Express.js Lab Marking Criteria

Task 1: Beyblade Battle Arena API (10 marks)

API Setup (2 marks)

- Initialize Express.js application (1 mark)
- Create data structure for Beyblade information (1 mark)

Route Implementation (6 marks)

- GET /beyblades (1 mark)
- GET /beyblades/:name (1 mark)
- POST /battle (2 marks)
    - Accept two Beyblade names (0.5 mark)
    - Simulate battle based on stats (1 mark)
    - Return winner and battle details (0.5 mark)
- PUT /beyblades/:name/upgrade (2 marks)
    - Upgrade Beyblade's stats (1 mark)
    - Return updated Beyblade information (1 mark)

Error Handling (2 marks)

- Implement basic error handling (1 mark)
- Handle missing entries with appropriate messages (1 mark)

Task 2: Power Rangers Team Assemble! (10 marks)

1. Route Implementation (8 marks)
    - GET /rangers (2 marks)
        - Return list of all Rangers (1 mark)
        - Include name, color, and special ability (1 mark)
    - POST /teams (2 marks)
        - Create new team with name and Rangers (1 mark)
        - Validate Rangers' existence and availability (1 mark)
    - GET /teams/:teamName (2 marks)
        - Retrieve team details (1 mark)
        - Include team name, members, and power level (1 mark)
    - DELETE /teams/:teamName (2 marks)
        - Disband team (1 mark)
        - Return confirmation message (1 mark)
2. Error Handling (2 marks)
    - Implement basic error handling (1 mark)

    - Handle missing entries with appropriate messages (1 mark)

Task 3: Galactic Space Station Management API (20 marks)

1. API Setup and Data Structures (3 marks)

- Initialize Express.js application (1 mark)
- Create appropriate data structures for: (2 marks)
    - Space station information (0.5 mark)
    - Inhabitants (0.5 mark)
    - Resources (0.25 mark)
    - Research projects (0.25 mark)
    - Security protocols (0.25 mark)
    - Maintenance schedules (0.25 mark)

2. Route Implementation (14 marks)

- GET /station-info (1 mark)
    - Return correct information (name, capacity, current population) (1 mark)
- GET /inhabitants (1 mark)
    - Return list of all inhabitants with correct information (1 mark)
- PUT /station-info (2 marks)
    - Update name, description, and maximum capacity (1 mark)
    - Ensure capacity reduction doesn't go below current population (1 mark)
- PUT /inhabitants/:name (3 marks)
    - Update purpose of stay and assigned quarters (1 mark)
    - Update access level (1 mark)
    - Implement verification system for access level updates (1 mark)
- PUT /resources/:resourceName (2 marks)
    - Update current quantity, maximum capacity, and criticality level (2 marks)
- PUT /research-projects/:projectName (2 marks)
    - Update project status, lead researcher, allocated resources, and priority level (2 marks)
- PUT /security-protocols (1.5 marks)
    - Update access levels, restricted areas, and emergency procedures (1.5 marks)
- PUT /maintenance-schedule (1.5 marks)
    - Allow rescheduling, crew assignments, and task priority adjustments (1.5 marks)

3. Error Handling (3 marks)

- Implement basic error handling for invalid requests (1 mark)
- Use appropriate error codes (e.g., 404 for "not found") (1 mark)
- Handle missing entries with appropriate messages to the user (1 mark)

Demo Questions (-2 to 0 marks)

For each task, assess the student's ability to answer demo questions:

- Cannot answer at all: -2 marks
- Answers incorrectly: -1 mark
- Answers correctly: 0 marks (no deduction)

Total possible deduction: -6 marks (2 per task)

Total Marks

- Task 1: 10 marks
- Task 2: 10 marks
- Task 3: 20 marks
- Demo Questions: Up to -6 marks deduction

Total: 40 marks