

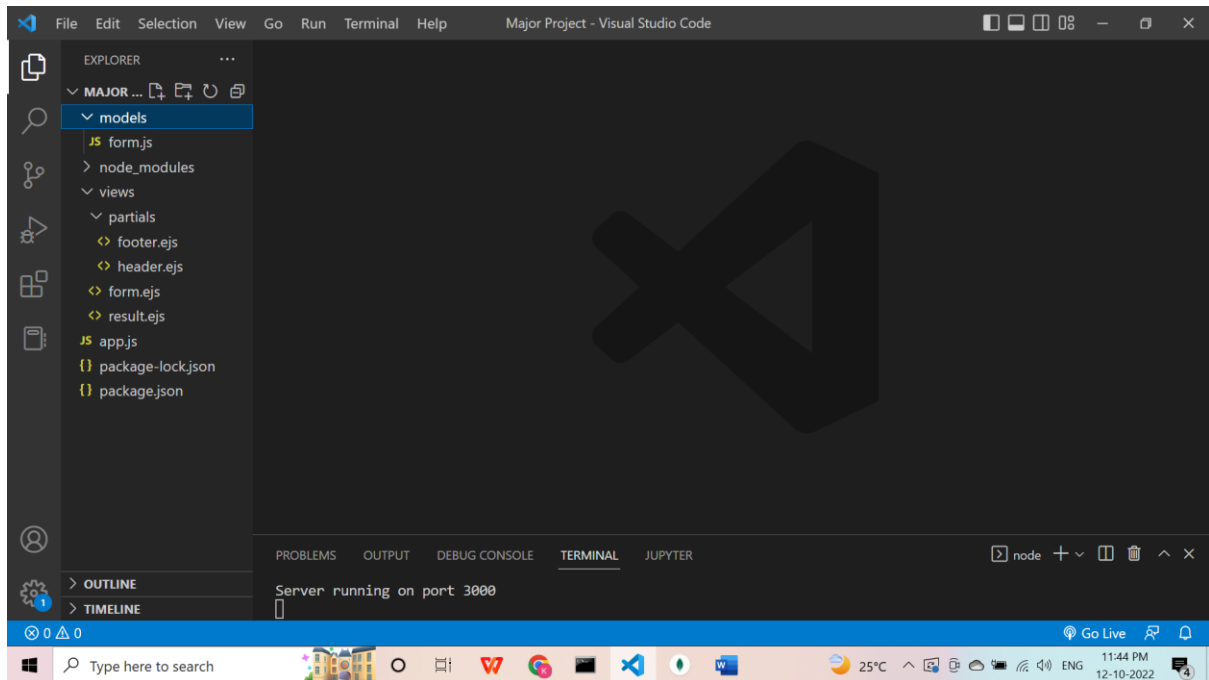
MAJOR PROJECT

MongoDB with Nodejs August Major Project

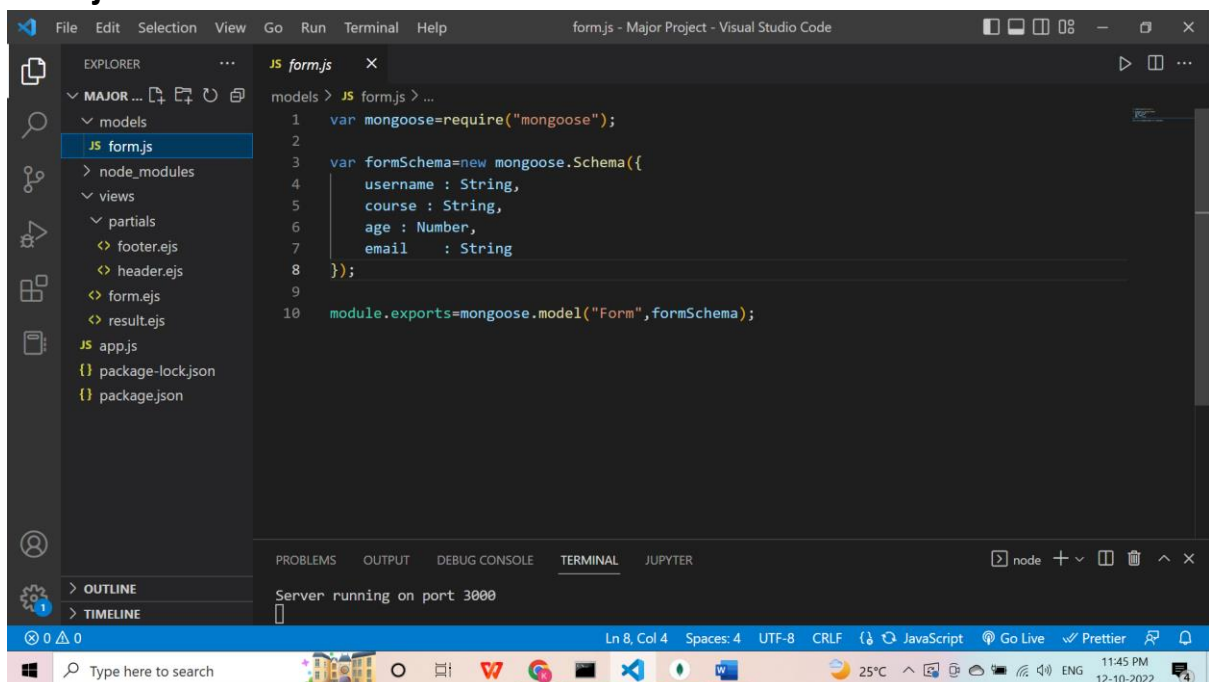
-Submitted by:ASHALI KISHORE

Email: ashalikishore@gmail.com

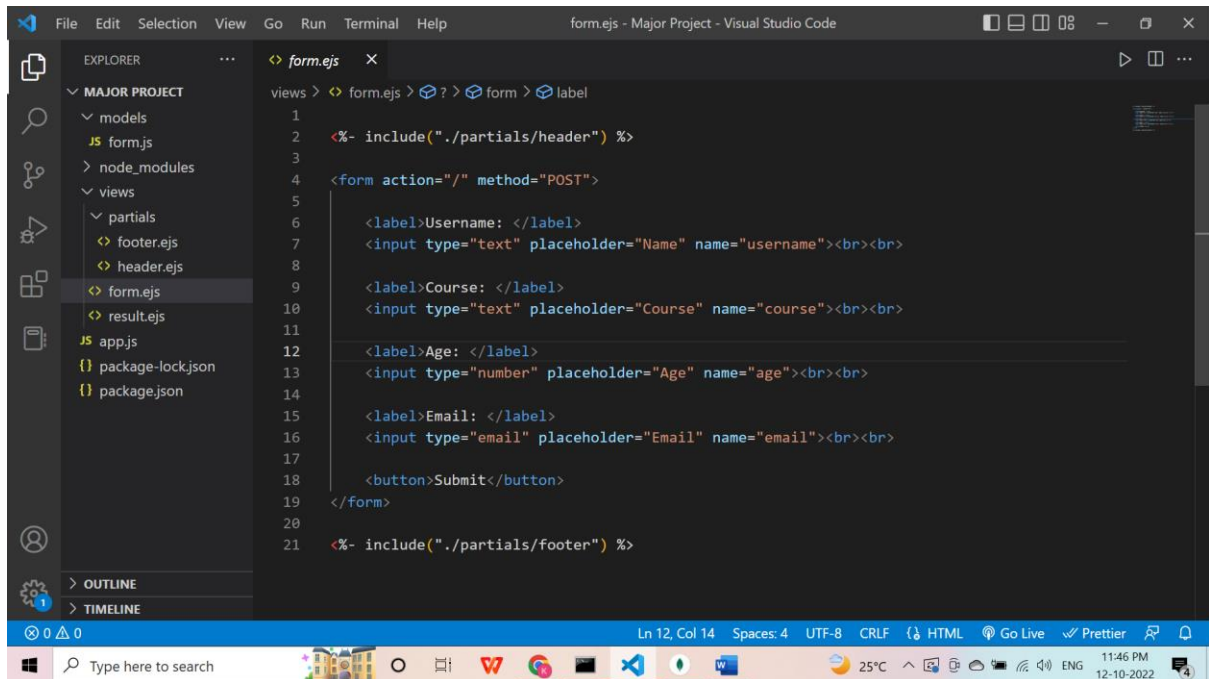
FILE STRUCTURE:



form.js:



form.ejs:



```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21<%= include("../partials/header") %>

<form action="/" method="POST">

  <label>Username: </label>
  <input type="text" placeholder="Name" name="username"><br><br>

  <label>Course: </label>
  <input type="text" placeholder="Course" name="course"><br><br>

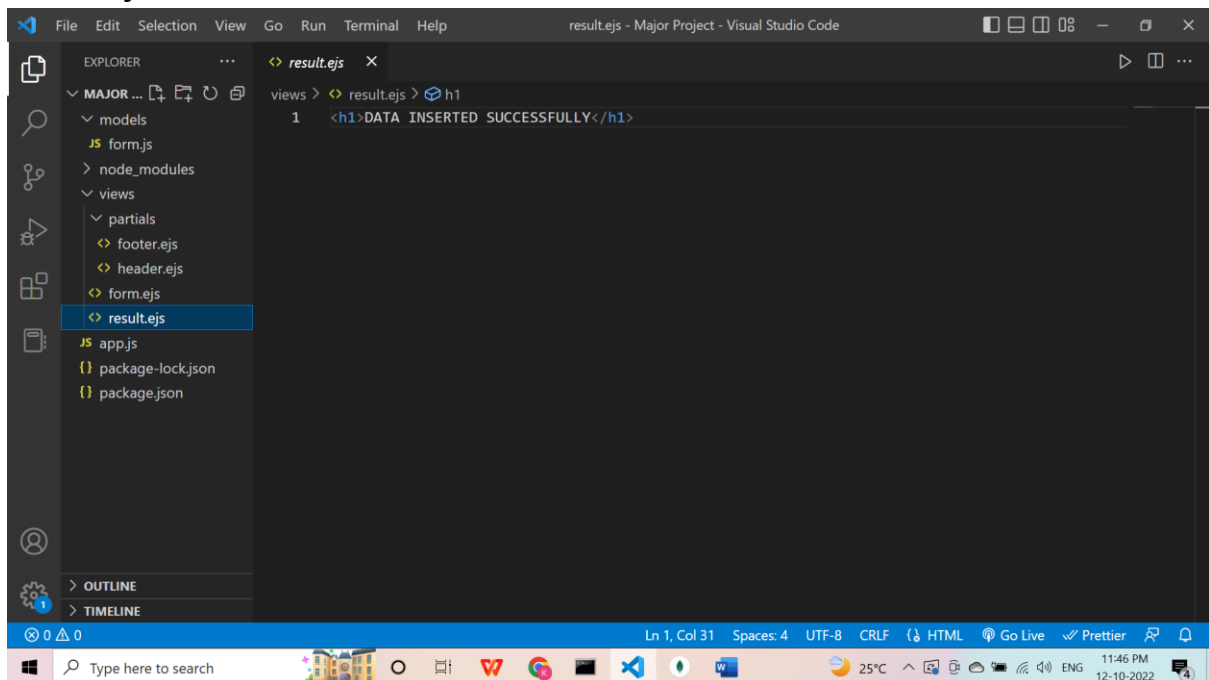
  <label>Age: </label>
  <input type="number" placeholder="Age" name="age"><br><br>

  <label>Email: </label>
  <input type="email" placeholder="Email" name="email"><br><br>

  <button>Submit</button>
</form>

<%= include("../partials/footer") %>
```

result.ejs:



```
1 <h1>DATA INSERTED SUCCESSFULLY</h1>
```

app.js:

The image displays two screenshots of a Visual Studio Code editor window, showing the development of an Express.js application. The top screenshot shows the initial setup of the app, and the bottom screenshot shows the addition of a POST endpoint for user registration.

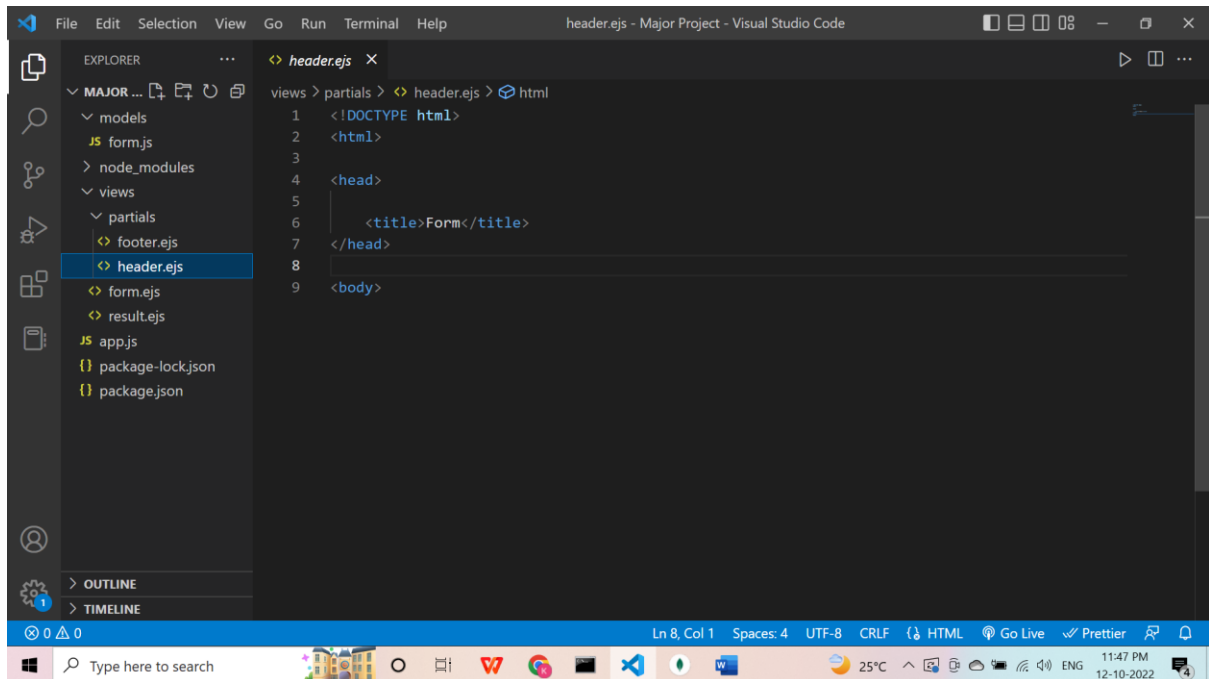
Top Screenshot:

```
1 const express = require("express");
2 const app=express();
3 var mongoose=require("mongoose");
4 var bodyParser=require("body-parser");
5
6 var Form=require("./models/form");
7
8 mongoose.connect("mongodb://localhost/form",{
9   useNewUrlParser: true,
10  useUnifiedTopology: true
11 });
12
13
14 app.set('view engine','ejs');
15 app.use(bodyParser.urlencoded({extended:true}));
16
17 app.get("/",function(req,res){
18   res.render("form");
19 });
20
21 app.get('/result',(req,res)=>{
22   res.render('result');
23 });
24
```

Bottom Screenshot:

```
25 app.post("/",function(req,res){
26   var username=req.body.username;
27   var course=req.body.course;
28   var age=req.body.age;
29   var email=req.body.email;
30   var f={username: username,email:email, course:course, age:age};
31   Form.create(f,function(err,newlyCreatedForm){
32     if(err)
33     {
34       console.log(err);
35     }
36     else{
37       res.redirect("/result");
38     }
39   });
40 });
41
42 app.listen(3000, function() {
43   console.log('Server running on port 3000');
44 });
```

header.js:

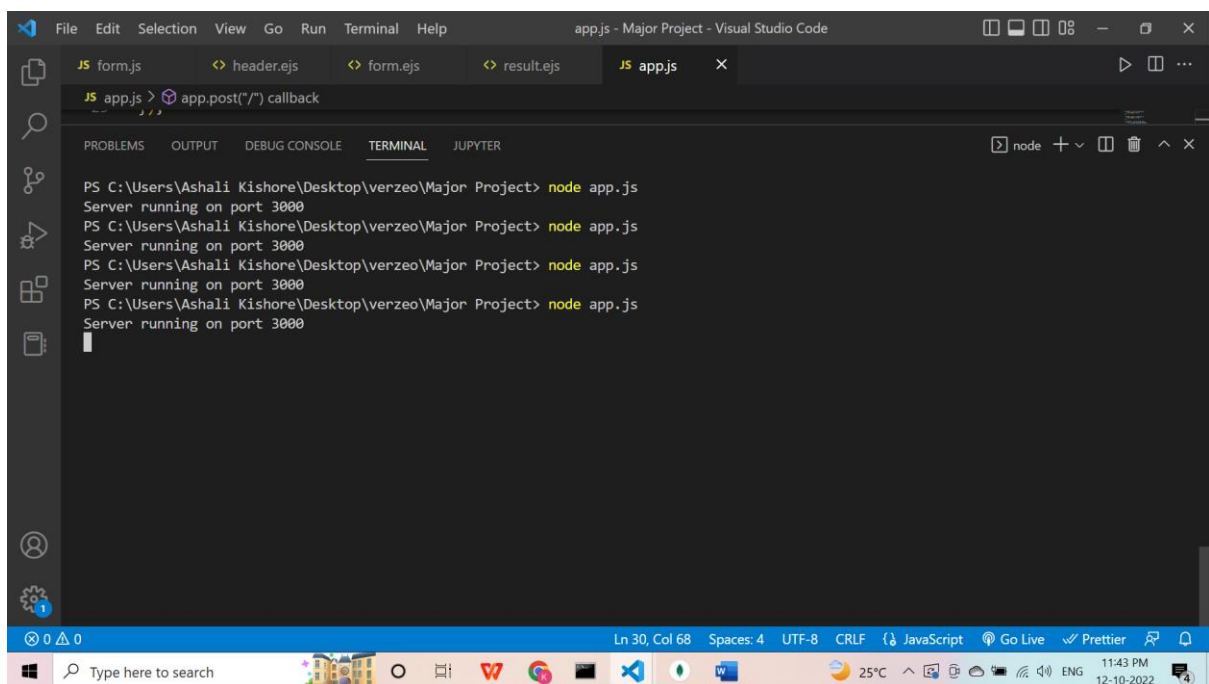


The screenshot shows the Visual Studio Code interface with the 'header.ejs' file open in the Editor. The Explorer view on the left shows the project structure, including 'MAJOR ...', 'models', 'node_modules', 'views', 'partials', 'footer.ejs', 'header.ejs', 'form.ejs', 'result.ejs', 'app.js', 'package-lock.json', and 'package.json'. The Editor view shows the content of 'header.ejs' with the following code:

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5
6   <title>Form</title>
7 </head>
8
9 <body>
```

The status bar at the bottom indicates the file is at Line 8, Column 1, with 4 spaces, UTF-8 encoding, and CRLF line endings. The system tray shows the date and time as 11:47 PM on 12-10-2022.

TERMINAL:



The screenshot shows the Visual Studio Code interface with the 'app.js' file open in the Editor. The Explorer view on the left shows the project structure, including 'form.js', 'header.ejs', 'form.ejs', 'result.ejs', 'app.js', 'package-lock.json', and 'package.json'. The Editor view shows the content of 'app.js' with the following code:

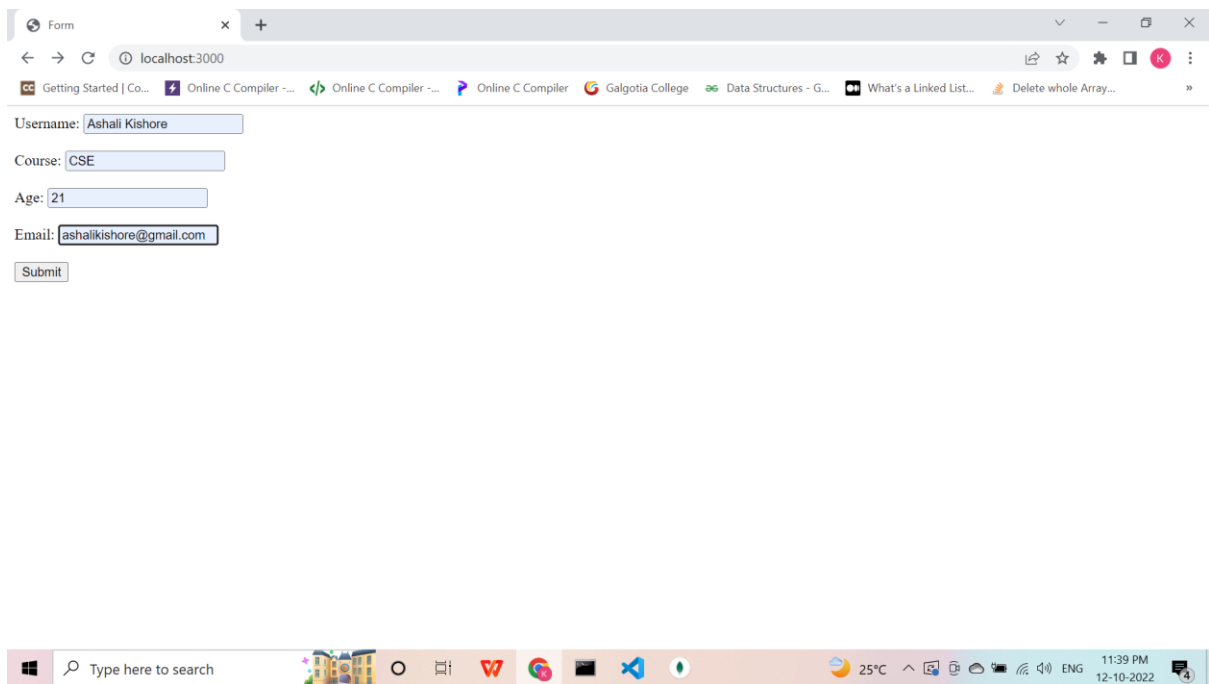
```
1 app.post("/") callback
```

The Terminal view at the bottom shows the output of running the 'node app.js' command. The output is:

```
PS C:\Users\Ashali Kishore\Desktop\verzeo\Major Project> node app.js
Server running on port 3000
PS C:\Users\Ashali Kishore\Desktop\verzeo\Major Project> node app.js
Server running on port 3000
PS C:\Users\Ashali Kishore\Desktop\verzeo\Major Project> node app.js
Server running on port 3000
PS C:\Users\Ashali Kishore\Desktop\verzeo\Major Project> node app.js
Server running on port 3000
```

The status bar at the bottom indicates the file is at Line 30, Column 68, with 4 spaces, UTF-8 encoding, and CRLF line endings. The system tray shows the date and time as 11:43 PM on 12-10-2022.

LOCAL HOST:



Form

localhost:3000

Getting Started | Co... Online C Compiler ~... Online C Compiler ~... Online C Compiler Galgotia College Data Structures - G... What's a Linked List... Delete whole Array...

Username:

Course:

Age:

Email:

Type here to search

25°C 11:39 PM 12-10-2022

localhost:3000/result

localhost:3000/result

Getting Started | Co... Online C Compiler ~... Online C Compiler ~... Online C Compiler Galgotia College Data Structures - G... What's a Linked List... Delete whole Array...

DATA INSERTED SUCCESSFULLY



UPDATED DATABASE(MongoDB):

The screenshot displays the MongoDB Compass application window. The title bar reads "MongoDB Compass - localhost:27017/form.forms". The interface is divided into several sections:

- Left Sidebar:** Shows the connection "localhost:27017" with 5 DBs and 6 Collections. The "form" database is expanded, showing collections "local", "test7", and "col7". The "forms" collection is selected.
- Top Bar:** Includes "Connect", "View", "Collection", and "Help" menus. The "Documents" tab is active for the "form.forms" collection. It shows 1 Document and 1 Index.
- Filter Bar:** Contains a filter input with the text "{ field: 'value' }", an "OPTIONS" button, a green "FIND" button, a "RESET" button, and a refresh icon.
- Document View:** Displays a single document in JSON format:

```
{ "_id": ObjectId('634702ff2e4d6673ea885981'), "username": "Ashali Kishore", "course": "CSE", "age": 21, "email": "ashalikishore@gmail.com", "_v": 0 }
```
- Bottom Bar:** Shows the status "MONGOSH" and a search bar.

The Windows taskbar at the bottom indicates the system time is 11:40 PM on 12-10-2022, with a temperature of 25°C.