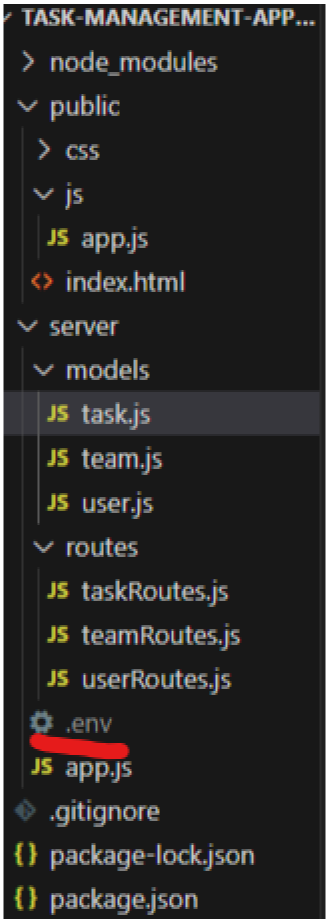
CP4485

Milestone 4: Access Control - Security

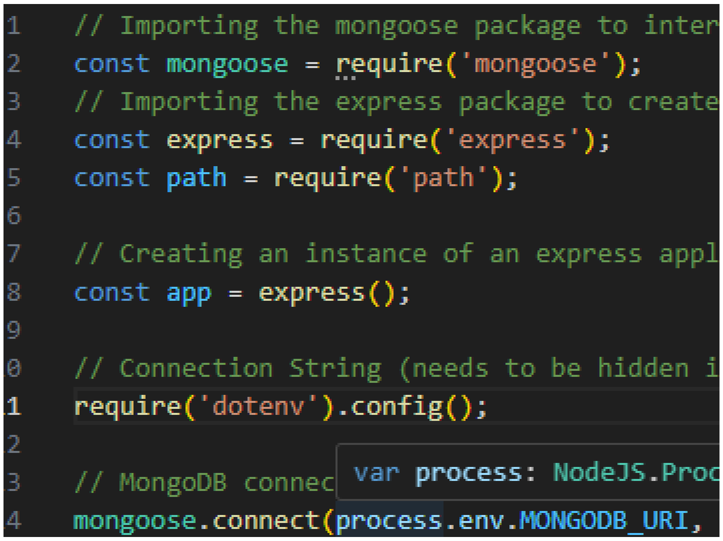
John-Michael Woodrow
John Cumby
Mohammad Aftab

In terms of security we didn't go through with user authentication but we did ensure the safe handling of environment variables such as the connection string for the mongodb cluster and general input validation.

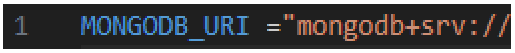
We set up the connection string(MONGODB_URI) in a .env file to access it securely without revealing our connection string to anyone that could browse/pull the code from the github repository by adding the .env file to the gitignore which serves as a specification of what files to hide when working with a github repository.



```
TASK-MANAGEMENT-APP...
> node_modules
v public
  > css
  v js
    app.js
  index.html
v server
  v models
    task.js
    team.js
    user.js
  v routes
    taskRoutes.js
    teamRoutes.js
    userRoutes.js
  .env
  app.js
.gitignore
package-lock.json
package.json
```

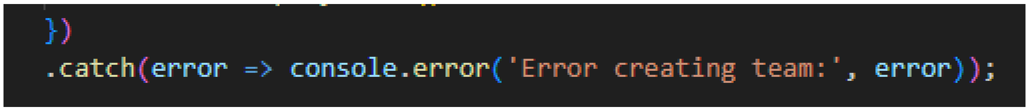


```
// Importing the mongoose package to inter
const mongoose = require('mongoose');
// Importing the express package to create
const express = require('express');
const path = require('path');

// Creating an instance of an express appl
const app = express();

// Connection String (needs to be hidden i
require('dotenv').config();

// MongoDB connec  var process: NodeJS.Proc
mongoose.connect(process.env.MONGODB_URI,
```



```
MONGODB_URI ="mongodb+srv://
```

We also handled input data to ensure that improper input did not break our application.



```
})
.catch(error => console.error('Error creating team:', error));
```

We could have used data encryption to store passwords for users and other common practices for authentication if that remained in the scope of our project.