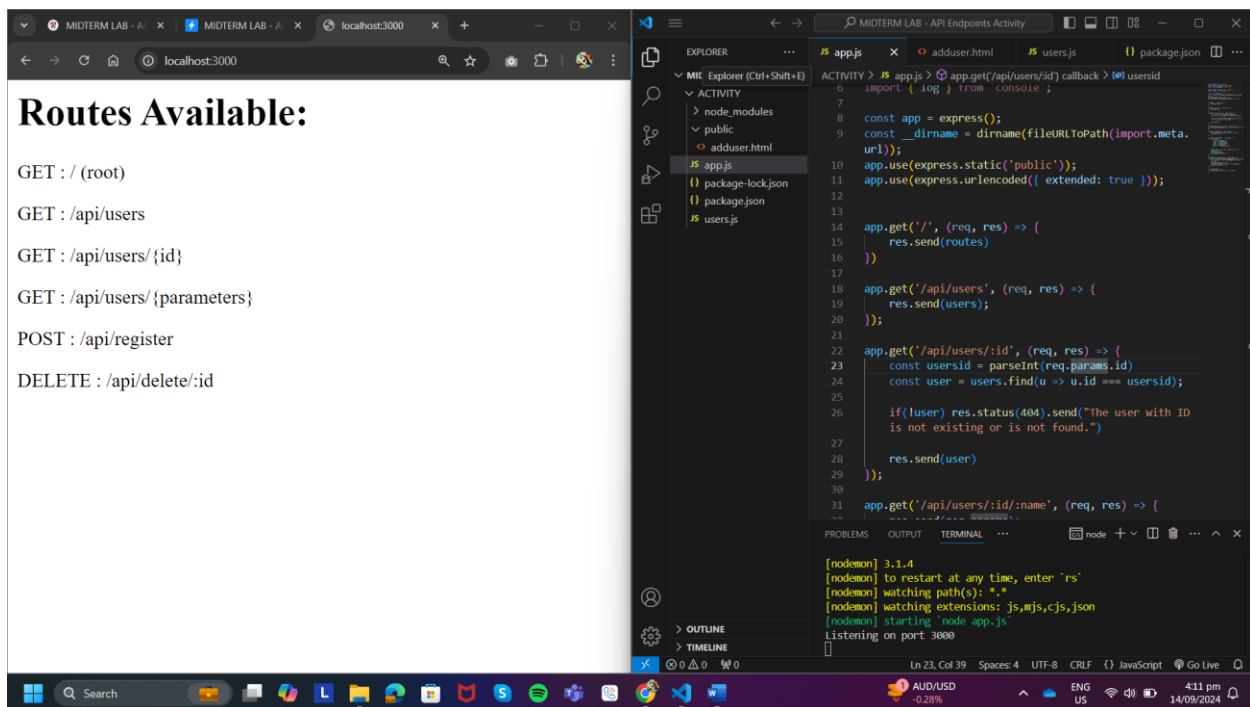


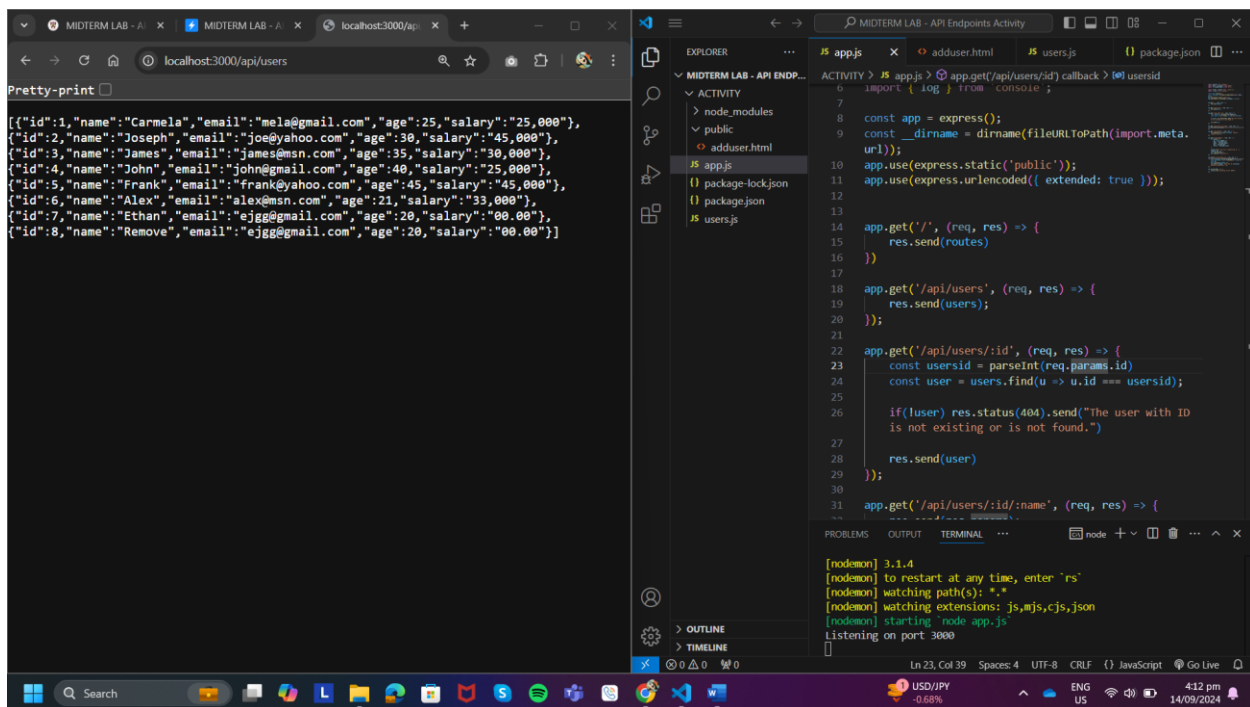
Name: Gonzales, Ethan James G.

Section: WD-302

1. Root Route: Displays available routes



2. GET /api/users: Returns all users.



3. GET /api/users/: Fetches a specific user by ID.

The screenshot shows a web browser at `localhost:3000/api/users/3` displaying a JSON response for user ID 3. The response is: `{ "id": 3, "name": "James", "email": "james@msn.com", "age": 35, "salary": "30,000" }`. The VS Code editor shows the `app.js` file with the following code:

```
1 import { log } from console;
2
3 const app = express();
4 const __dirname = dirname(fileURLToPath(import.meta.url));
5 app.use(express.static('public'));
6 app.use(express.urlencoded({ extended: true }));
7
8 app.get('/', (req, res) => {
9   res.send(routes)
10 })
11
12 app.get('/api/users', (req, res) => {
13   res.send(users);
14 });
15
16 app.get('/api/users/:id', (req, res) => {
17   const userid = parseInt(req.params.id)
18   const user = users.find(u => u.id === userid);
19
20   if(!user) res.status(404).send("The user with ID is not existing or is not found.");
21   res.send(user)
22 });
23
24 app.get('/api/users/:id:name', (req, res) => {
25   // ...
26 })
```

The terminal shows the following output:

```
[nodeemon] 3.1.4
[nodeemon] to restart at any time, enter 'rs'
[nodeemon] watching path(s): *.*
[nodeemon] watching extensions: js,mjs,cjs,json
[nodeemon] starting 'node app.js'
Listening on port 3000
```

4. GET /api/users/: Retrieves and displays parameters.

The screenshot shows a web browser at `localhost:3000/api/users/4/John` displaying a JSON response for user ID 4. The response is: `{ "id": "4", "name": "John" }`. The VS Code editor shows the `app.js` file with the following code:

```
13
14 app.get('/', (req, res) => {
15   res.send(routes)
16 })
17
18 app.get('/api/users', (req, res) => {
19   res.send(users);
20 });
21
22 app.get('/api/users/:id', (req, res) => {
23   const userid = parseInt(req.params.id)
24   const user = users.find(u => u.id === userid);
25
26   if(!user) res.status(404).send("The user with ID is not existing or is not found.");
27   res.send(user)
28 });
29
30 app.get('/api/users/:id:name', (req, res) => {
31   res.send(req.params);
32 });
33
34 app.get('/api/adduser', (req, res) => {
35   res.sendFile(path.join(__dirname, 'public', 'adduser.html'));
36 });
37
38
```

The terminal shows the following output:

```
[nodeemon] 3.1.4
[nodeemon] to restart at any time, enter 'rs'
[nodeemon] watching path(s): *.*
[nodeemon] watching extensions: js,mjs,cjs,json
[nodeemon] starting 'node app.js'
Listening on port 3000
```

5. POST /api/users: Adds a new user through form submission.

The screenshot shows a web browser at `localhost:3000/api/adduser` displaying a "Register Now" form. The form fields are: Name (Harry Potter), Email (theboywholived@gmail.com), Age (18), Salary (123456), and a Submit button.

The VS Code editor shows the `app.js` file with the following code:

```
13 app.get('/', (req, res) => {
14   res.send(routes)
15 })
16
17 app.get('/api/users', (req, res) => {
18   res.send(users);
19 });
20
21
22 app.get('/api/users/:id', (req, res) => {
23   const usersId = parseInt(req.params.id)
24   const user = users.find(u => u.id === usersId);
25
26   if(!user) res.status(404).send("The user with ID
27   is not existing or is not found.");
28
29   res.send(user)
30 });
31
32 app.get('/api/users/:id:name', (req, res) => {
33   res.send(req.params);
34 });
35
36 app.get('/api/adduser', (req, res) => {
37   res.sendFile(path.join(__dirname, 'public',
38     'adduser.html'));
39 });
```

The terminal shows the following output:

```
[nodemon] 3.1.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node app.js`
Listening on port 3000
```

The screenshot shows a web browser at `localhost:3000/api/users` displaying a "Pretty-print" button and a list of users. The list contains 9 users, including the newly added user:

```
[{"id":1,"name":"Carmela","email":"mela@gmail.com","age":25,"salary":"25,000"},
{"id":2,"name":"Joseph","email":"joe@yahoo.com","age":30,"salary":"45,000"},
{"id":3,"name":"James","email":"james@msn.com","age":35,"salary":"30,000"},
{"id":4,"name":"John","email":"john@gmail.com","age":40,"salary":"25,000"},
{"id":5,"name":"Frank","email":"frank@yahoo.com","age":45,"salary":"45,000"},
{"id":6,"name":"Alex","email":"alex@msn.com","age":21,"salary":"33,000"},
{"id":7,"name":"Ethan","email":"ejgg@gmail.com","age":20,"salary":"00.00"},
{"id":8,"name":"Remove","email":"ejgg@gmail.com","age":20,"salary":"00.00"},
{"id":9,"name":"Harry Potter","email":"theboywholived@gmail.com","age":18,"salary":"123456"}]
```

The VS Code editor shows the same `app.js` file as above.

The terminal shows the following output:

```
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node app.js`
Listening on port 3000
A new user is added with id : 9
```

6. DELETE /api/delete/: Deletes a user by ID.

The screenshot shows a VS Code workspace with a REST client request and the server code. The REST client request is a DELETE request to `http://localhost:3000/api/delete/8`. The response is a 200 OK status with a message: `"message": "User with ID 8 deleted successfully."`. The server code is in `app.js` and includes the following logic:

```
app.post('/api/adduser', (req, res) => {
  let newUser = {
    id: users.length + 1,
    name: req.body.name,
    email: req.body.email,
    age: req.body.age,
    salary: req.body.salary
  }
  console.log("A new user is added with id : $
    {newUser.id}");
  users.push(newUser);
  res.redirect('/api/users');
});

app.delete('/api/delete/:id', (req, res) => {
  const usersid = parseInt(req.params.id)
  const userIndex = users.findIndex(u => u.id ===
    usersid);
  users.splice(userIndex, 1);
  res.send({ message: 'User with ID ${usersid}
    deleted successfully.' });
});

app.listen(3000, () => {
  console.log('listening on port 3000');
});
```

The screenshot shows the server code in `app.js` and the terminal output. The server code is the same as in the previous screenshot. The terminal output shows the following messages:

```
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node app.js`
listening on port 3000
A new user is added with id : 9
```

STACKBLITZ LINK:

<https://stackblitz.com/edit/stackblitz-starters-flwzt7?file=API-Endpoints%2Fapp.js>