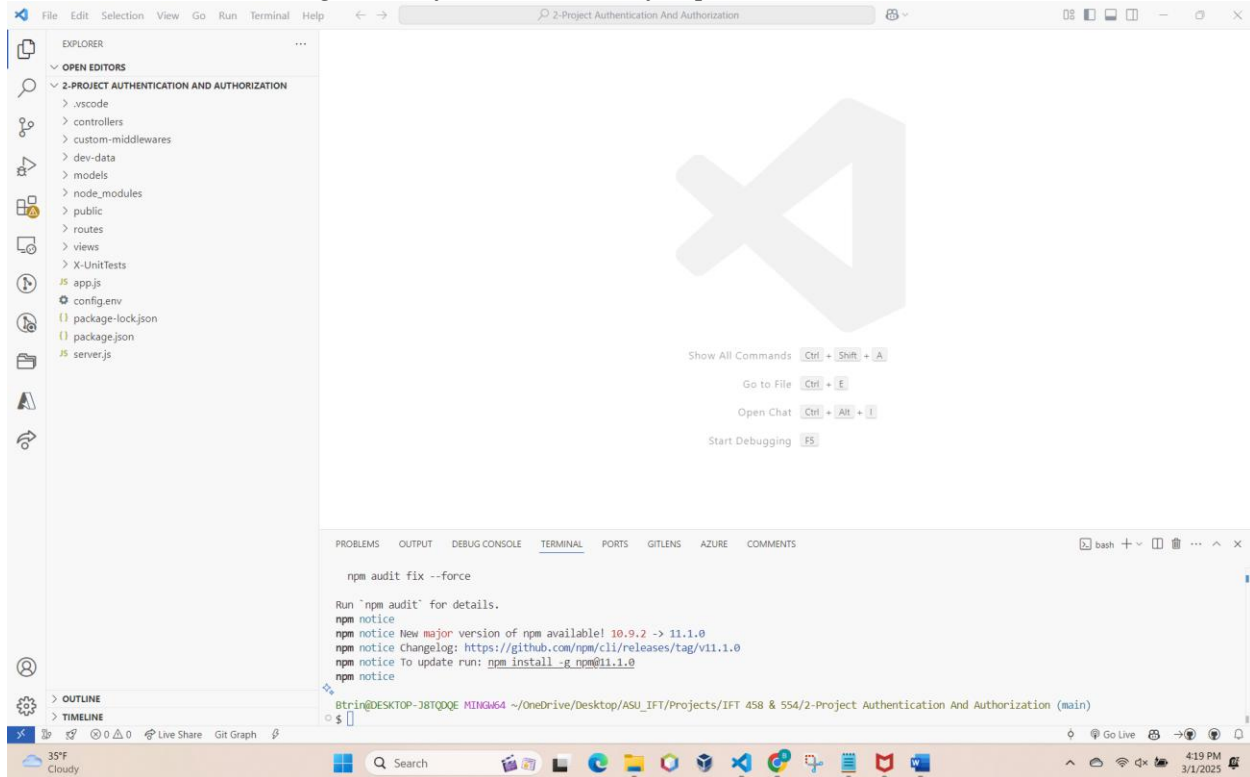


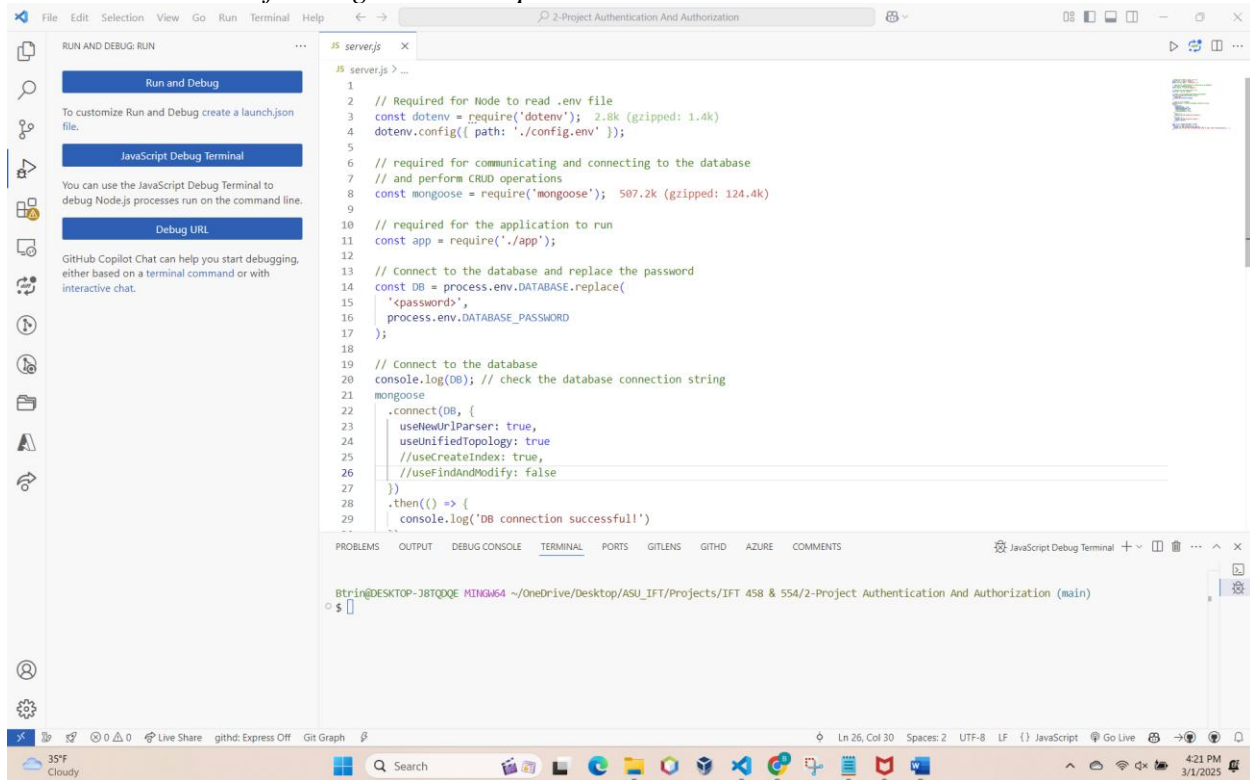
Module 4Activity 1**Brandon Trinkle****IFT 458: Middleware Prog & Database Sec****Professor: Dinesh Sthapit****Mar 1, 2025**

Module 4 Activity 1

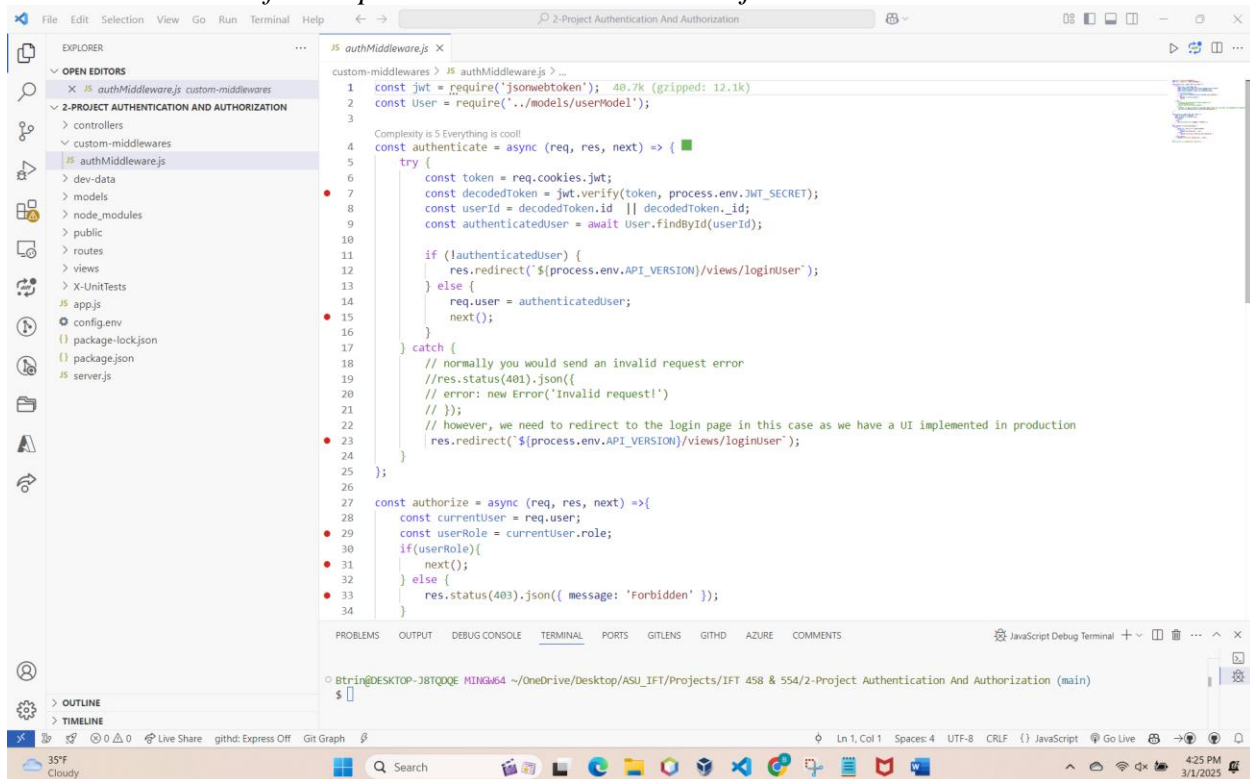
1. Terminal showing successful installation of dependencies.



2. Screenshot of Debug Terminal open in VSCode.



3. Screenshot of Breakpoints set in authMiddleware.js.



4. Screenshot of "Debugger Attached" message in the terminal.

The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the output of running a Node.js application using nodemon. The message "Debugger attached." is visible, indicating that the debugger is successfully connected to the application. Below this message, the terminal shows the application's startup sequence, including the directory path, the file being executed, and the port it is running on (4000). The application is a REST API for authentication and authorization, and it is currently running on port 4000. The terminal also shows a warning about the deprecated 'punycode' module and a message about the MongoDB database connection. The application is running on port 4000, and the terminal shows the message "Debugger attached.".

```

custom-middleware.js > JS authMiddleware.js > ...
1 const jwt = require('jsonwebtoken'); 40.7k (gzipped: 12.1k)
2 const User = require('../models/userModel');
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS GITHD AZURE COMMENTS
JavaScript Debug Terminal + - - - - -

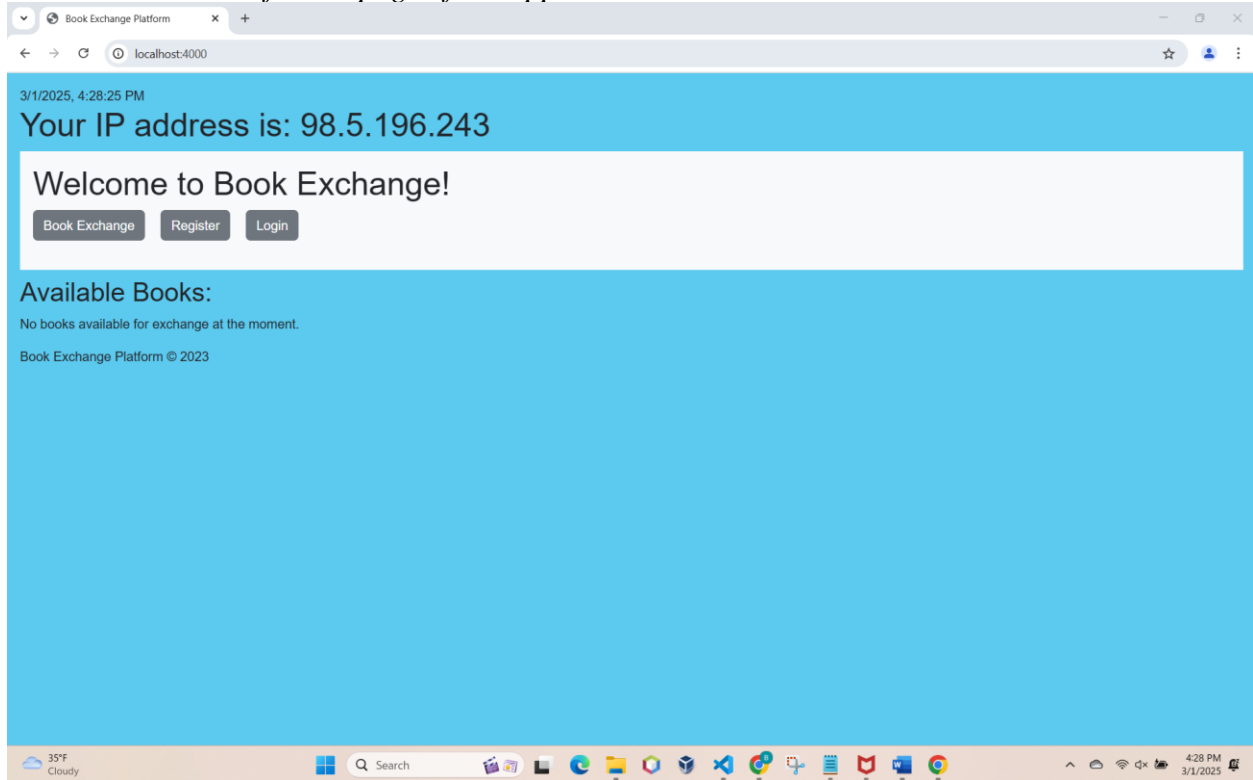
btr@DESKTOP-38TQDQE MINGW64 ~\OneDrive\Desktop\ASU_IFT\Projects\IFT 458 & 554\2-Project Authentication And Authorization (main)
$ nodemon ./server.js
Debugger attached.
[nodemon] 3.1.9
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node ./server.js`
Debugger attached.
dirname C:\Users\btr\OneDrive\Desktop\ASU_IFT\Projects\IFT 458 & 554\2-Project Authentication And Authorization
views C:\Users\btr\OneDrive\Desktop\ASU_IFT\Projects\IFT 458 & 554\2-Project Authentication And Authorization\views
public C:\Users\btr\OneDrive\Desktop\ASU_IFT\Projects\IFT 458 & 554\2-Project Authentication And Authorization\public
viewsUrl /api/v1/views
mongodbsrv://iftuser:u3eKQW@1lkoVY@cluster1.kateams.mongodb.net/users
(node:4748) [DEP0004] DeprecationWarning: The 'punycode' module is deprecated. Please use a userland alternative instead.
(Use 'node --trace-deprecation ...' to show where the warning was created)
App running on port 4000...
To test the IFT 458 REST App Click Or Type: http://localhost:4000...
DB connection failed!
MongoServerError: bad auth : Authentication failed.
    at Connection.onMessage (C:\Users\btr\OneDrive\Desktop\ASU_IFT\Projects\IFT 458 & 554\2-Project Authentication And Authorization\node_modules\
mongodb\lib\cmapped\connection.js:207:30)
    at MessageStream.canonimous (C:\Users\btr\OneDrive\Desktop\ASU_IFT\Projects\IFT 458 & 554\2-Project Authentication And Authorization\node_mod
ules\mongodb\lib\cmapped\connection.js:60:60)
    at MessageStream.emit (node:events:524:28)
    at processIncomingData (C:\Users\btr\OneDrive\Desktop\ASU_IFT\Projects\IFT 458 & 554\2-Project Authentication And Authorization\node_modules\m
ongodb\lib\cmapped\message_stream.js:132:20)
    at MessageStream.write (C:\Users\btr\OneDrive\Desktop\ASU_IFT\Projects\IFT 458 & 554\2-Project Authentication And Authorization\node_modules\
mongodb\lib\cmapped\message_stream.js:33:9)
    at writeOrBuffer (node:internal\streams\writable:572:12)
    at _write (node:internal\streams\writable:501:10)
    at Writable.write (node:internal\streams\writable:510:10)
    at TLSSocket.ondata (node:internal\streams\readable:1009:22)
    at TLSSocket.emit (node:events:524:28) {
  ok: 0,
  code: 8000,
  codeName: 'AtlasError',
  [Symbol(errorLabels)]: Set(1) { 'HandshakeError' }
}

```

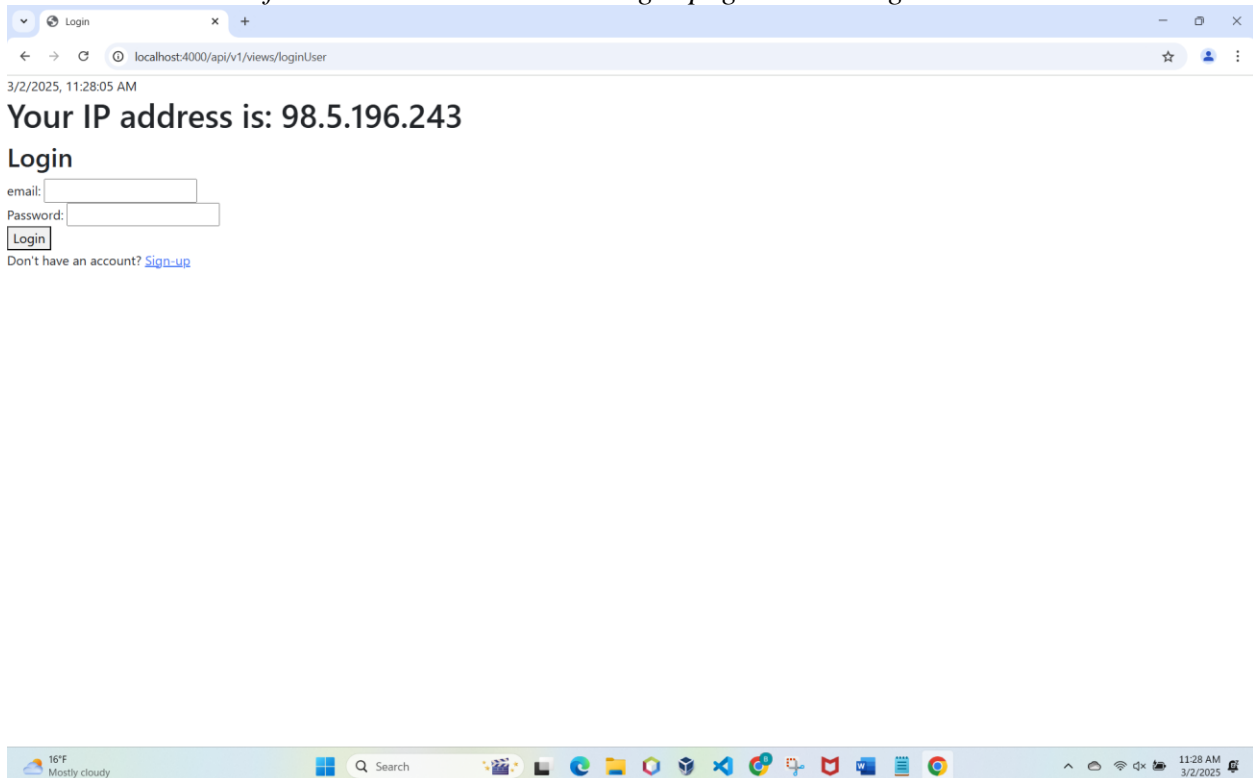
The screenshot also shows a list of breakpoints in the 'authMiddleware.js' file, which are used for debugging the application. The breakpoints are located at lines 7, 15, 23, 29, 31, and 33 of the file. The breakpoints are set on the 'authMiddleware.js' file, and they are used to debug the application's behavior. The breakpoints are set on the 'authMiddleware.js' file, and they are used to debug the application's behavior. The breakpoints are set on the 'authMiddleware.js' file, and they are used to debug the application's behavior.

Note: The iftuser password in config.env didn't work. I changed the MongoDB database and password to the one created earlier in class. This wasn't specified in the instructions, but it was necessary to complete the assignment on time.

5. Screenshot of Homepage of the application in the browser.



6. Screenshot of the "Add New Book Exchange" page and the login screen.



3/2/2025, 11:30:03 AM

Your IP address is: 98.5.196.243

Authorization Successful!

Welcome, Brandon Trinkle!

btrinkle1@asu.edu!

Token,

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY3YzQ4NzhiNTA4MmWJmN2FhOWNlMDFhMSIsImh0dCI6MTc0MDkzMzAwMywiZXhwIjoxNzQxNzk3MDA3fQ.hOC3xhZYd6HU0OmRRpUxR1Er-rHPpwZIP8Vcn09_ceM!

You can now access the platform's features.

[Go to home](#)

7. Screenshot of "Add New Book Exchange" page, indicating access to protected resources.

Book Exchange Platform - Add New Book Exchange

localhost:4000/api/v1/books

3/2/2025, 11:33:15 AM

Your IP address is: 98.5.196.243

Add a New Book for Exchange

Welcome, ! Dashboard Logout

Title:

Author:

Genre:

Description:

Book Cover: No file chosen

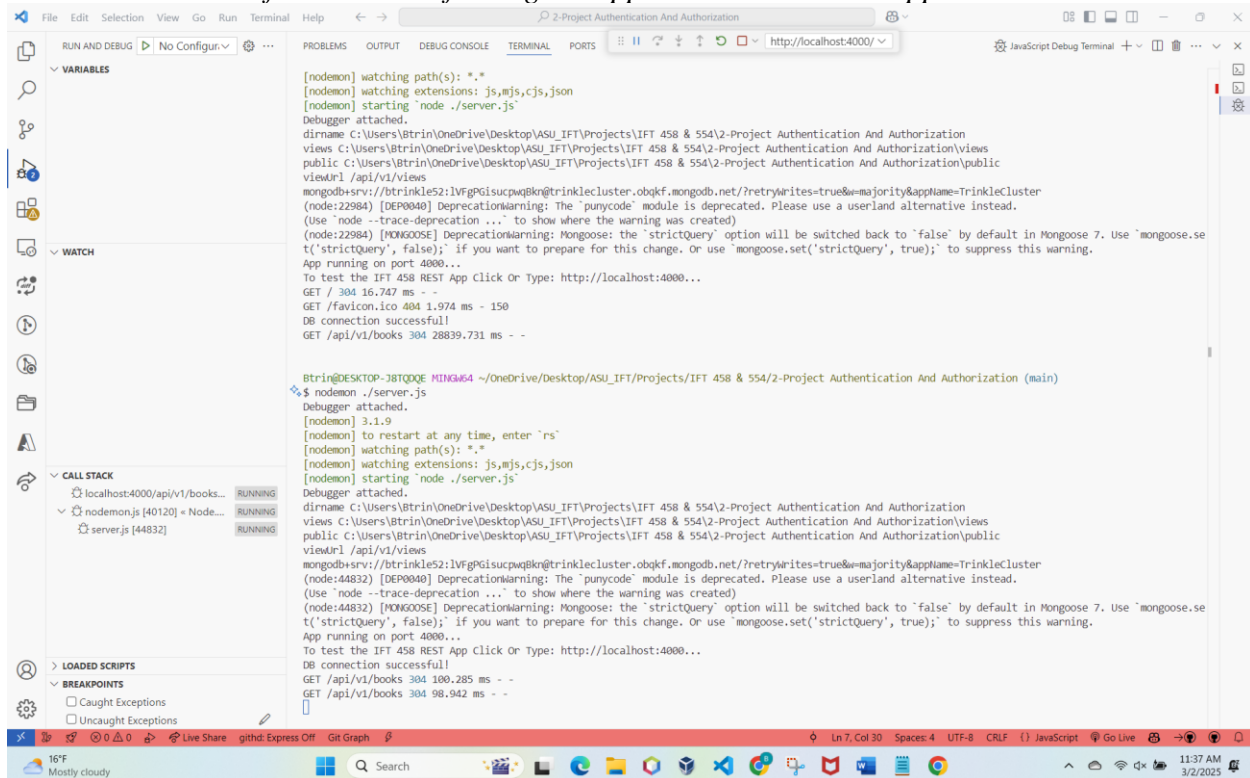
Book Exchange Platform © 2023

16°F Mostly cloudy

Search

11:34 AM 3/2/2025

8. Screenshot of terminal confirming the application has been stopped and restarted.



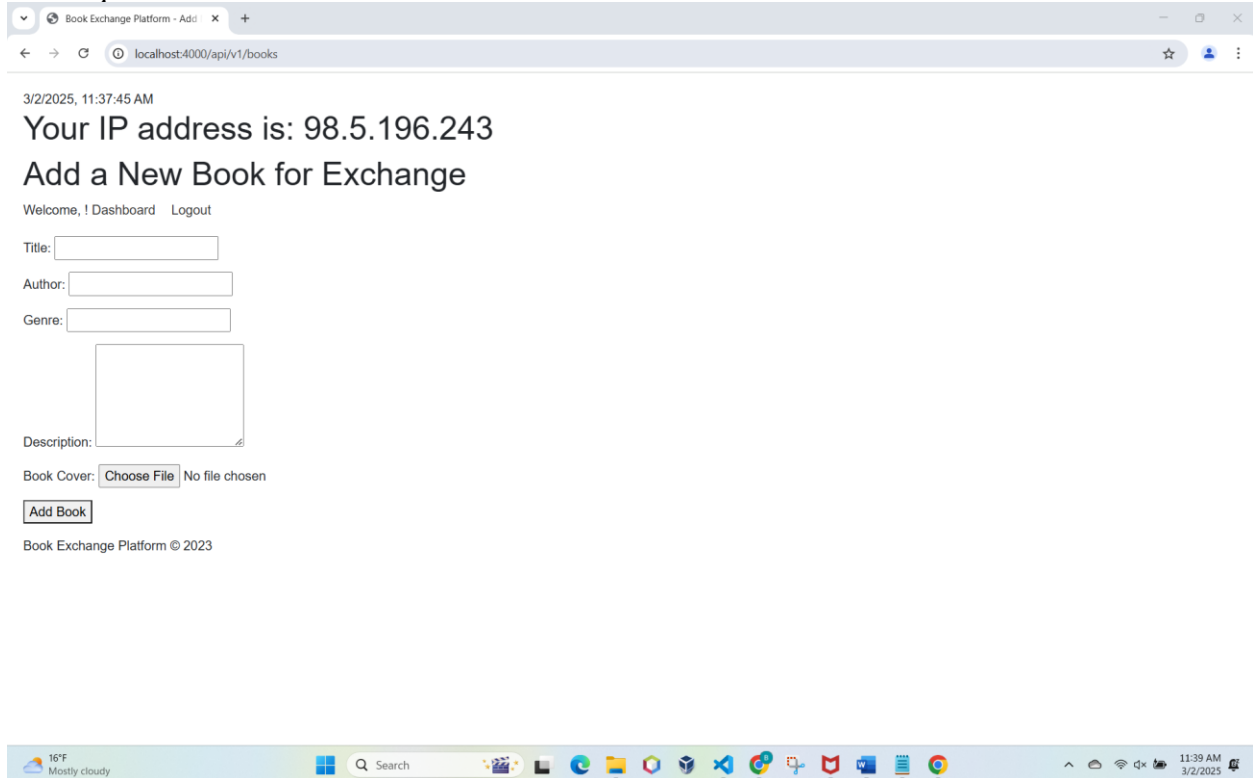
```

[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting 'node ./server.js'
Debugger attached.
dirname C:\Users\Btrin\OneDrive\Desktop\ASU_IFT\Projects\IFT 458 & 554\2-Project Authentication And Authorization
views C:\Users\Btrin\OneDrive\Desktop\ASU_IFT\Projects\IFT 458 & 554\2-Project Authentication And Authorization\views
public C:\Users\Btrin\OneDrive\Desktop\ASU_IFT\Projects\IFT 458 & 554\2-Project Authentication And Authorization\public
viewUrl /api/v1/views
mongodb+srv://btrinkle52:1VfpgGisucpug@trinklecluster.obqkf.mongodb.net/?retryWrites=true&majority=&appName=TrinkleCluster
(node:22984) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use a userland alternative instead.
(node:22984) [MONGODB] DeprecationWarning: Mongoose: the 'strictQuery' option will be switched back to 'false' by default in Mongoose 7. Use 'mongoose.set('strictQuery', false);' if you want to prepare for this change. Or use 'mongoose.set('strictQuery', true);' to suppress this warning.
App running on port 4000...
To test the IFT 458 REST App Click Or Type: http://localhost:4000...
GET / 304 16.747 ms - -
GET /favicon.ico 404 1.974 ms - 150
DB connection successful!
GET /api/v1/books 304 28839.731 ms - -

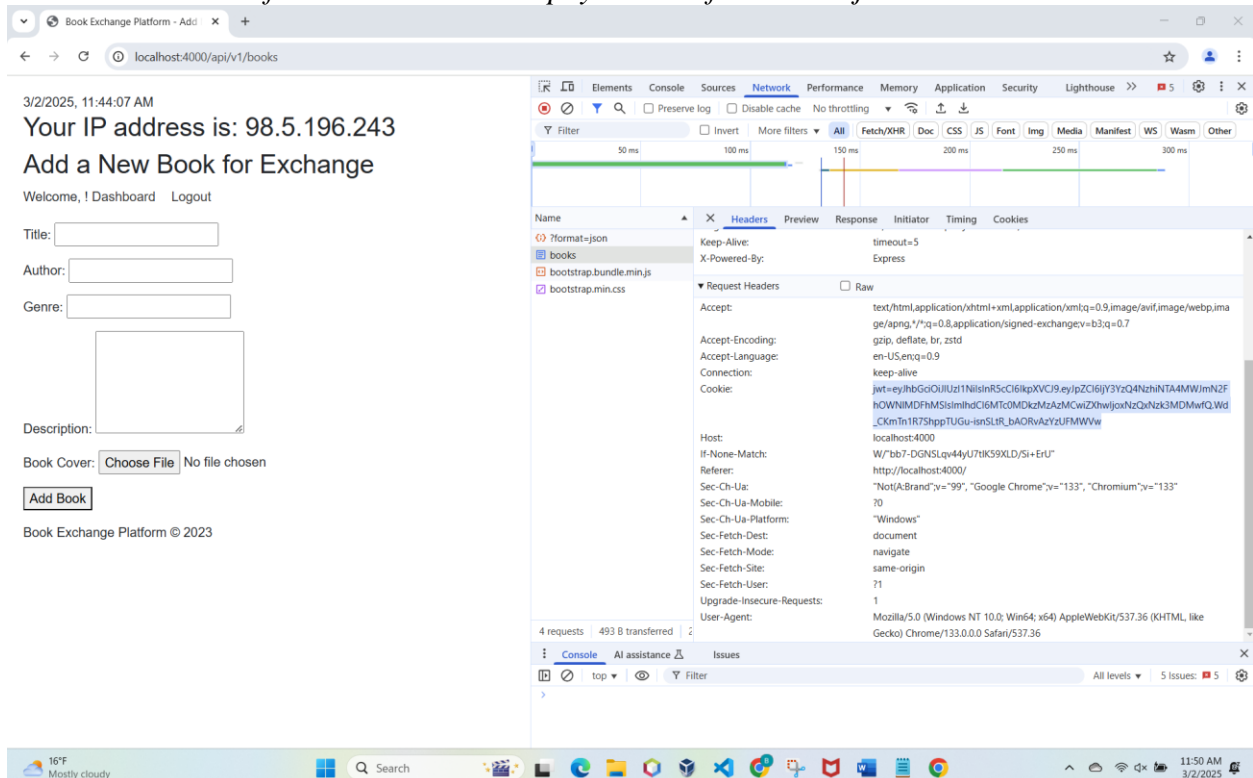
Btrin@DESKTOP-38TQDQE MINGW64 ~/OneDrive/Desktop/ASU_IFT/Projects/IFT 458 & 554/2-Project Authentication And Authorization (main)
$ nodemon ./server.js
Debugger attached.
[nodemon] 3.1.9
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting 'node ./server.js'
Debugger attached.
dirname C:\Users\Btrin\OneDrive\Desktop\ASU_IFT\Projects\IFT 458 & 554\2-Project Authentication And Authorization
views C:\Users\Btrin\OneDrive\Desktop\ASU_IFT\Projects\IFT 458 & 554\2-Project Authentication And Authorization\views
public C:\Users\Btrin\OneDrive\Desktop\ASU_IFT\Projects\IFT 458 & 554\2-Project Authentication And Authorization\public
viewUrl /api/v1/views
mongodb+srv://btrinkle52:1VfpgGisucpug@trinklecluster.obqkf.mongodb.net/?retryWrites=true&majority=&appName=TrinkleCluster
(node:44832) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use a userland alternative instead.
(node:44832) [MONGODB] DeprecationWarning: Mongoose: the 'strictQuery' option will be switched back to 'false' by default in Mongoose 7. Use 'mongoose.set('strictQuery', false);' if you want to prepare for this change. Or use 'mongoose.set('strictQuery', true);' to suppress this warning.
App running on port 4000...
To test the IFT 458 REST App Click Or Type: http://localhost:4000...
DB connection successful!
GET /api/v1/books 304 100.285 ms - -
GET /api/v1/books 304 98.942 ms - -

```

Note: Was able to stop and restart the server and did not have to log back in. However, this appears like a security vulnerability.



10. Screenshot of JWT token data and payload verification on jwt.io.



The image shows two screenshots from a computer screen. The top screenshot is of the JWT.io website, which is a JSON Web Token (JWT) decoder. The 'Encoded' section contains a JWT token: `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY3YzQ4NzhNTA4MWMmN2FhOWNlMDFhMlSiImIhdCI6MTc0MDkzMzAzMCwiZXhwIjoxNzQxNzk3MDMwfw.Wd_CKmTn1R7ShppTUGu-1snSLtr_bAORvAzYzUFMWVw`. The 'Decoded' section shows the token's structure: Header (Algorithm: HS256, Type: JWT), Payload (ID: 67c4878b5081bf7aa9ce01a1, iat: 1740933030, exp: 1741797030), and Signature (HMACSHA256). The bottom screenshot is of the MongoDB Atlas interface. It shows the 'test.users' collection with a single document. The document contains fields: _id (ObjectId('67c4878b5081bf7aa9ce01a1')), name ('Brandon Trinkle'), email ('btrinkle@asu.edu'), role ('client_'), password ('62a6129zht5y5pLc1BQz8p2cM5Oaesc1u6r5sDmfXNg.TvOV0d5r16A1d7/u_'), active (true), roles ('client_'), and __v (0).

JWT Token Decoder

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY3YzQ4NzhNTA4MWMmN2FhOWNlMDFhMlSiImIhdCI6MTc0MDkzMzAzMCwiZXhwIjoxNzQxNzk3MDMwfw.Wd_CKmTn1R7ShppTUGu-1snSLtr_bAORvAzYzUFMWVw
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "id": "67c4878b5081bf7aa9ce01a1",
  "iat": 1740933030,
  "exp": 1741797030
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
)
☐ secret base64 encoded
```

MongoDB Atlas

test.users

STORAGE SIZE: 20KB LOGICAL DATA SIZE: 204B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 40KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

Generate queries from natural language in Compass

Filter Type a query: { field: 'value' } **Reset** **Apply** **Options**

QUERY RESULTS: 1-1 OF 1

```
1 _id: ObjectId('67c4878b5081bf7aa9ce01a1') ObjectID
2 name: "Brandon Trinkle" String
3 email: "btrinkle@asu.edu" String
4 role: "client_" String
5 password: "62a6129zht5y5pLc1BQz8p2cM5Oaesc1u6r5sDmfXNg.TvOV0d5r16A1d7/u_" String
6 active: true Boolean
7 roles: "client_" String
8 __v: 0 Int32
```

CANCEL **UPDATE**

Note: Jwt token decoder confirms that the hash encoded in our jwt token matches the hash assigned to our _id stored in MongoDB. The specific users collection stores login information – which uses jwt (json web token) as per the code specified in authController.js and userModel.js.