National University of Computer and Emerging Sciences

Lahore Campus

0023

Web Engineering Lab (SL3003)

Date: May 22nd 2025

Instructor(s)

Mr. Ahmad Jawad Mustasim
Mr. Muhammad Hassan Raza
Ms. Sana Ejaz

Final Exam

Total Time (Hrs):	**2.5**
Total Marks:	**60**
Total Questions:	**2**

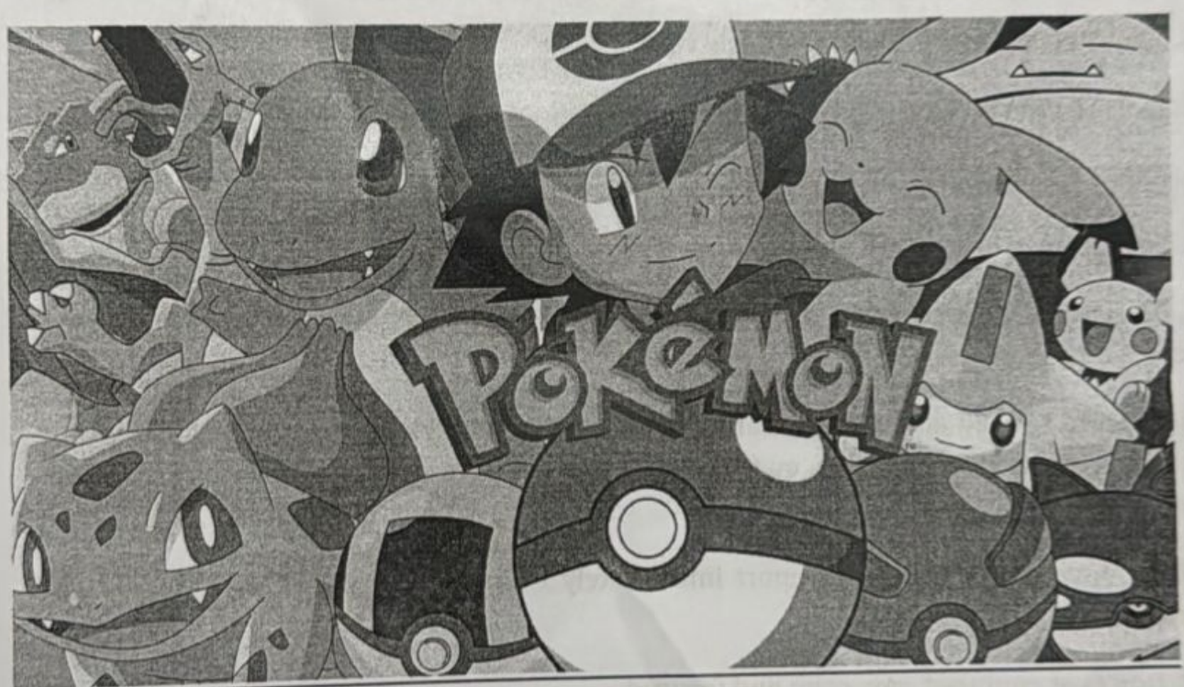
22L-7983 Roll No

BSE-6A Section

Student Signature

Do not write below this line

Build a Pokémon Website



Exam and Submission instructions: READ ALL INSTRUCTIONS CAREFULLY.

1. Understanding the question is also part of the exam, so do not ask for any clarification. Make suitable ASSUMPTIONS in case of any issues.

2. Final Submissions should be done on xeon. (\\exam) in your respective folder. Question must be saved with your roll no e.g: "**22L-1234.zip**". Multiple submissions are not allowed (if done, only the first one will be considered). In case of missing or corrupt file submission, all the responsibility will be on the student himself.

3. Submission should be in in zip folder. Zip folder should contain folders for screenshots, frontend folder and backend folder. Don't submit node modules. Remove node modules folder from frontend and backend before submitting.

4. Your cell phones/smart watches MUST be turned off and placed far away from the PCs.

5. It is your responsibility to protect your code and save it from being copied. If you don't protect it, all matching codes will be considered copy/cheating cases. No leniency on plagiarism.

6. This is an closed book paper and internet is not allowed in this exam. You are required to make sure you delete everything from the Desktop of your windows. Also delete all the files permanently from Recycle Bin and Trash respectively for Windows, no student should be found using any means of code sharing.

7. Submission should also have screenshot of the output. Even if your work isn't working submit screenshot of errors because it carries marks.

8. Your final exam will be immediately cancelled if:

a. You are seen talking, whispering, borrowing or looking at someone's PC.

b. A USB is found attached to your PC.

c. You are seen using a cell phone/smart watch.

9. Plagiarism will be checked during the evaluation, even if your systems had no internet access during the exam. So don't try to be over smart.

10. Ensure to read all questions carefully and their marks division, attempt your exam accordingly. Don't leave any question try to attempt everything according to your understanding

11. If found any issue in PC report immediately to invigilator, lab coordinator or lab support.

12. Don't get confused, stay calm and focused.

13. Weightage of final exam is 40%.

BEST OF LUCK ☺

Attempt all questions.

Part 1: Backend *(30 marks)*

1. Ensure correct file structure (models, controllers, routes). ***(1.5 marks)***
2. Use MongoDB Compass/ Mongodb for VS Code (VS Code Extension) for database connection with express js. ***(1.5 marks)***
3. Create following 2 collections: ***(3 marks)***

   Trainer
   - name (String – should be unique)
   - region (String)
   - password (String) (should be stored encrypted)

   Pokemon
   - name (String)
   - type (String)
   - level (number)
   - trainerId (References the Trainer collection)

4. Create Register Trainer API. Password should be encrypted ***(4 marks)***

   Insert following data using Postman/ Thunder Client (also take screenshot of successful insert of at least 1 trainer):

   Trainer 1: Ash
   ```
   {
    "name": "Ash",
    "region": "Kanto",
    "password": "ash123"
   }
   ```

   Trainer 2: Brock
   ```
   {
    "name": "Brock",
    "region": "Pewter",
    "password": "brock123"
   }
   ```

   Trainer 3: Misty
   ```
   {
    "name": "Misty",
    "region": "Cerulean",
    "password": "misty123"
   }
   ```

5. Create Get all Trainer API (anyone can see registered Trainers) ***(2 marks)***
6. Create login Trainer API. Do session handling by using JWT (implement middleware) ***(3 marks)***

7. Create Add Pokemon API. Pokemon of a Trainer can only be added after his/her login. ***(4 marks)***
Insert following data using POSTMAN/ Thunder Client (also take screenshot of successful insert of at least 1 pokemon):

Ash's Pokémons

```
{
 "name": "Pikachu",
 "type": "Electric",
 "level": 25,
}
{
 "name": "Charizard",
 "type": "Fire",
 "level": 36,}
{
 "name": "Bulbasaur",
 "type": "Grass",
 "level": 20,
}
```

Brock's Pokémons

```
{
 "name": "Onix",
 "type": "Rock",
 "level": 30,
}
```

Misty's Pokémons

```
{
 "name": "Staryu",
 "type": "Water",
 "level": 22,
}
```

8. Create Get all pokemons API (should also be working without login) ***(2 marks)***
9. Create Get API to fetch all pokemons of specific Trainer (Trainer._id should be passed in params) (should also be working without login) ***(3 marks)***
10. Test your all 6 APIs (registerTrainer, loginTrainer, addPokemon, getTrainers, getPokemons, getTrainerPokemons) on Postman/ Thunder Client and attach screenshots ***(6 marks)***

Part 2: Frontend - React JS *(30 marks)*

1. Create React Routes for following pages/ components:
   - Home Page: Show message like 'Welcome to World of Pokemon, website developed by Your Name - Your Roll No' ***(1 mark)***
   - Get All Pokemons Page (fetch part 1's API using axois) ***(2 mark)***
   - Get All Trainers Page (fetch part 1's API using axois) ***(2 mark)***

2. Add Navbar Component showing all these routes. ***(3 marks)***

3. Add 404 Page for unknown routes. ***(1 mark)***

4. Implement Nested and Dynamic Routing for Trainer Detail Page:
   When a user clicks on a specific trainer from the "All Trainers" page, it should navigate to a dedicated page showing the details and Pokémons of that trainer using dynamic route parameter. (path will be "/trainer/:id") ***(4 marks)***

5. In All Pokemons page, add tab/button to highlight pokemon with highest level (use useRef for highlight) ***(4 marks)***

6. Apply basic CSS styling for a neat and clean look. ***(2 mark)***

7. Navbar and layout should also be responsive ***(3 marks)***

8. Add screenshots of your all 4 pages (home, all pokemons, all trainer, trainer specific- any one trainer page screenshot) ***(8 marks)***

Marks Distribution (Total 60)

Task	Description	Marks
Part 1 (Backend)		**30**
File Structure, DB Connection & DB Collections	Proper separation (models, controllers, routes)	6
Register Trainer API	Proper API, validation, and error handling, encryption of password	4
Get All Trainers API	Functional and returns data properly	2
Login Trainer	session handling middleware using JWT	3
Add Pokemon API	Functional, Should associate with trainer	4
Get All Pokemons API	Functional	2
Get Pokemons Trainer-wise	Query filtering correctly	3
Postman/ Thunder Client Screenshots	Proves testing of APIs	6
Part 2 (Frontend)		**30**
Home page, Pokemons page, Trainers page	Fetching all data correctly using axios + Personalized message shown (Name + Roll No)	5
Routes Setup (React Router) + Navbar component	All required pages working	3
404 page	Properly shown for wrong routes	1
Dynamic/Nested Routing	Pokemons inside trainer-specific page	4
Highlight highest level pokemon	Logic + useRef highlight	4
CSS Styling + Responsiveness	Clean and readable layout + Responsive navbar and layout	5
Screenshots	Your frontend pages	8