BRAC
UNIVERSITY

Inspiring Excellence

CSE470: Software Engineering

Section: 5

Group: 3

SPRINT 2

Sadat Mahmud	22301301	
Rana Mustafa	21101060	*(Scrum Master)*
Atoshi Samadder	21301706	
Nabil Nashit	21201060	

Software: VisiNexus – Connecting insights through vision

Requirements

1. Real-time demos for computer vision applications such as object detection, emotion recognition, etc.
2. Interactive tools for learning and understanding computer vision applications
3. Documentation and tutorials
4. API access and integration guide
5. User Management
6. Virtual coins for purchasing accessibility

Framework Setup

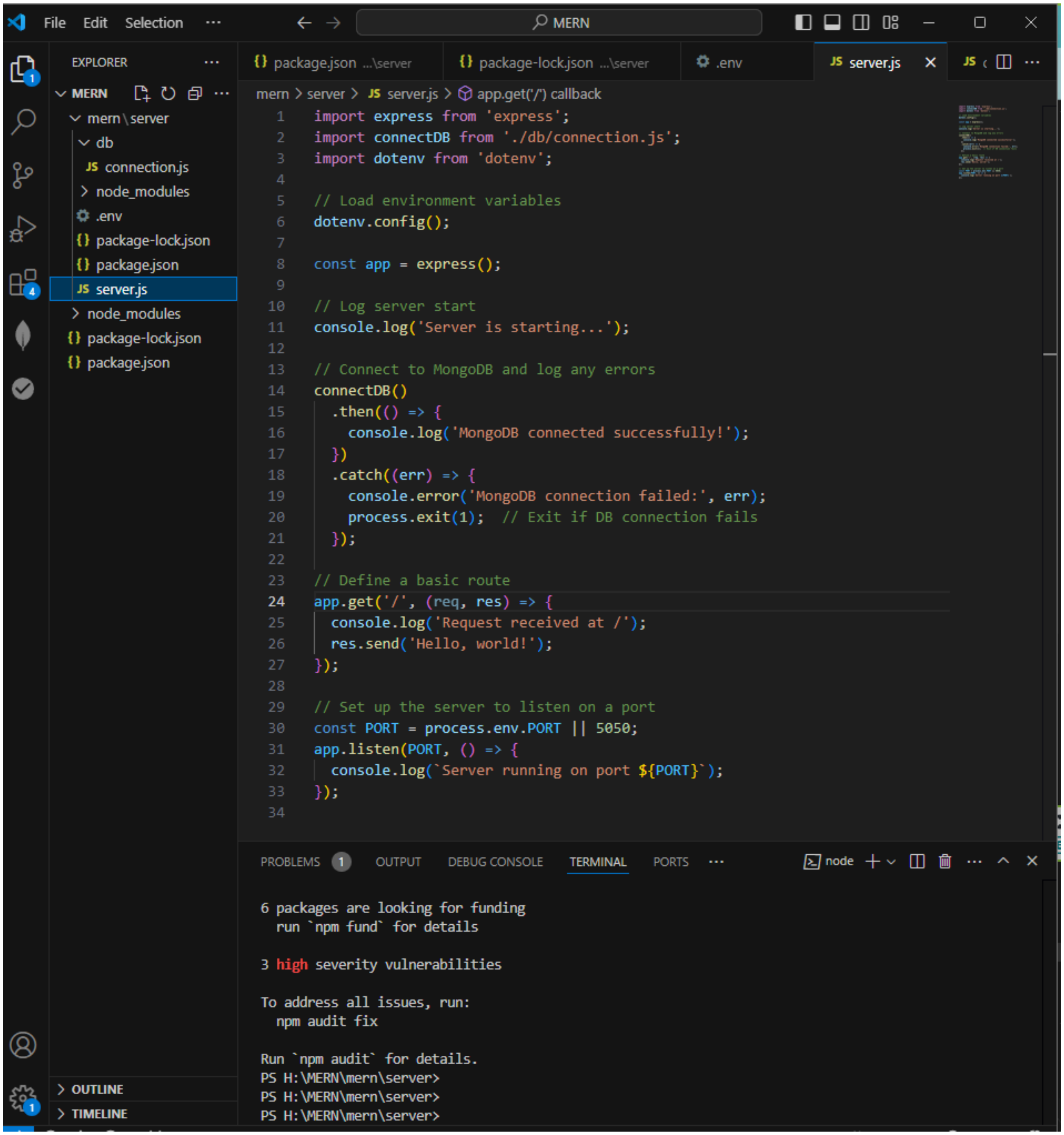
Frontend	React.js
Backend	Express.js
Computer Vision Models	OpenCV and TensorFlow
Database	MongoDB
Hosting	GoDaddy

Workload Distribution

Sadat Mahmud	Project Manager: Oversee environmental setup, and proper deployment
Rana Mustafa	Computer Vision Specialist: Optimize Computer Vision Models
Atoshi Samadder	Frontend Developer: Develop the user interface
Nabil Nashit	Backend Developer: Build the server-side logic and API endpoints

1. **Initial Setup of the MERN Stack**

- MongoDB: The database for storing your app's data, using the free cluster0 for our database.
- Express.js: The backend framework running on Node.js to handle HTTP requests and responses.
- React.js: The frontend JavaScript library for building the user interface we will use for our projects.
- Node.js: A JavaScript runtime environment used for running the server-side code.



```javascript
import express from 'express';
import connectDB from './db/connection.js';
import dotenv from 'dotenv';

// Load environment variables
dotenv.config();

const app = express();

// Log server start
console.log('Server is starting...');

// Connect to MongoDB and log any errors
connectDB()
  .then(() => {
    console.log('MongoDB connected successfully!');
  })
  .catch((err) => {
    console.error('MongoDB connection failed:', err);
    process.exit(1);  // Exit if DB connection fails
  });

// Define a basic route
app.get('/', (req, res) => {
  console.log('Request received at /');
  res.send('Hello, world!');
});

// Set up the server to listen on a port
const PORT = process.env.PORT || 5050;
app.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`);
});
```

```
6 packages are looking for funding
  run `npm fund` for details

3 high severity vulnerabilities

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
PS H:\MERN\mern\server>
PS H:\MERN\mern\server>
PS H:\MERN\mern\server>
```

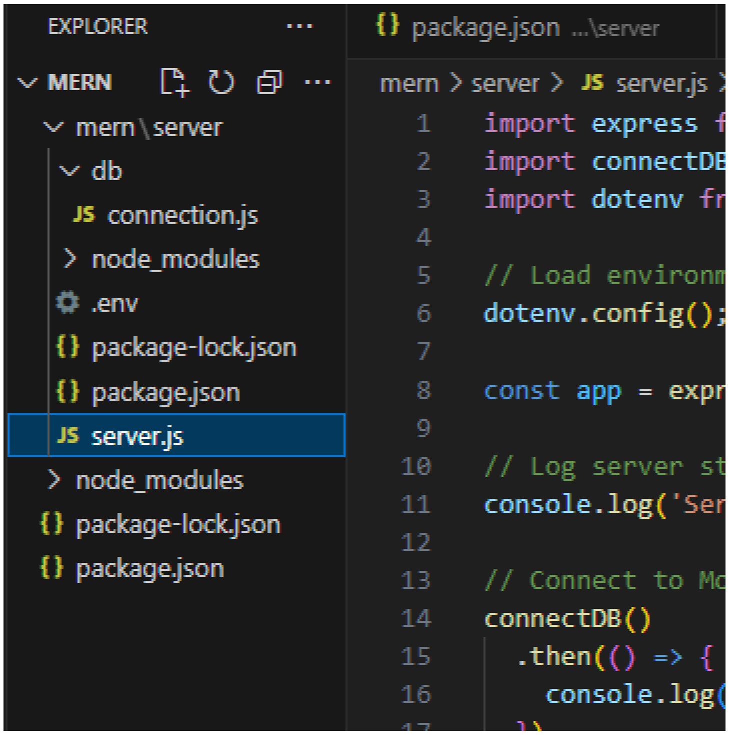
2. Installing the Necessary Libraries

Using the terminal of VS Code I have installed the required libraries , giving the code below:

```
npm init -y

npm install express mongoose dotenv

npm install --save-dev nodemon
```

3. Project Structure :

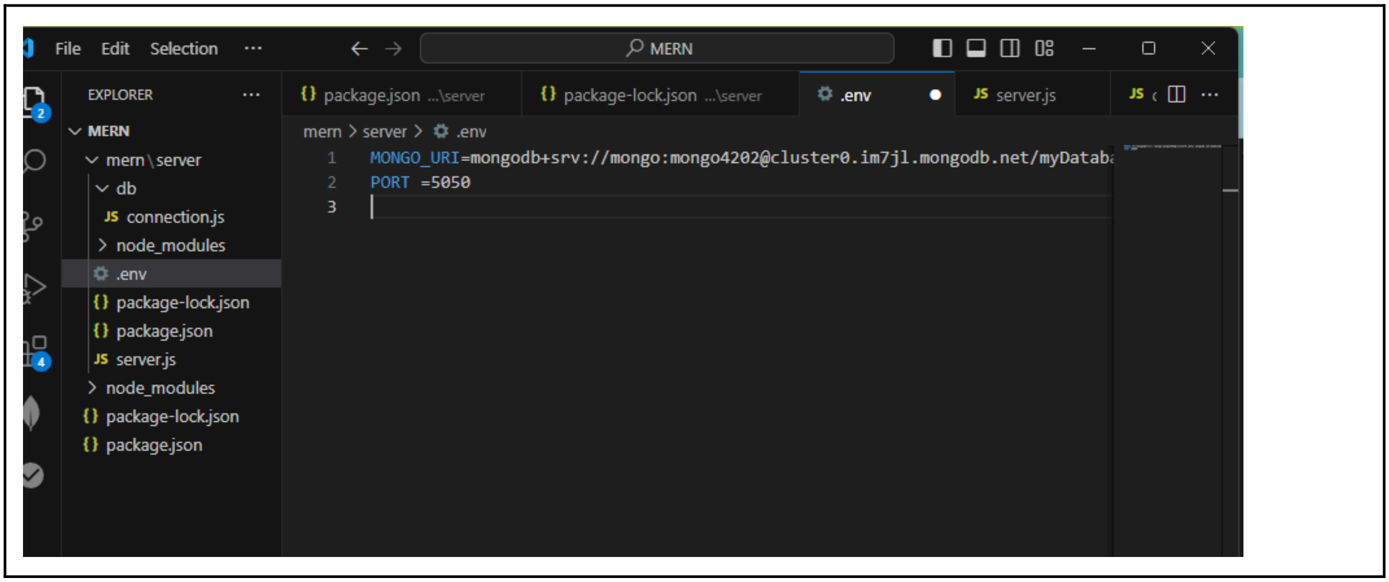
MERN(folder name that I have used for storing my setup)



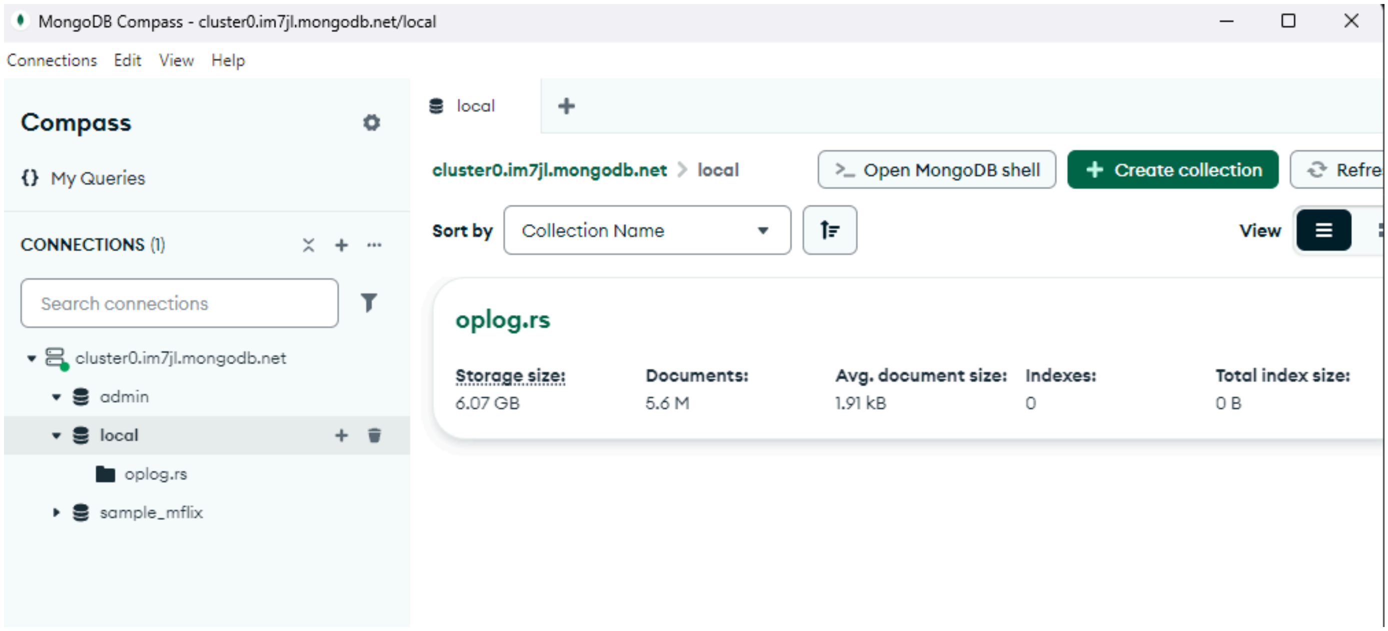
```
import express f
import connectDB
import dotenv fr

// Load environm
dotenv.config();

const app = expr

// Log server st
console.log('Ser

// Connect to Mo
connectDB()
  .then(() => {
    console.log(
```

4. MongoDB Setup:

I am using the MongoDB ATLAS, using the free version cluster0 and then connecting to my VS code environment.



```
MONGO_URI=mongodb+srv://mongo:mongo4202@cluster0.im7jl.mongodb.net/myDataba
PORT =5050

```



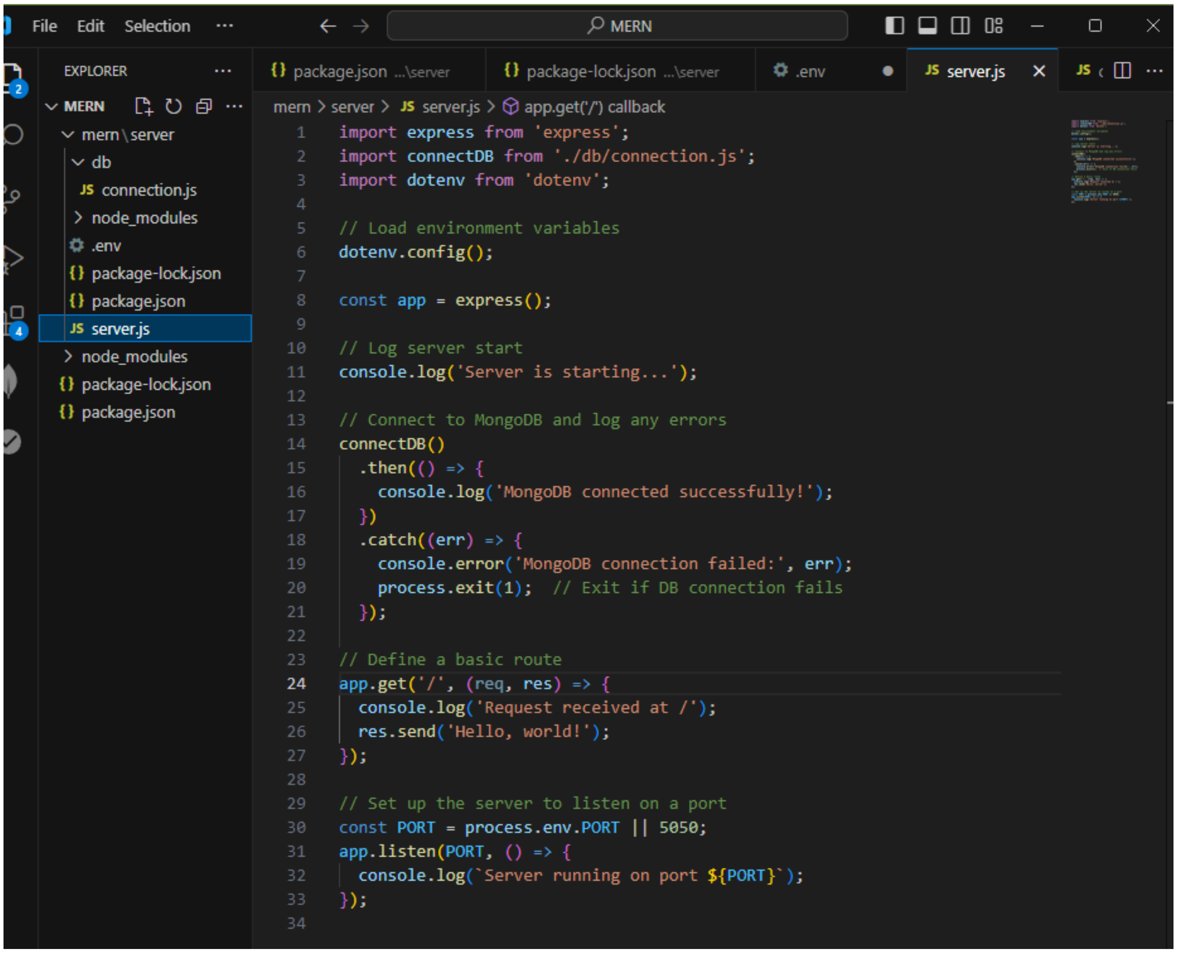
oplog.rs

Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
6.07 GB	5.6 M	1.91 kB	0	0 B

5. Server setup :

In the server .js mainly the `import connectDB from './db/connection.js';`

connects to the database.Using the “dotenv” for setting up the environment variables



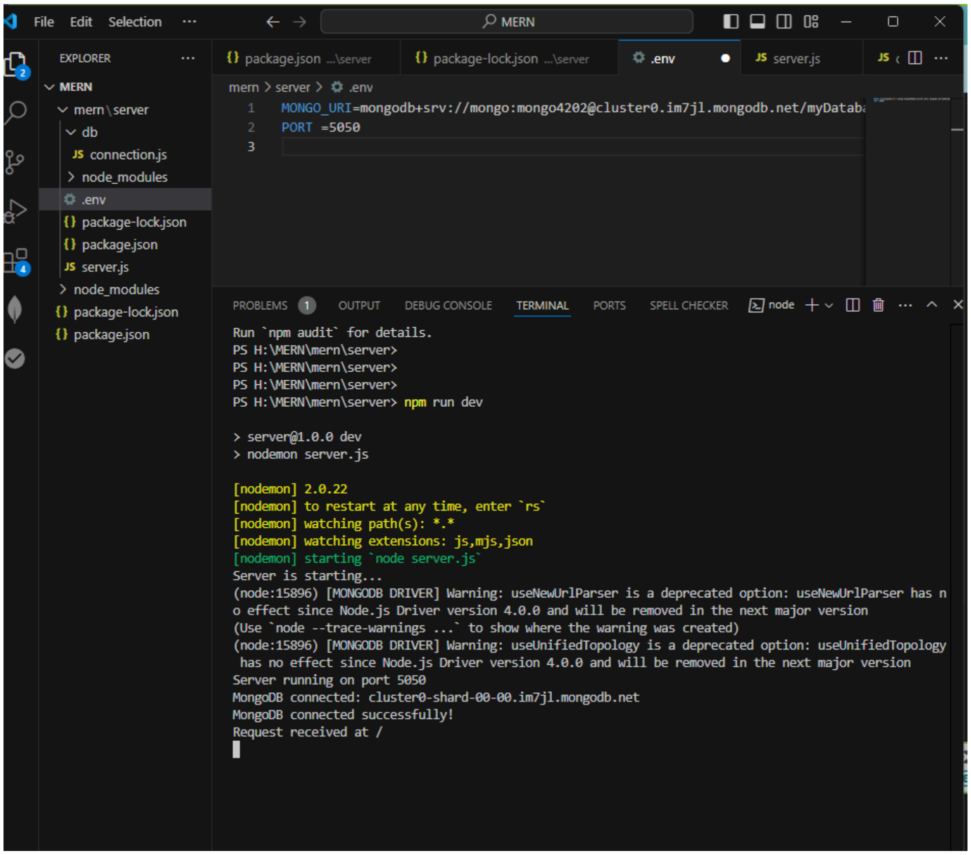
```javascript
import express from 'express';
import connectDB from './db/connection.js';
import dotenv from 'dotenv';

// Load environment variables
dotenv.config();

const app = express();

// Log server start
console.log('Server is starting...');

// Connect to MongoDB and log any errors
connectDB()
  .then(() => {
    console.log('MongoDB connected successfully!');
  })
  .catch((err) => {
    console.error('MongoDB connection failed:', err);
    process.exit(1);  // Exit if DB connection fails
  });

// Define a basic route
app.get('/', (req, res) => {
  console.log('Request received at /');
  res.send('Hello, world!');
});

// Set up the server to listen on a port
const PORT = process.env.PORT || 5050;
app.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`);
});
```

Under the mern/server :

```
Directory: H:\MERN\mern\server


Mode              LastWriteTime        Length Name
----             -------------        ------ ----


d-----       12/14/2024  10:04 PM            .env
d-----       12/14/2024   9:13 PM           node_modules
-a----       12/14/2024   9:39 PM        35473 package-lock.json
-a----       12/14/2024   9:13 PM          318 package.json
-a----       12/14/2024   9:15 PM            0 server.js
```

Running the server:

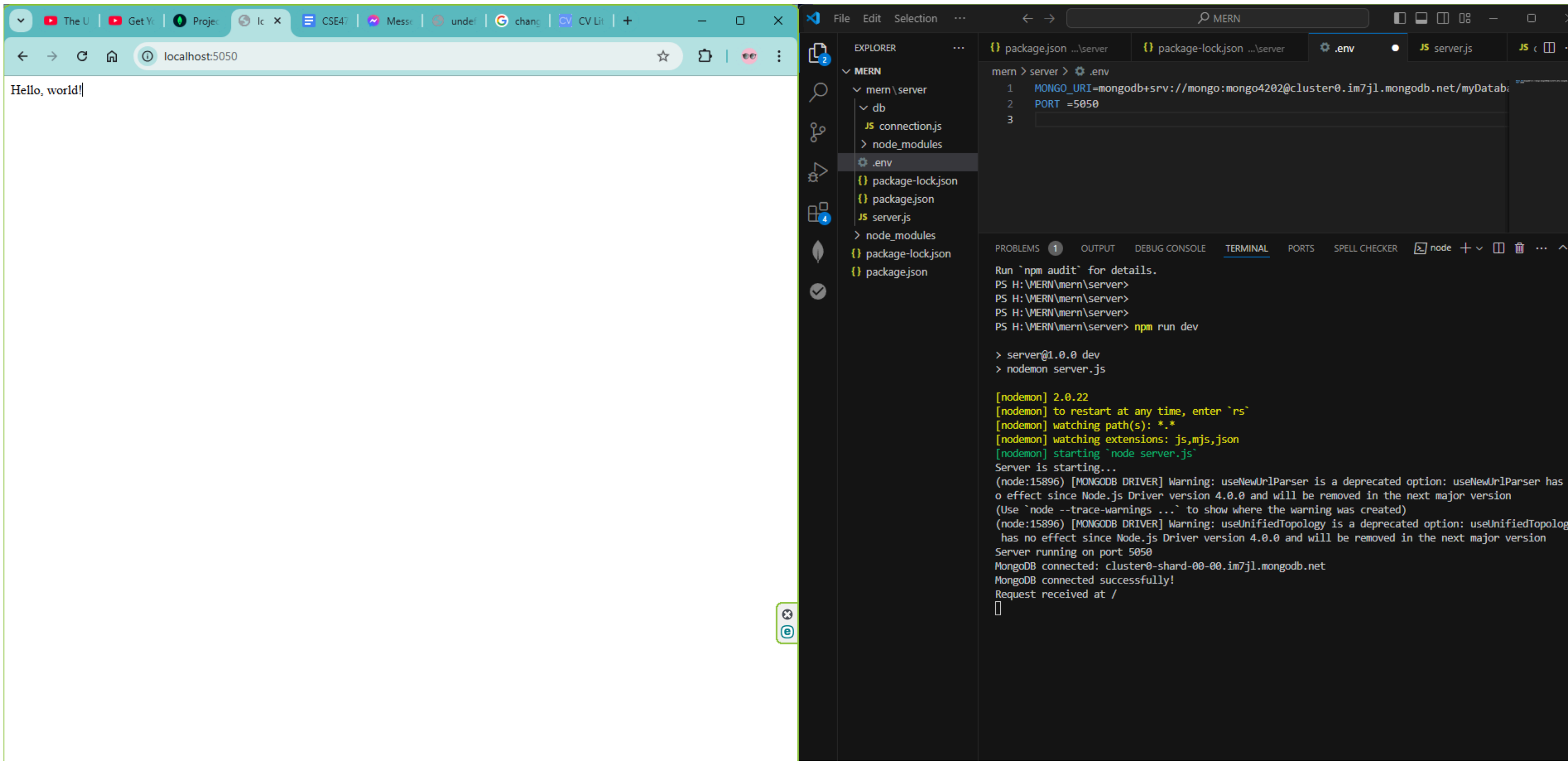
Using the “ npm run dev “I have successfully load the web-paged and this runs “nodemon”which will watch for file changes and restart the server automatically.



```
MONGO_URI=mongodb+srv://mongo:mongo4202@cluster0.im7jl.mongodb.net/myDataba
PORT =5050
```

```
Run `npm audit` for details.
PS H:\MERN\mern\server>
PS H:\MERN\mern\server>
PS H:\MERN\mern\server>
PS H:\MERN\mern\server> npm run dev

> server@1.0.0 dev
> nodemon server.js

[nodemon] 2.0.22
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node server.js`
Server is starting...
(node:15896) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has n
o effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:15896) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology
 has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server running on port 5050
MongoDB connected: cluster0-shard-00-00.im7jl.mongodb.net
MongoDB connected successfully!
Request received at /
```

web-page : “http://localhost:5050/”



Hello, world!

```
MONGO_URI=mongodb+srv://mongo:mongo4202@cluster0.im7jl.mongodb.net/myDataba
PORT =5050
```

```
Run `npm audit` for details.
PS H:\MERN\mern\server>
PS H:\MERN\mern\server>
PS H:\MERN\mern\server>
PS H:\MERN\mern\server> npm run dev

> server@1.0.0 dev
> nodemon server.js

[nodemon] 2.0.22
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node server.js`
Server is starting...
(node:15896) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has
o effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:15896) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopolog
 has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server running on port 5050
MongoDB connected: cluster0-shard-00-00.im7jl.mongodb.net
MongoDB connected successfully!
Request received at /
```