus we have implemented create ng-controller, ng-model, and expressions in AngularJS.

**DOP: / /2023 DOS: / /2023**

**Experiment No: 05**

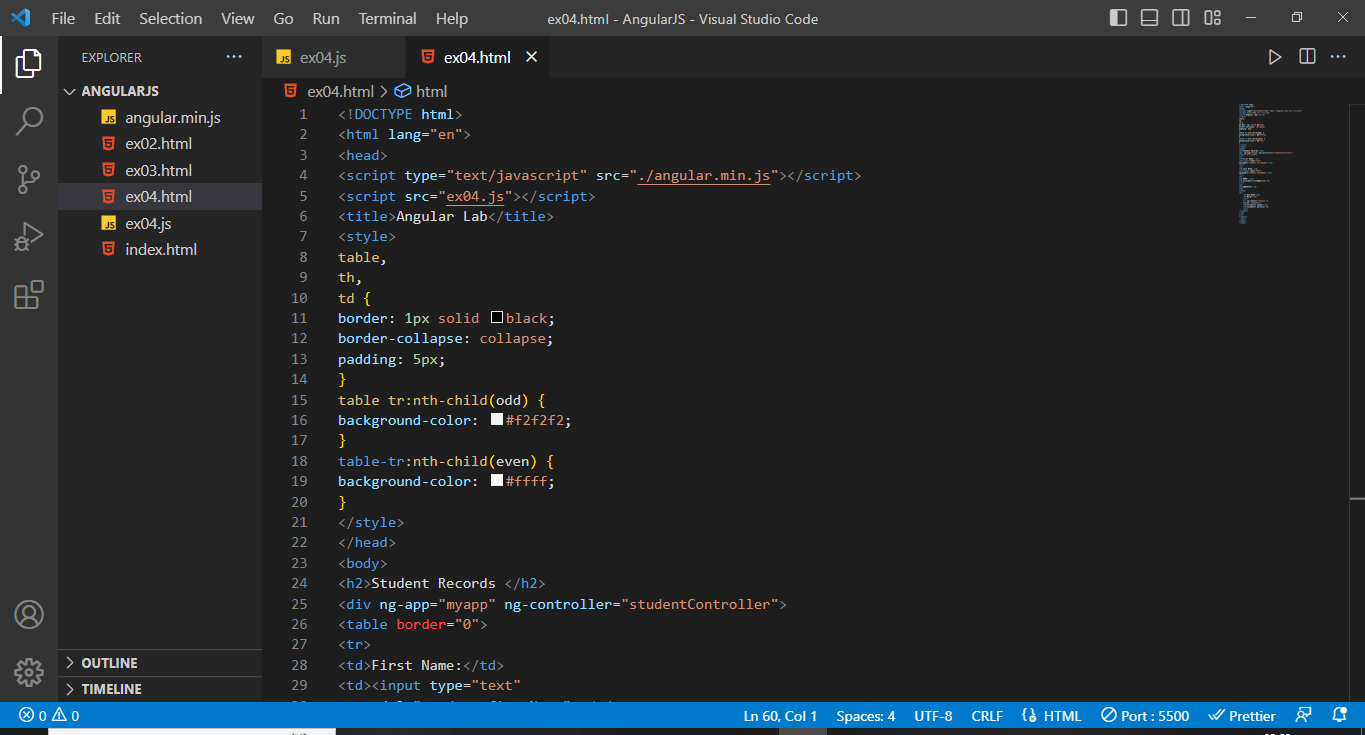
**Title:** Create an application for like student records using AngularJS.

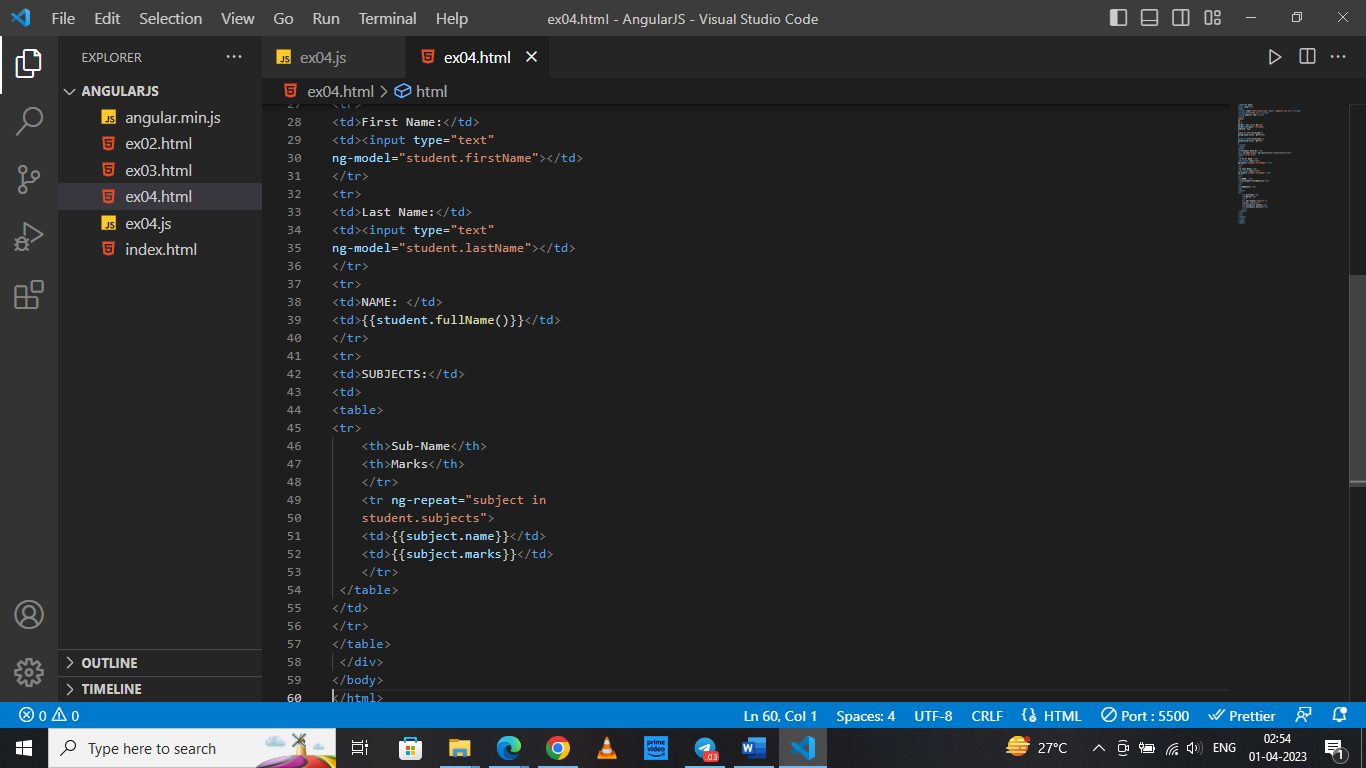
**Theory:**

* **NG-CONTROLLER**: The ng-controller directive adds a controller to your application. In the controller, you can write code, and make functions and variables, which will be parts of an object, available inside the current HTML element. In AngularJS this object is called a scope.
* **NG-MODEL**: Ng-model is a directive in Angular. JS represents models and its primary purpose is to bind the “view” to the “model”. The ng-model directive binds the value of HTML controls (input, select, text area) to application data.
* **NG-APP:** The ng-app directive is a starting point for the AngularJS Application. It initializes the AngularJS framework automatically. AngularJS framework will first check for the ng-app directive in an HTML document after the entire document is loaded and if ng-app is found, it bootstraps itself and compiles the HTML template.

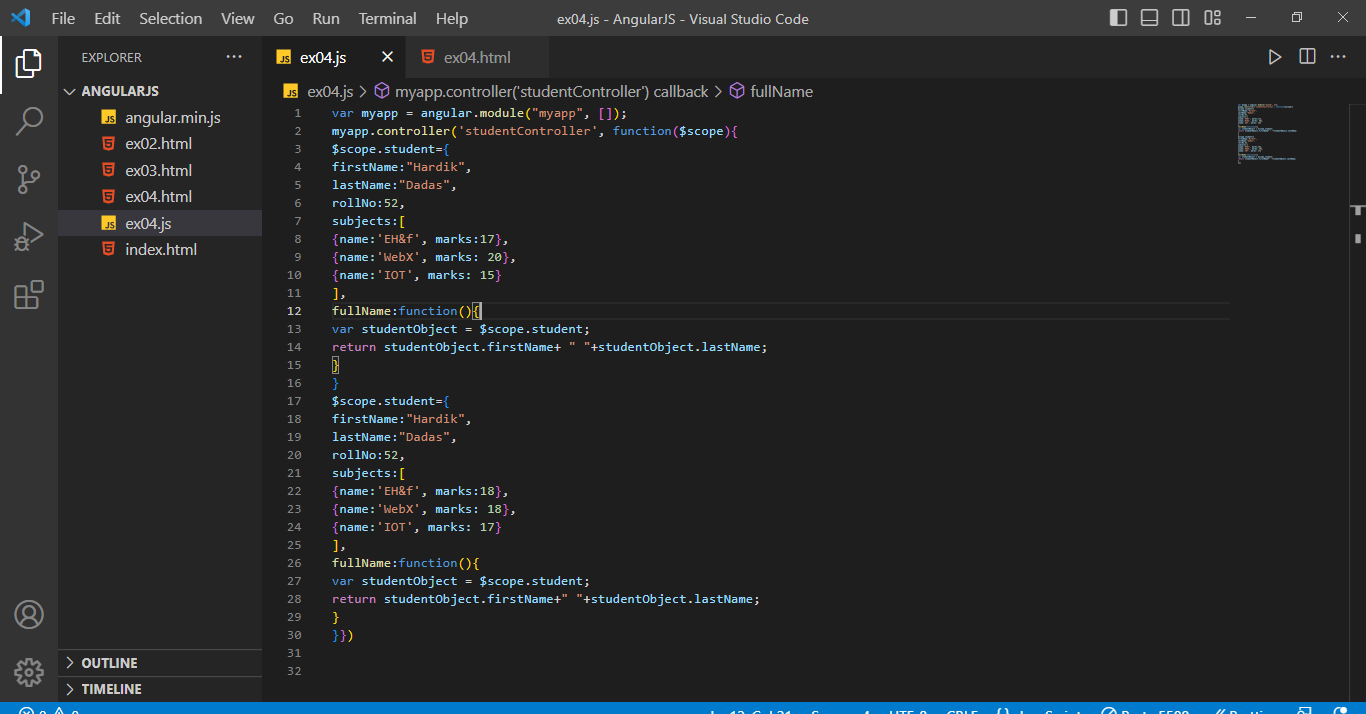
**Input:**

exp.html (file)

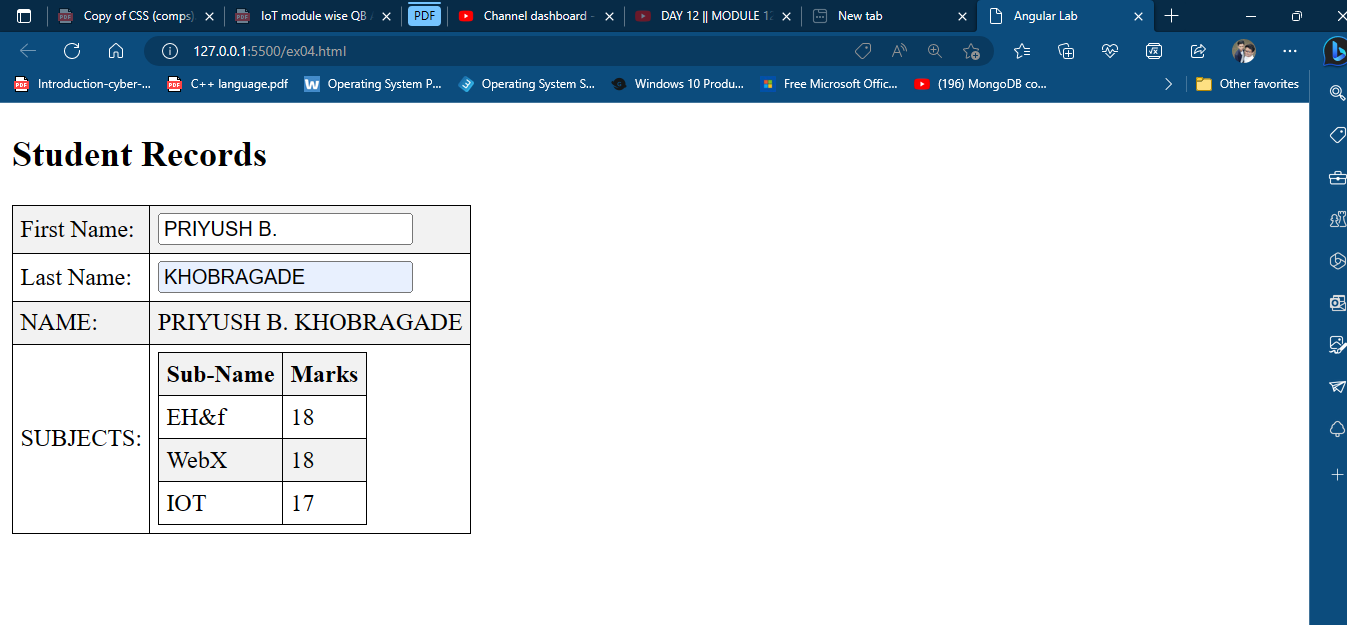




exp.js (file)



**Output:**



**Conclusion: -** Thus we have created an application for student records using AngularJs.

**DOP: / /2023 DOS: / /2023**

**Experiment No: 06**

**Title:** Build Crud Operation using MongoDB.

**Theory:**

**🔹What is MongoDB:**

* MongoDB is an open-source document database that provides high performance, high availability, and automatic scaling.
* In simple words, you can say that - Mongo DB is a document-oriented database. It is an open source product, developed and supported by a company named 10gen.
* MongoDB is available under General Public license for free, and it is also available under Commercial license from the manufacturer.
* The manufacturing company 10gen has defined MongoDB as:
* "MongoDB is a scalable, open source, high performance, document-oriented database." - 10gen

**🔹Features of MongoDB:**

These are some important features of MongoDB:

1. Support ad hoc queries: In MongoDB, you can search by field, range query and it also supports regular expression searches.

2. Indexing: You can index any field in a document.

3. Replication: MongoDB supports Master Slave replication.

4. Duplication of data: MongoDB can run over multiple servers. The data is duplicated to keep the system up and also keep its running condition in case of hardware failure.

5. Load balancing: It has an automatic load balancing configuration because of data placed in shards.

6. Supports map reduce and aggregation tools.

7. Uses JavaScript instead of Procedures.

8. It is a schema-less database written in C++.

9. Provides high performance.

10. Stores files of any size easily without complicating your stack.

11. Easy to administer in the case of failures.

**🔹CRUD operations**

CRUD (Create, Read, Update, Delete) operations allow you to work with the data stored in MongoDB. The CRUD operation documentation is categorized into two sections: Read Operations find and return documents stored within your MongoDB database. Write Operations to insert, modify, or delete documents in your MongoDB database.

**Create Operations –**

|  |  |
| --- | --- |
| **Method** | **Description** |
| **db.collection.insertOne()** | **It is used to insert a single document in the collection.** |
| **db.collection.insertMany()** | **It is used to insert multiple documents in the collection.** |
| **db.createCollection()** | **It is used to create an empty collection.** |

**Read Operations –**

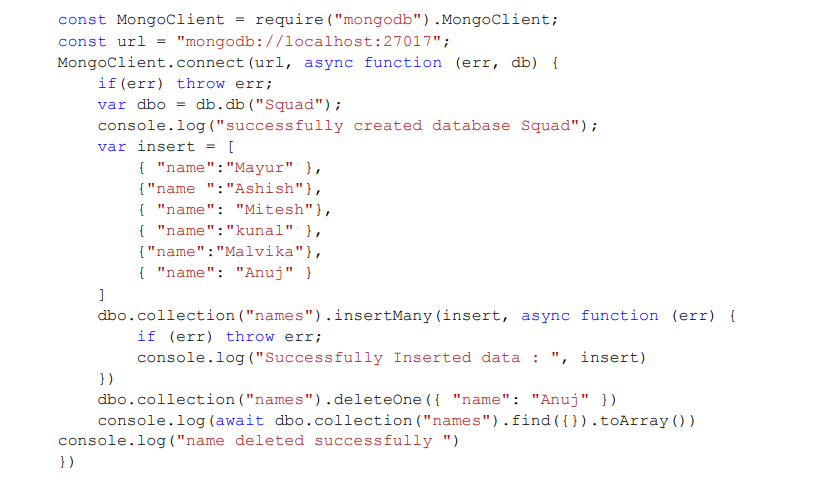
|  |  |
| --- | --- |
| **Method** | **Description** |
| **db.collection.find()** | **It is used to retrieve documents from the collection.** |

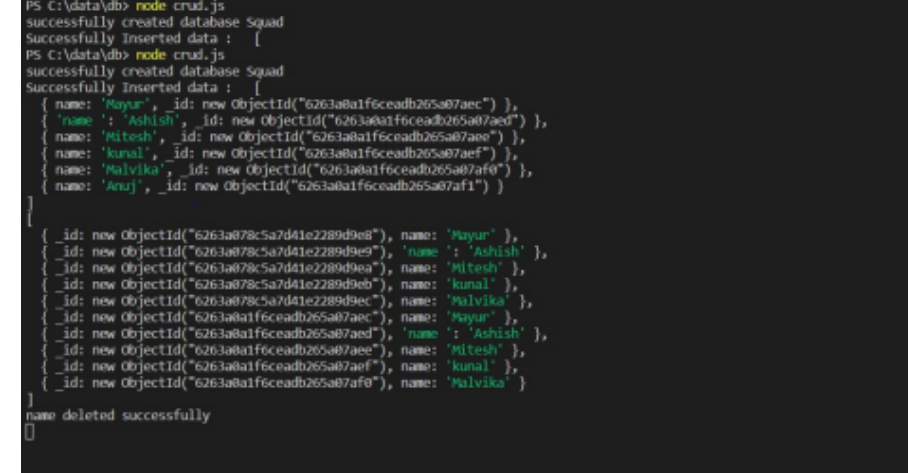
**Update Operations –**

|  |  |
| --- | --- |
| **Method** | **Description** |
| **db.collection.updateOne()** | **It is used to update a single document in the collection that satisfy the given criteria.** |
| **db.collection.updateMany()** | **It is used to update multiple documents in the collection that satisfy the given criteria.** |
| **db.collection.replaceOne()** | **It is used to replace single document in the collection that satisfy the given criteria** |

**Delete Operations –**

|  |  |
| --- | --- |
| **Method** | **Description** |
| **db.collection.deleteOne()** | **It is used to delete a single document from the collection that satisfy the given criteria.** |
| **db.collection.deleteMany()** | **It is used to delete multiple documents from the collection that satisfy the given criteria.** |





**Conclusion: -** In this program, we learn the crud operation i.e. (create, read, update, delete) in MongoDB using NodeJS.

**DOP: / /2023 DOS: / /2023**

**Experiment No: 07**

**Title** Build a REST-ful API(CRUD) using MongoDB.

**Theory:**

**🔹API:**

API is a set of definitions and protocols for building and integrating application software. It’s sometimes referred to as a contract between an information provider and an information user—establishing the content required from the consumer (the call) and the content required by the producer (the response).

Mongoose is an ODM (Object Document Mapping) tool for Node.js and MongoDB. It helps you convert the objects in your code to documents in the database and vice versa.

Before proceeding to the next section, Please install MongoDB in your machine if you have not done already. Checkout the official MogngoDB installation manual for any help with the installation.

Selecting the appropriate database for your REST API is crucial, and based on current trends, MongoDB is one of the most popular databases for web applications. MongoDB REST API is simple to set up and allows you to store and retrieve documents, making it great for Unstructured Data. Using Express JS as the backend web server with MongoDB as the document store is a common way of implementing the MongoDB REST API strategy. This approach conveniently links MongoDB’s Document Model to the JSON-based REST API payloads. You can use Express to build a backend middle tier that runs on Node.js and exposes REST API routes to your application. The Node.js Driver also connects the Express.js server to the MongoDB Atlas cluster.

**🔹Steps**:

**Step** 1: Setting up the Project If you have Node.js installed, you can run the following command to start the application from the command line:

npm init -y

The above command will create a package.json file.

**Step** 2: Installing Application Dependencies For MongoDB REST API to run, you need a file that will serve as the application’s command center. When you ask npm to run your application, it will initially run this file. This file can include object instances of both your modules and third-party modules installed from the npm directory. You created a file named app.js, which will be the application’s main entry point, and installed a few dependencies that are required to run your application using the commands above.

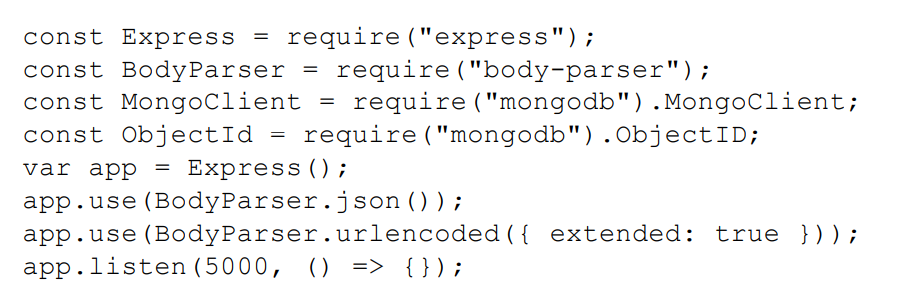
touch app.js

npm install express mongodb body-parser –save

**These are the dependencies:**

* Express: A framework for Node.js.
* MongoDB: The MongoDB team has supplied an official module to let your Node.js application communicate with MongoDB.
* Body-parser: It will handle request bodies with Express.

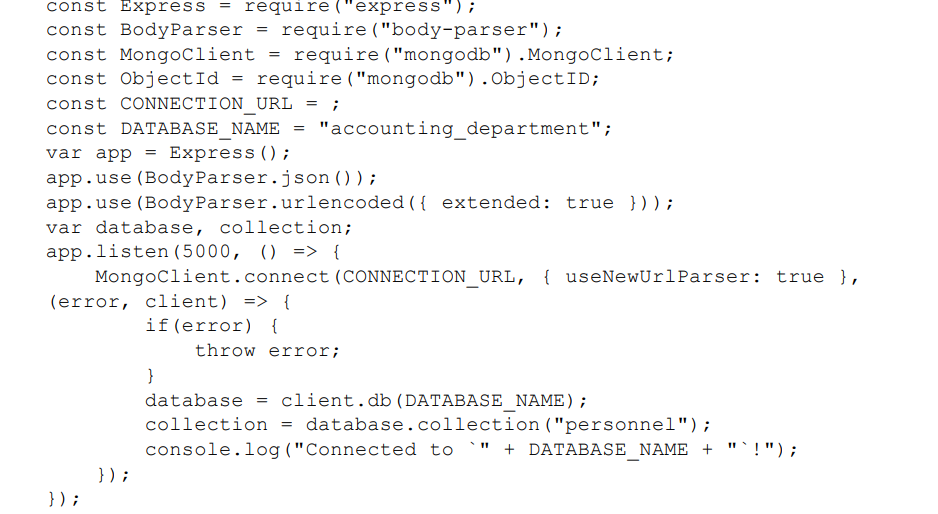
**Step** 3: Run Code Here, you are importing the dependencies you downloaded before. Use the Express object to initialize the express framework, which will utilize the express framework to start the server and run your application on a certain port, as well as configure the body-parser, which is a middleware that parses incoming chunks of data.



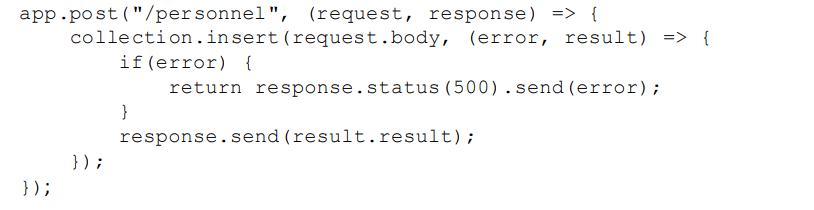
**Step** 4: Testing Application for MongoDB REST API

node app.js

**Step** 5: Establishing Connection with MongoDB REST API For this, you will need the MongoDB REST API connection string. Choose Clusters from the Atlas dashboard, then the Overview page, and then the Connect button. You will need to add the string to the app.js file and perform the following code adjustments.



**Step** 6: Build MongoDB REST API Endpoint Next, you will need to establish and query endpoints for the data. To add the data, we’ll need to construct an endpoint. To app.js, add the following code:



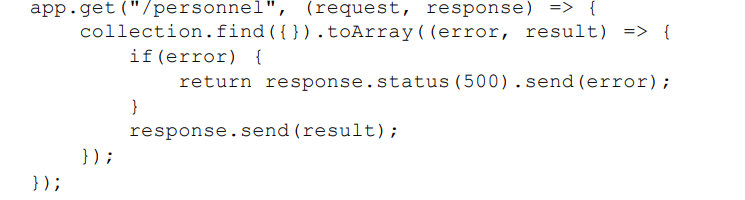
**Step** 7: Testing the MongoDB REST API



You’ll see the personnel record for John Doe added into the MongoDB database of ‘accounting department.’

**GET**

Now let’s create an endpoint to retrieve all the records data. Add the following code to app.js:



The goal here is to return all data in our collection representing people. We have no query conditions, hence the empty {} in the find command, and the results get converted into an array. Let’s test this out using cURL, a command-line tool for transferring data and supports HTTP; a very good ad-hoc tool for testing REST services.



**Conclusion: -** We learned how to create a REST-ful API using MongoDB.

**DOP: / /2023 DOS: / /2023**

**Experiment No: 10**

**Title:** A) Configuring Express Settings and creating Express application using request and response objects.

**Theory:**

**🔹What is Express.js:**

Express is a fast, assertive, essential and moderate web framework of Node.js. You can assume express as a layer built on the top of the Node.js that helps manage a server and routes. It provides a robust set of features to develop web and mobile applications.

Let's see some of the core features of Express framework:

* It can be used to design single-page, multi-page and hybrid web applications.
* It allows to setup middlewares 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.

**🔹Express.js Request Object**

Express.js Request and Response objects are the parameters of the callback function which is used in Express applications.

The express.js request object represents the HTTP request and has properties for the request query string, parameters, body, HTTP headers, and so on.

**Syntax:**

1. app.get('/', function (req, res) {
2. // --
3. })

**Request Object Methods:**

1. req.accepts (types)
2. req.get(field)
3. req.is(type)
4. req.param(name [, defaultValue])

**🔹Express.js Response Object:**

The Response object (res) specifies the HTTP response which is sent by an Express app when it gets an HTTP request.

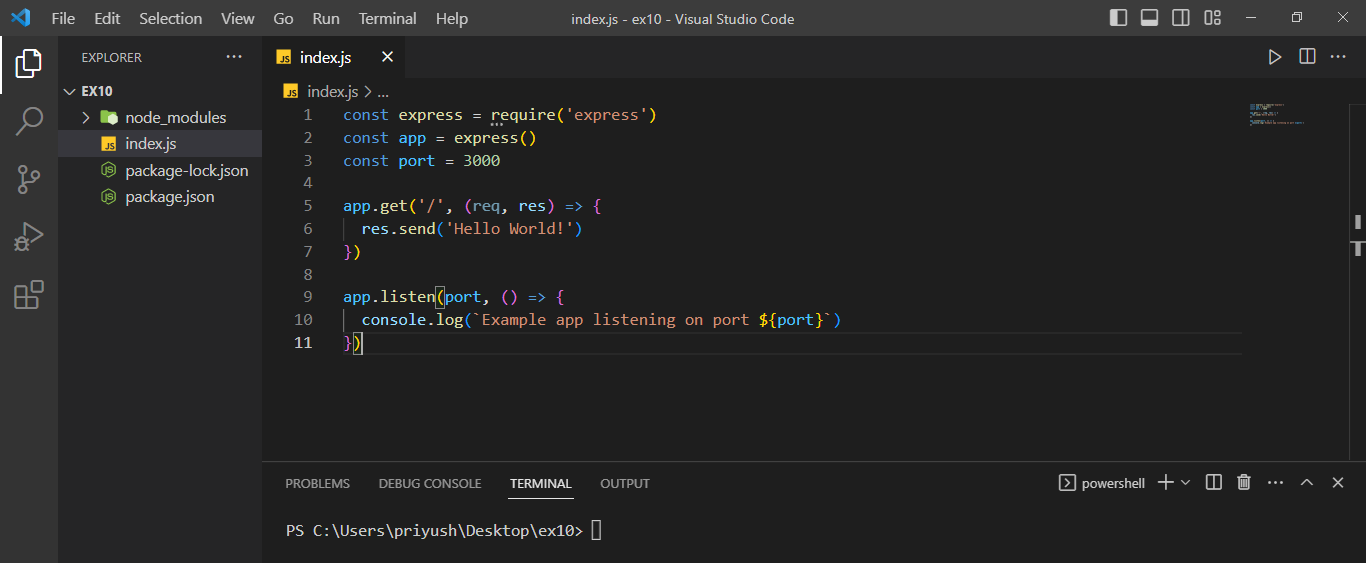
**What it does**

* It sends response back to the client browser.
* It facilitates you to put new cookies value and that will write to the client browser (under cross domain rule).
* Once you res.send() or res.redirect() or res.render(), you cannot do it again, otherwise, there will be uncaught error.

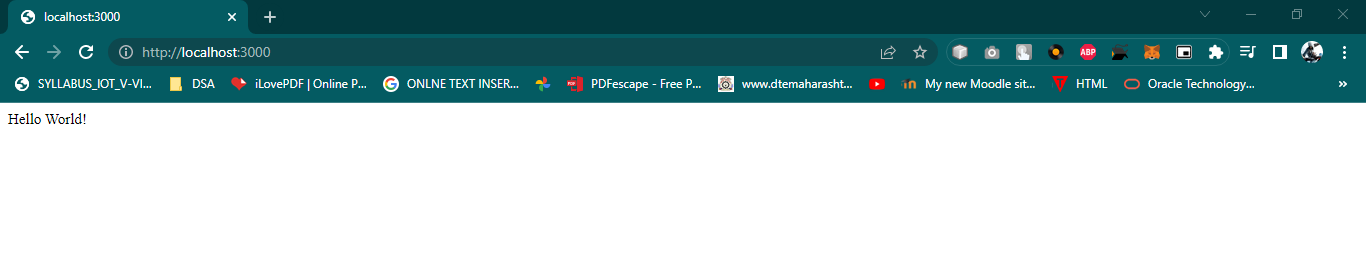
**Response Object Methods:**

1. Response Append method
2. Response Attachment method
3. Response Cookie method
4. Response ClearCookie method
5. Response Download method
6. Response End method
7. Response Format method
8. Response Get method
9. Response JSON method:

**Input:**



**Output:**



**Conclusion: -** we understanding Configuring Express Settings and creating Express application using request and response objects.

**DOP: / /2023 DOS: / /2023**

**Experiment No: 10**

**Title:**  B) Build Express application by Sending and Receiving Cookie

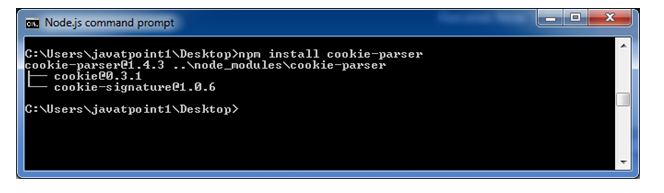
**🔹What are cookies**

Cookies are small piece of information i.e. sent from a website and stored in user's web browser when user browses that website. Every time the user loads that website back, the browser sends that stored data back to website or server, to recognize user.



**🔹Install cookie**

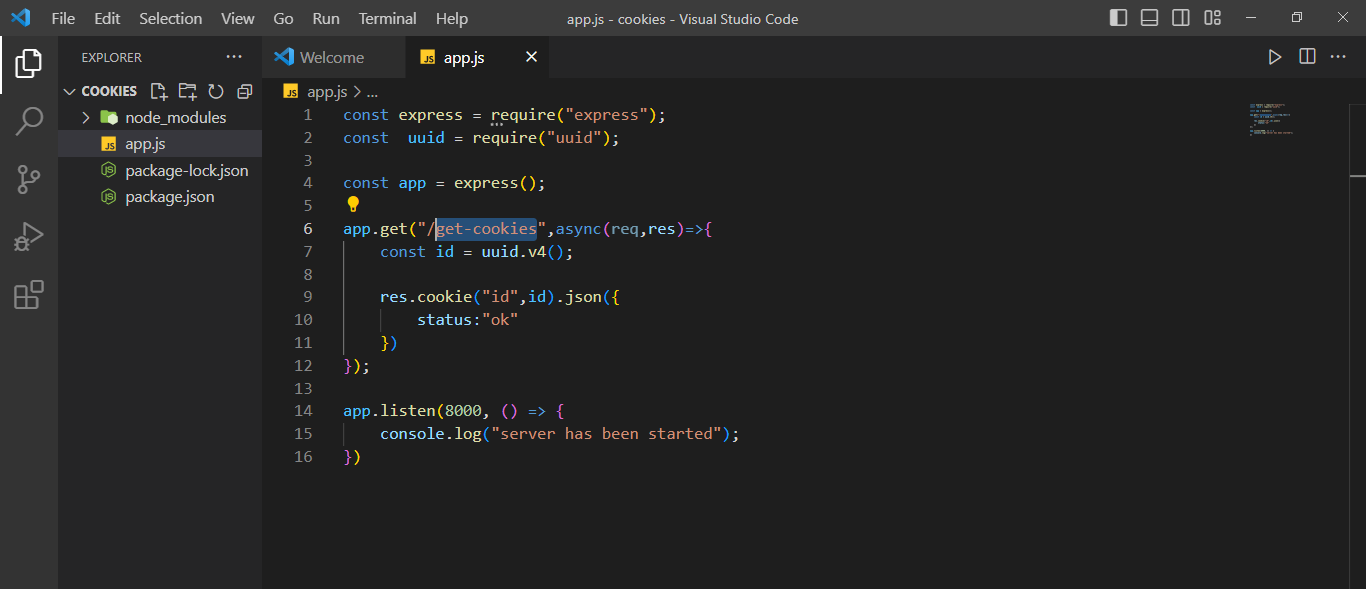
You have to acquire cookie abilities in Express.js. So, install cookie-parser middleware through npm by using the following command:



**Step**:

1. npm init –y
2. Define a route:Cookie-parser parses Cookie header and populate req.cookies with an object keyed by the cookie names.

**Input:**

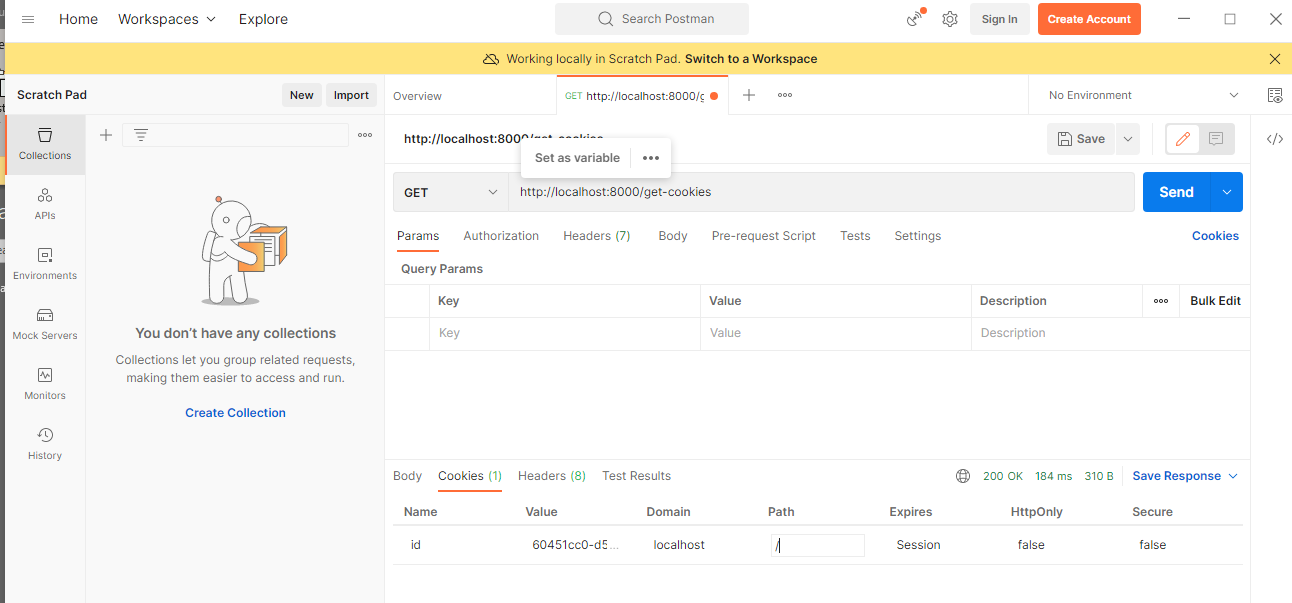


**Output:**

Open the page **http://localhost:8000/** on your browser:

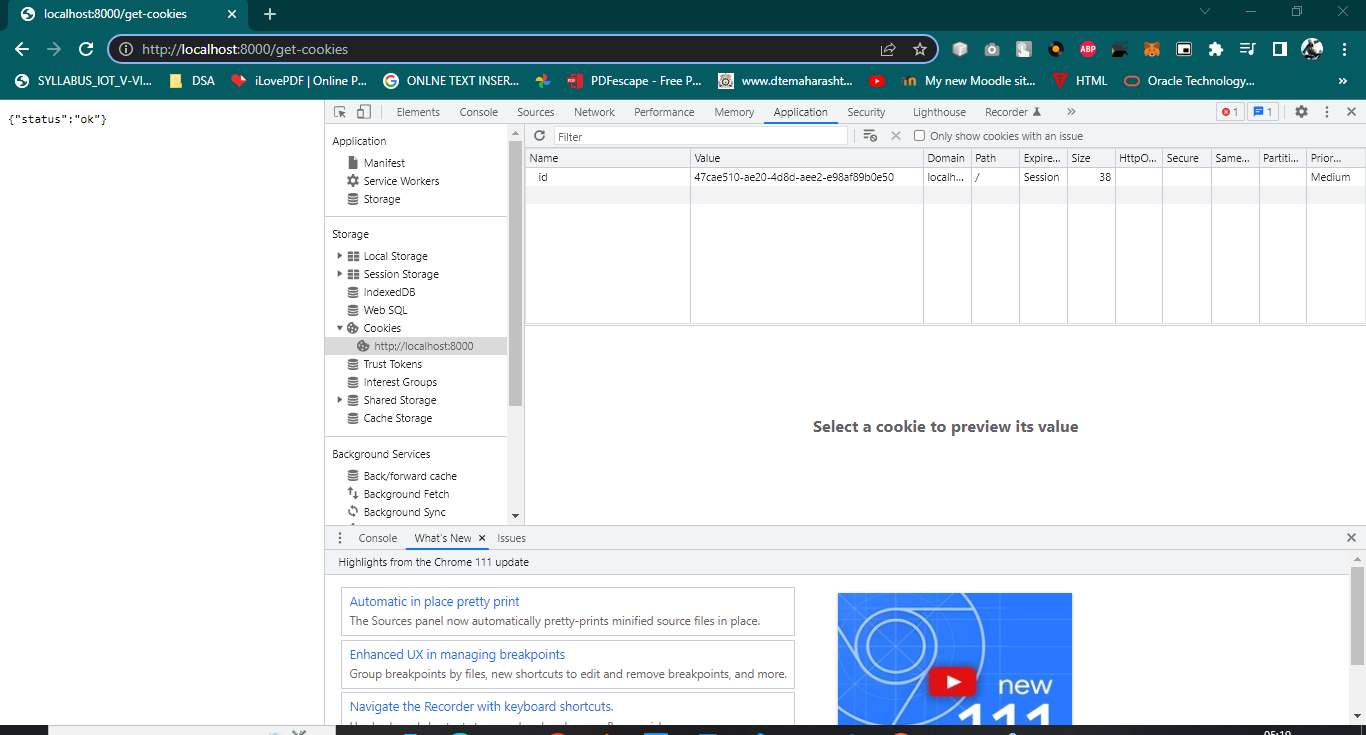
**Set cookie:**

Now open **http://localhost:8000/get-cookies** to set the cookie:



**Get cookie:**

Now open**http://localhost:8000/get-cookies** to get the cookie:



**Conclusion:** We implement the Build Express application by Sending and Receiving Cookie