

MVC Stack

- Open the project folder you have created last week and Add following directories(folders)
 - public
 - models
 - controllers
 - views
 - routes

Required Packages

open the terminal and install the following packages use the below command

```
npm install express express-ejs-layouts express-fileupload express-session mongodb  
mongoose ejs dotenv cookie-parser connect-flash
```

Creating the database connection

- Create a file inside models name it as database.js got to your index.js and cut the following code and paste it into the database.js and save the file

```
// Database connection  
mongoose.connect(process.env.DATABASE_URI,{  
  useUnifiedTopology:true,  
  useNewUrlParser:true  
})  
.then(() =>{console.log('Database Connected')})  
.catch ((err)=>{  
  console.log(`database not connected ->${err.message}`)  
})
```

- Add the following line to the top of the database.js

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

Creating the Model

- Create a file inside models name it as studentModel.js copy the following code from index.js

```
//Schema  
const mongoose = require('mongoose')// additional line to import the mongoose
```

```
const studentSchema = new Schema({
  name:String,
  age:Number,
  address:String
});

module.exports=mongoose.model('Student', studentSchema);// make this module
importable in other modules
```

- add the following line into the database.js at the bottom to import the model that we created just now

```
require('./studentModel') //
```

- your database.js will look like this

```
const mongoose = require('mongoose');
mongoose.connect(process.env.DATABASE_URI,{
  useUnifiedTopology:true,
  useNewUrlParser:true
})
.then(() =>{console.log('database connected')})
.catch ((err)=>{
  console.log(`database not connected ->${err.message}`)
})

require('./studentModel')
```

Creating the Controllers

- Go to your controllers folder and create a file name with studentController.js and add the following code 1- Import the database model 2- Import the student model 3- Write the controller code and export it using **exports**

```
require ('../models/database')//importing the database.js
const Student=require('../models/studentModel')//importing the
exports.HomePage=(req, res)=>{
  res.render('index');
```

Creating the Routes

go to your routes folder and create a file name with studentRoutes.js and add the following code

```
const express = require('express');
const router=express.Router()

const studentController=require('../controllers/studentController')

router.get('/',studentController.HomePage);

module.exports=router
```

- replace your index.js with the following code,

```
require('dotenv').config()
const express = require('express');
const app = express()
const expressLayouts=require('express-ejs-layouts')
var port=3000

app.use(express.urlencoded({extended:true}))//passing data in url(hidden)
app.use(express.static('public'))//path to the static files(images,css,js)
app.use(expressLayouts)//for using templates such as header ,footer ,body

// server listen
app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
---
```

Set tamplate and view engines

- add the following to the index.js before the **//server listen**

```
app.set ('layout','./layouts/main')//create layouts/main inside views
app.set('view engine','ejs')
```

Creating the template of your application/website

- Go to the views and create a folder called layouts, inside the layouts folder create a file called main.ejs
- To add the html hold (shift ! and tab) and add this to the main.ejs

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <main>
    <%- body -%>
  </main>
</body>
</html>
```

- Go to views and create index.ejs,(this will be your homepage) add this code and test it on the browser localhost:3000

```
<h1>hello from ejb world</h1>
```

- create following file inside **views directory**
 - create_student.ejs

Adding bootstrap to your layout

- If you want to use bootstrap go to bootstrap website and add **CSS** and **bundleJS Bootstrap**
- Add bootstrap css and js in **views/layouts/main.ejs** file your file should look like this.

```
<!-- FILE: views/layouts/main.ejs -->
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!--bootstrap css -->
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhF1dvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
  <title>Document</title>
</head>
<!--Navigation Bar will go here -->
<body>
  <div class="container">
```

```

    <main>
      <%- body -%>

    </main>
  </div>
  <!--bootstrap js bundle-->
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYs0g+OMhuP+IlRH9sENB00LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
</body>
</html>

```

Adding the Navigation Bar before (tag) for your main.ejs page

```

<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="/student">Students</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-
expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
    <div class="navbar-nav">
      <a class="nav-link active" href="/">Home <span class="sr-only">(current)
</span></a>
      <a class="nav-link" href="/create_student">Create Student</a>
    </div>
  </div>
</nav>

```

Process of creating mvc page

Add Student (create_student page)

- open create_student.ejs and add following code or get a form from bootstrap [Form](#)

```

<h2>Add new Student</h2>
<form method="POST" action="create_student">
  <div class="form-group">
    <label for="name">Student Name</label>
    <input type="text" class="form-control" name="name">
  </div>
  <div class="form-group">
    <label for="email">Email address</label>
    <input type="email" class="form-control" name="email">
  </div>
  <button class="btn btn-primary">Add Student</button>

```

```
</form>
```

Add route (Going from one webpage to the other webpage)

- Add following code in **studentRoutes.js** before **module.exports=router**

```
// File: routes/studentRoutes.js
router.get('/create_student', studentController.CreatePage);
```

Add controller

- Add following code in **studentController.js**

```
// File: controllers/studentController.js
// create form view
exports.CreatePage=(req, res)=>{
  res.render('create_student');
}
```

- Go to navigation bar in your app [localhost](#) click create link you will see **add new student** form

Create **POST** Process

To add data in to the database we need to post data. To post data we will add the following to our code.

- create post route
- add controller
- store data to databse

create post router

add following code in **studentRoutes.js**

```
// File: routes/studentRoutes.js
router.post('/create_student', studentController.CreateStudent);
```

Add controller

Add following code in **studentController.js**

```
// File: controllers/studentController.js
// submit form (store data in database)
exports.CreateStudent=(req, res)=>{
```

```
    console.log(req.body);  
  }
```

- your studentController file will look like this

```
require ('../models/database')  
const Student=require('../models/studentModel')  
  
exports.HomePage= async (req, res)=>{  
  
    const students = await Student.find({})  
  
    res.render('index',{students});  
  
}  
// create form view  
exports.CreatePage=(req, res)=>{  
    res.render('create_student');  
}  
// submit form (store data in database)  
exports.CreateStudent=(req, res)=>{  
    console.log(req.body);//comment this in the next step  
}
```

- If you add student using create_student form you will be able to see the data as below on your terminal

```
database connected  
{ name: 'chat', email: 'abc@live.com' }
```

Adding Data to the database

- Add following code in **studentController.js** after exports.CreateStudent=(req, res)=>{

```
// File: controllers/studentController.js  
  
let name =req.body.name  
let email =req.body.email  
if(email !==''&& name !==''){  
    const student = new Student({  
        name:name,  
        email:email  
    })  
    student.save()  
}else{
```

```
}
console.log('student data created')
res.redirect('/')
}
```

Display data from the MongoDB to a webpage

- Add following code in **index.ejs** bootstrap table

```
<% if (typeof students !== 'undefined' && students.length > 0) { %>
<table class="table">
  <thead>
    <tr>
      <th scope="col">#</th>
      <th scope="col">Name</th>
      <th scope="col">Email</th>

    </tr>
  </thead>
  <tbody>
    <% students.forEach( function(student, index){ %>
    <tr>
      <th scope="row"><%- index+1 %></th>
      <td><%- student.name %></td>
      <td><%- student.email %></td>

    </tr>
    <% }) %>
  </tbody>
</table>
<% } %>
```