Question 1

Mistake 1:

There are two APIs that use the path parameters:

- **/api/users/:imageId**
- **/api/users/:userId**

You can get users by image or user ID. This creates ambiguity since we don’t know if a request like GET /api/users/5 will fetch the users by image IDs that match 5, or users with user ID 5. We can fix this ambiguity by using more distinguishable paths. We can use imageId as a parameter for something else like /api/images/:imageId, so there’s no conflict.

Mistake 2:

Here, we are using the GET method to log a user out.

```
app.get("/api/users/logout", (req, res)
```

We don’t have to request any resources when logging out. A more appropriate method like DELETE for removing resources should be used to log a user out.

Mistake 3:

We aren’t supposed to request a body for GET methods according to the API specification for GET.

```
app.get("/api/users/images", (req, res) => {
   let currentUser = req.body.userId;
```

We could pass userId as a query parameter instead.

Mistake 4:

The only error we catch is a SQL_CONNECTION_ERROR.

```
/api/users/:userId/images
…
catch(error) { if(error.type == "SQL_CONNECTION_ERROR")
```

We can fix this by handling all possible errors.

Question 2:

Path: /api/employees/:id
Method: GET
Body: N/A
Description: Display an employee from the database based on ID
Responses:

- 404
    - Description: Employee with given ID not found

      ```
      {
         "error": "No such employee"
      ```

```
}
```

- 200
    - Description: Successfully fetch employee

      ```
      {
         "address": {
            "street": "456 Elm St",
            "city": "Metropolis",
            "province": "NY",
            "zip": "10001"
         },
         "emergencyContact": {
            "name": "John Smith",
            "relationship": "Husband",
            "phone": "555-765-4321"
         },
         "_id": "666dce53a537510a1ef36e65",
         "firstName": "Jane",
         "lastName": "Smith",
         "dateOfBirth": "1975-06-15T00:00:00.000Z",
         "contactNumber": "555-123-4567",
         "email": "jane.smith@example.com",
         "role": "Doctor",
         "department": "Pediatrics",
         "employeeID": "EMP67890",
         "officeLocation": "Building B, Room 202",
         "notes": "Dr. Smith has over 20 years of experience in pediatrics!\n",
         "createdAt": "2024-06-15T17:24:35.489Z",
         "updatedAt": "2024-06-16T22:11:42.244Z",
         "__v": 0
      }
      ```
- 500
    - Description: Server Error, couldn't get the employee

      ```
      {
        "error": "Failed to fetch employee"
      }
      ```

Path: /api/employees/
Method: POST
Body:

- **firstName**
    - String, The employee's first name
- **lastName**
    - String, The employee's last name
- **dateOfBirth**

- Date, The date of employee’s birth

- **contactNumber**
  - String, Employee’s phone number
- **Email**
  - String, Employee’s email address
- **Address**
  - Description: Employee’s address
  - **Street**
    - String, street name
  - **City**
    - String, name of city
  - **Province**
    - String, name of province
  - **zip**
    - String, zip code
- **emergencyContact**
  - Description: Emergency contact info for employees
  - **name**
    - String, name of the contact
  - **relationship**
    - String, relation between employee and contact
  - **phone**
    - string, phone number of contact
- **role**
  - String, role of employee in the hospital (doctor, nurse, administrative)
- **department**
  - String, Department the employee works in
- **employeeID**
  - String,
- **officeLocation**
  - String, location of employee’s office
- **notes**
  - String,

Description: Create a new employee

Responses:

- 201
  - Description: successfully added new employee

    ```
    {
      "_id": "60d21b4667d0d8992e610c85",
      "firstName": "John",
      "lastName": "Doe",
      "dateOfBirth": "1990-01-01T00:00:00.000Z",
      "contactNumber": "123-456-7890",
      "email": "john.doe@example.com",
    ```

```
 "address": "123 Main St, Anytown, USA",
 "emergencyContact": "Jane Doe - 123-456-7890",
 "role": "Nurse",
 "department": "Pediatrics",
 "employeeID": "E001",
 "officeLocation": "Building A, Room 101",
 "notes": "Notes about John Doe"
}
```

- 400
    - Description: bad request, new employee not created

      ```
      {
       "error": "Missing required fields"
      }
      ```

Path: /api/employees/:id
Method: DELETE
Body: N/A
Description: Remove the employee with the given ID from the database
Responses:

- 404
    - Description: Employee with given ID not found

      ```
      {
         "error": "No such employee"
      }
      ```
- 200
    - Successfully removed employee

      ```
      {
       "message": "Employee deleted successfully"
      }
      ```

Path: /api/employees/:id
Method: PATCH
Body:

- **firstName**
    - String, The employee’s first name
- **lastName**
    - String, The employee’s last name
- **dateOfBirth**
    - Date, The date of employee’s birth
- **contactNumber**
    - String, Employee’s phone number
- **Email**
    - String, Employee’s email address
- **Address**

  - Description: Employee’s address
  - **Street**
    - String, street name
  - **City**
    - String, name of city
  - **Province**
    - String, name of province
  - **zip**
    - String, zip code
- **emergencyContact**
  - Description: Emergency contact info for employees
  - **name**
    - String, name of the contact
  - **relationship**
    - String, relation between employee and contact
  - **phone**
    - string, phone number of contact
- **role**
  - String, role of employee in the hospital (doctor, nurse, administrative)
- **department**
  - String, Department the employee works in
- **employeeID**
  - String,
- **officeLocation**
  - String, location of employee’s office
- **notes**
  - String,

Description: Update the information of the employee with the given ID

Responses:

- 404
  - Description: Employee with given ID not found
    ```
    {
       "error": "No such employee"
    }
    ```
- 200
  - Successfully updated employee
    ```
    {
      "message": "Employee info updated successfully"
    }
    ```