

Task 1: Fix Login

Endpoint: POST /auth/login The server should generate a session and log an OTP to the console.

Test Command:

```
curl -X POST http://localhost:3000/auth/login \
-H "Content-Type: application/json" \
-d '{"email":<YOUR EMAIL@example.com>,"password":"password123"}'
```

The screenshot shows a Mac desktop environment. In the center is a terminal window titled 'apple — zsh — 136x33' with the command and its output:

```
apple@Apples-MacBook-Pro ~ % curl -X POST http://localhost:3000/auth/login \
-H "Content-Type: application/json" \
-d '{"email":"peter123@gmail.com","password":"peter0786"}'
{"status": "success", "message": "OTP sent", "loginSessionId": "030d0ee52f08704803e3b06cb29d218d"}
apple@Apples-MacBook-Pro ~ %
```

To the right of the terminal is a code editor window titled 'JS mockDb.js M'. A tooltip 'start typing to dismiss or don't' is visible over the editor area. Below the code editor is another terminal window titled 'zsh' with the message 'node broke...'. At the bottom of the screen is the macOS Dock containing various application icons.

Task 2: Fix OTP Verification

Endpoint: POST /auth/verify-otp The server fails to verify the OTP correctly. You need to find out why. Hint: Check data types and how cookies are set.

Test Command: (Replace <loginSessionId> and <otp> with values from Task 1)

```
curl -c cookies.txt -X POST http://localhost:3000/auth/verify-otp \
-H "Content-Type: application/json" \
-d '{"loginSessionId":<loginSessionId>,"otp":<otp_from_logs>}'
```

The screenshot shows a macOS desktop environment. In the center is a terminal window titled 'apple' with the command:

```
curl -c cookies.txt -X POST http://localhost:3000/auth/verify-otp \
-H "Content-Type: application/json" \
-d '{"loginSessionId":<loginSessionId>,"otp":<otp_from_logs>}'
```

The output of the command is:

```
apple@Apples-MacBook-Pro ~ % curl -c cookies.txt -X POST http://localhost:3000/auth/verify-otp \
-H "Content-Type: application/json" \
-d '{"loginSessionId":<loginSessionId>,"otp":<otp_from_logs>}'
{"status":"success","message":"OTP verified"}%
apple@Apples-MacBook-Pro ~ %
```

To the right of the terminal is a timeline tool window titled 'Timeline'. It shows a sequence of events:

- POST /auth/login -> 200 (1ms)
- POST /auth/verify-otp -> 401 (0ms)
- POST /auth/login
- [OTP] Session 33afbd3a2d543881e7d68fd51b22effe generated. OTP: 685332
- POST /auth/login -> 200 (1ms)
- POST /auth/verify-otp
- POST /auth/verify-otp -> 200 (1ms)

The timeline tool has two tabs: 'OUTLINE' and 'TIMELINE', with 'OUTLINE' currently selected.

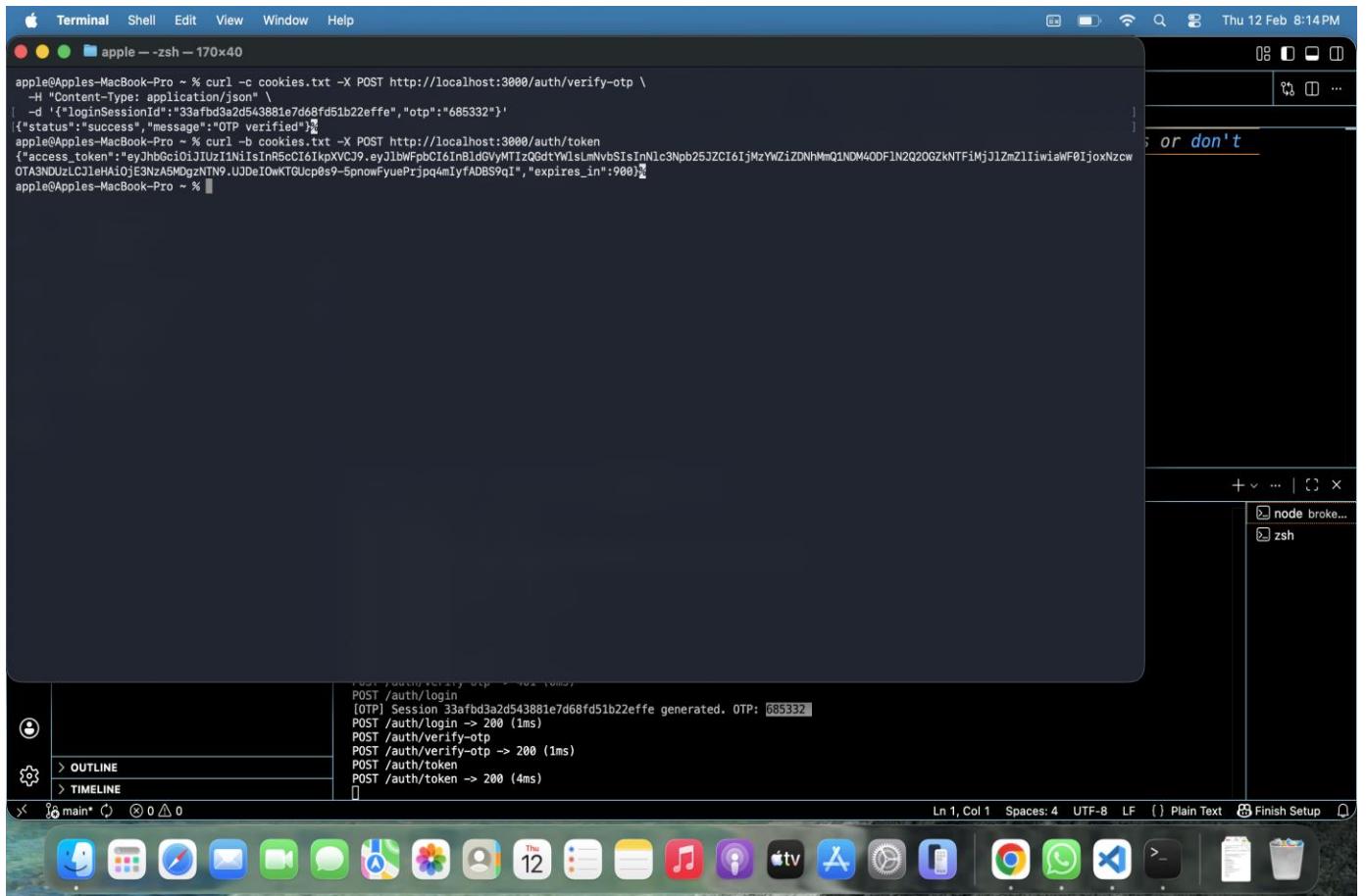
At the bottom of the screen is the macOS dock with various application icons.

Task 3: Fix Token Generation

Endpoint: POST /auth/token This endpoint is supposed to issue a JWT, but it has a bug in how it reads the session.

Test Command:

```
# Uses the cookie captured in Task 2
curl -b cookies.txt -X POST http://localhost:3000/auth/token
```



The screenshot shows a macOS Terminal window with the following content:

```
apple@Apples-MacBook-Pro ~ % curl -c cookies.txt -X POST http://localhost:3000/auth/verify-otp \
-H "Content-Type: application/json" \
-d '{"loginSessionId":"33afbd3a2d543881e7d68fd51b22effe","otp":"685332"}'
{"status": "success", "message": "OTP verified"}
apple@Apples-MacBook-Pro ~ % curl -b cookies.txt -X POST http://localhost:3000/auth/token
{"access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbC16InBldGVyMjIzQdtyWlslnNbSIisInNlc3Npb25JZCI6IJMzYwZizDNHMmQ1NDM4ODF1N2Q2OGZkNTFiMjJlZmZliwiawF0IjoxNzcwOTAsNDUzLCJleHAiOiJE3NzA5MDgzNTN9.UJDcIWKTGUcp0s9-5pnowfyuePrjq4mIyfADBS9qI", "expires_in": 900}
apple@Apples-MacBook-Pro ~ %
```

The terminal window has a status bar at the top indicating the date and time: Thu 12 Feb 8:14 PM. The bottom of the window shows the macOS Dock with various application icons.

Task 4: Fix Protected Route Access

Endpoint: GET /protected Ensure the middleware correctly validates the token.

Test Command:

```
# Replace <jwt> with the token from Task 3  
curl -H "Authorization: Bearer <jwt>" http://localhost:3000/protected
```

```
apple@Apples-MacBook-Pro ~ % curl -c cookies.txt -X POST http://localhost:3000/auth/verify-otp \
-H "Content-Type: application/json" \
-d '{"loginSessionId":"33afbd3a2d543881e7d68fd51b22effe","otp":"685332"}'
{"status": "success", "message": "OTP verified"}]
apple@Apples-MacBook-Pro ~ % curl -b cookies.txt -X POST http://localhost:3000/auth/token \
{"access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXCV39.eyJlbWFpbC16InBldGVyMTIzQGdtYWlsLmNvbSIsInNlc3Npb25JZCI6IjMzYwZiZDNhMmQ1NDM4ODF1N2Q2OGZkNTFiMjJlZmZliiwiWF0IjoxNzcwOTA3NDUzLC1leHAIoJE3NzA5MDgZNTP9JUJeI0wKTGUcp0s9-5pnowFyuePrjpq4mIyfADBS9I", "expires_in": "900"}]
apple@Apples-MacBook-Pro ~ % curl -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXCV39.eyJlbWFpbC16InBldGVyMTIzQGdtYWlsLmNvbSIsInNlc3Npb25JZCI6IjMzYwZiZDNhMmQ1NDM4ODF1N2Q2OGZkNTFiMjJlZmZliiwiWF0IjoxNzcwOTA3NDUzLC1leHAIoJE3NzA5MDgZNTP9JUJeI0wKTGUcp0s9-5pnowFyuePrjpq4mIyfADBS9I" http://localhost:3000/protected
{"status": "success", "message": "Access granted", "user": {"email": "peter12@gmail.com", "sessionId": "33afbd3a2d543881e7d68fd51b22effe", "iat": 1770907453, "exp": 1770908353}, "suuuuuuuu"}
apple@Apples-MacBook-Pro ~ %
```