CA1 Individual Report

|  |  |
| --- | --- |
| Name | Tan Eu Zin |
| Student Id | P2415927 |
| Class | DAAA/FT/1B/03 |
| Github Repository URL | <https://github.com/ST0503-BED/bed-ca1-zin246> |
| Github Account ID | zin246 |

For each competencies, find links to pull requests/commits/files that demonstrate the completion of the requirement. Replace each “**?**” with your Self Rating.

For Self Rating, you may rate yourself accordingly if you feel that you:

1. Have little or **no** understanding. and did not attempt the requirement
2. Have **limited** understanding of the specific competency
3. Have **basic** understanding and only able to replicate examples from tutorials/practicals.
4. Have **adequate** understanding and can extend from what you have learned to fulfil specifications.
5. Have **solid** understanding in the specific competency, able work on the requirement without much references.
6. Have **excellent** understanding and implemented the requirement according to latest industry guidelines, best practices and documentations.

**Important**

1. You are require to provide for each competency:
   * A brief **description**
   * **One or two** of your best implementations with URL **link** to respective repository request/commits/files.
     + Below is an example of linking to specific script file

<https://github.com/expressjs/express/blob/master/examples/static-files/index.js>

* + **The implementations may come from Section A or B.**

1. You are to ensure the hyperlink in this document works. **Failure to do so will result in a 50% deduction of marks.**
2. You are to **provide screenshots using POSTMAN** to perform API test**.**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Competencies | Describe What Was Done | Self Rating |
| 1 | Architecture | *(How you organize and structure of your project's codebase?)*  **The project is structured in model, route and controller format. This separates the project into three sections, model for SQL, route to set the routes of requests and controller to handle incoming data and errors that occur.** [**https://github.com/ST0503-BED/bed-ca1-zin246/tree/main/src**](https://github.com/ST0503-BED/bed-ca1-zin246/tree/main/src) **As seen in the src folder, there are three folders that separate the project into three parts, models, controllers and routes, which is how I structured the project.** | 4/5 |
| 2 | Dependency Management | *(How you manage the package.json, external dependencies and libraries used in your project? What scripts were written in your package.json and why?)*  **The package.json file is used to manage dependencies and scripts for project. There are also external dependencies and libraries used such as mysql2, nodemon and dotenv. I manage the package.json file by using the dev script to run my project by typing (npm run dev) in the terminal. There are also start, dev and init\_tables scripts written in the package.json file. This can be seen in the whole of the package.json file here :**  [**https://github.com/ST0503-BED/bed-ca1-zin246/blob/main/package.json**](https://github.com/ST0503-BED/bed-ca1-zin246/blob/main/package.json) | 3/5 |
| 3 | API Design | *(How well you adhere to RESTful conventions and provide consistent and well-documented endpoints?)*  **Http methods such as GET, PUT, POST, DELETE are used appropriately in my project, and each endpoint returns appropriate status codes. Furthermore, endpoints are clearly separated such as /heroes/equip and /heroes/unequip for example, as seen in the file, where the endpoints are meant for equipping and unequipping weapons, which are clearly stated in the routing. This ensures that endpoints are clear and well documented.**  [**https://github.com/ST0503-BED/bed-ca1-zin246/blob/main/src/routes/userHeroesRoutes.js**](https://github.com/ST0503-BED/bed-ca1-zin246/blob/main/src/routes/userHeroesRoutes.js) | 3/5 |
| 4 | Middleware Usage | *(How did you leverage on middleware functions for your application?)*  **I used middleware mostly to check for errors a part of error handling or check conditions before proceeding with some actions after verifying for suitable conditions. For example, as seen in –**  [**https://github.com/ST0503-BED/bed-ca1-zin246/blob/main/src/routes/battleRoutes.js**](https://github.com/ST0503-BED/bed-ca1-zin246/blob/main/src/routes/battleRoutes.js)  **looking at the first route for example, it uses one middleware to check if user exists, check if map and level exists, check if the specified level has been cleared before, and when all conditions are met (user, map and level exists and level has not been cleared) then it will use another middleware to fight the level and determine winner, then it will lastly mark the level as clearly and edit tables such that the level is marked to be cleared before.** | 4/5 |
| 5 | Database Design | *(What did you do to ensure effective data storage and retrieval? How did you normalise your tables?)*  **To ensure effective data storage, I stored related data in multiple tables that can be referred to separately so they do not become confuing. For example, map and levels are separately stored, but maps can be referred to using map\_id and levels can be referred to using map\_id and level. Furthermore, to normalise my tables, almost all tables have a indexes, such as challenge\_id, user\_id, inventory\_id, hero\_id and many more. This can be seen in my init\_tables file.**  [**https://github.com/ST0503-BED/bed-ca1-zin246/blob/main/src/configs/initTables.js**](https://github.com/ST0503-BED/bed-ca1-zin246/blob/main/src/configs/initTables.js) | 4/5 |
| 6 | SQL Queries | *(What was done to ensure smooth and responsive interactions with the database? Have you applied advances SQL Syntax?)*  **I ensured that my sql queries such as SELECT, INSERT, UPDATE only fetched or updated the necessary data to ensure smooth and responsive interactions with the database. Furthermore, I have also used subqueries to improve functionality of my sql code. These examples can be observed from my battleModel file.** [**https://github.com/ST0503-BED/bed-ca1-zin246/blob/main/src/models/battleModel.js**](https://github.com/ST0503-BED/bed-ca1-zin246/blob/main/src/models/battleModel.js)  **Especially in the levelComplete model, a subquery is used to update skillpoints.** | 4/5 |
| 7 | Functionality | *(Did your features implemented meets the specified requirements and fulfils its intended purpose?)*  **My features meet all the needs of the “user” playing the game. For example, as seen in -** [**https://github.com/ST0503-BED/bed-ca1-zin246/blob/main/src/routes/weaponShopRoutes.js**](https://github.com/ST0503-BED/bed-ca1-zin246/blob/main/src/routes/weaponShopRoutes.js)  **The user is allowed to view weapons and purchase weapons, these for example are all the needs of the user playing the game, purchasing a weapon from a shop, being fulfilled, and the code successfully does it which shows that the intended purpose is fulfilled.** | 4/5 |
| 8 | Code Quality | *(How did you organise your code to ensure maintainability, readability and adherence to coding best practices?)*  **Every file in my code has had their format edited using the prettier extension. This ensures that the code is readable. As seen in** [**https://github.com/ST0503-BED/bed-ca1-zin246/blob/main/src/controllers/battleController.js**](https://github.com/ST0503-BED/bed-ca1-zin246/blob/main/src/controllers/battleController.js)  **For example, every controller is also separated at the top and bottom with comments so that each controller is easy to view and identify. There are also comments used to label the controllers for each step so it is easy to identify which controllers are for which step.** | 4/5 |
| 9 | Modularity | *(How did you've organized your project to promote code reusability and maintainability?)*  **Some of the code (mostly middleware) used for error handling such as checking if user exists for example, can be used repeatedly for different endpoints. This can be seen in** [**https://github.com/ST0503-BED/bed-ca1-zin246/blob/main/src/routes/userHeroesRoutes.js**](https://github.com/ST0503-BED/bed-ca1-zin246/blob/main/src/routes/userHeroesRoutes.js)  **checkUserExistence is an example of such, which has been used throughout all 5 endpoints inside userHeroes files to check if user exists, used for error handling** | 3/5 |
| 10 | Error Handling | *(How did you manage errors, provide informative feedback, and handle exceptional situations? What additional scenarios did you considered for your error handling?)*  **To manage errors, I have used middleware to check for errors, and each error will return a different message from my code. For example, in the file below, it can be observed that checkUserExistence will return a specific error message to let the user know the error, checkMapAndLevel will also return another error if map or level or both does not exist. These can all be observed in the file below.**  [**https://github.com/ST0503-BED/bed-ca1-zin246/blob/main/src/controllers/battleController.js**](https://github.com/ST0503-BED/bed-ca1-zin246/blob/main/src/controllers/battleController.js) | 4/5 |
| 11 | Documentation | *(What was written for comments, readme and external documentation?)*  **Comments are mostly written to explain what parts of the project does so that both me and the person reading my code can easily understand what each part is for. Comments are also used to separate out parts of code for readability and to label them. For example in** [**https://github.com/ST0503-BED/bed-ca1-zin246/blob/main/src/controllers/userInventoryController.js**](https://github.com/ST0503-BED/bed-ca1-zin246/blob/main/src/controllers/userInventoryController.js) **comments are used to label the file as part of Section B, label the controllers for each endpoint and state what they are for, and also to separate out every single controller so it is easy to identify each controller and makes it look neat. Furthermore, DONE is written below for me to make a mental note that I am done with a file, and also for the reader to know that that is the end of the file.** | 4/5 |

**Entity Relationship Diagram**

|  |
| --- |
| ERD |
|  |

**Screenshots**

|  |  |  |
| --- | --- | --- |
| No. | Label | Screenshot |
| 1 | **Section A : 1**  **POST**  **/users**  **201** |  |
| 2 | **POST**  **/users**  **409** |  |
| 3 | **POST**  **/users**  **400** |  |
| 4 | **Section A : 2**  **GET**  **/users**  **200** |  |
| 5 | **Section A : 3**  **PUT**  **/users**  **/{user\_id}**  **200** |  |
| 6 | **PUT**  **/users**  **/{user\_id}**  **404** |  |
| 7 | **PUT**  **/users**  **/{user\_id}**  409 |  |
| 8 | **Sectoin A : 4**  **POST**  **/challenges**  **201** |  |
| 9 | **POST**  **/challenges**  **400** |  |
| 10 | **Section A : 5**  **GET**  **/challenges**  **200** |  |
| 11 | **Section A : 6**  **PUT**  **/challenges**  **/{challenge\_id}**  **200** |  |
| 12 | **PUT**  **/challenges**  **/{challenge\_id}**  **404** |  |
| 13 | **PUT**  **/challenges**  **/{challenge\_id}**  **400** |  |
| 14 | **PUT**  **/challenges**  **/{challenge\_id}**  **403** |  |
| 15 | **Section A : 7**  **DELETE**  **/challenges**  **/{challenge\_id}**  **204** |  |
| 16 | **DELETE**  **/challenges**  **/{challenge\_id}**  **404** |  |
| 17 | **Section A : 8**  **POST**  **/challenges**  **/{challenge\_id}**  **201** |  |
| 18 | **POST**  **/challenges**  **/{challenge\_id}**  **404** |  |
| 19 | **POST**  **/challenges**  **/{challenge\_id}**  **400** |  |
| 20 | **Section A : 9**  **GET**  **/challenges**  **/{challenge\_id}**  **200** |  |
|  | **GET**  **/challenges**  **/{challenge\_id}**  **404** |  |
| 21 | **Section B : Character : 1**  **GET**  **/characters**  **200** |  |
| 22 | **Section B : Character : 2**  **GET**  **/characters**  **/{class}**  **200** |  |
| 23 | **GET**  **/characters**  **/{class}**  **404** |  |
| 24 | **Section B : Character : 3**  **POST**  **/characters**  **/{user\_id}**  **200** |  |
| 25 | **POST**  **/characters**  **/{user\_id}**  **404** |  |
| 26 | **POST**  **/characters**  **/{user\_id}**  **409** |  |
| 27 | **Section B : WeaponShop :**  **1**  **GET**  **/shop**  **200** |  |
| 28 | **Section B : WeaponShop : 2**  **GET**  **/shop**  **/{class}**  **200** |  |
| 29 | **GET**  **/shop**  **/{class}**  **404** |  |
| 30 | **Section B : WeaponShop : 3**  **POST**  **/shop**  **/{user\_id}**  **201** |  |
| 31 | **POST**  **/shop**  **/{user\_id}**  **404** |  |
| 32 | **POST**  **/shop**  **/{user\_id}**  **404** |  |
| 33 | **POST**  **/shop**  **/{user\_id}**  **400** |  |
| 34 | **POST**  **/shop**  **/{user\_id}**  **409** |  |
| 35 | **Section B : UserHeroes : 1**  **GET**  **/heroes**  **/{user\_id}**  **200** |  |
| 36 | **GET**  **/heroes**  **/{user\_id}**  **404** |  |
| 37 | **Section B : UserHeroes : 2**  **PUT**  **/heroes**  **/{user\_id}**  **/{hero\_id}**  **200** |  |
| 38 | **PUT**  **/heroes**  **/{user\_id}**  **/{hero\_id}**  **400** |  |
| 39 | **PUT**  **/heroes**  **/{user\_id}**  **/{hero\_id}**  **404** |  |
| 40 | **PUT**  **/heroes**  **/{user\_id}**  **/{hero\_id}**  **404** |  |
| 41 | **Section B : UserHeroes : 3**  **PUT**  **/heroes**  **/equip**  **/{user\_id}**  **/{hero\_id}**  **201** |  |
| 42 | **PUT**  **/heroes**  **/equip**  **/{user\_id}**  **/{hero\_id}**  **400** |  |
| 43 | **PUT**  **/heroes**  **/equip**  **/{user\_id}**  **/{hero\_id}**  **404** |  |
| 44 | **PUT**  **/heroes**  **/equip**  **/{user\_id}**  **/{hero\_id}**  **404** |  |
| 45 | **PUT**  **/heroes**  **/equip**  **/{user\_id}**  **/{hero\_id}**  **404** |  |
| 46 | **PUT**  **/heroes**  **/equip**  **/{user\_id}**  **/{hero\_id}**  **400** |  |
| 47 | **PUT**  **/heroes**  **/equip**  **/{user\_id}**  **/{hero\_id}**  **404** |  |
| 48 | **Section B : UserHeroes : 4**  **PUT**  **/heroes**  **/unequip**  **/{user\_id}**  **/{hero\_id}**  **201** |  |
| 49 | **PUT**  **/heroes**  **/unequip**  **/{user\_id}**  **/{hero\_id}**  **404** |  |
| 50 | **PUT**  **/heroes**  **/unequip**  **/{user\_id}**  **/{hero\_id}**  **404** |  |
| 51 | **PUT**  **/heroes**  **/unequip**  **/{user\_id}**  **/{hero\_id}**  **404** |  |
| 52 | **PUT**  **/heroes**  **/unequip**  **/{user\_id}**  **/{hero\_id}**  **400** |  |
| 53 | **Section B : UserHeroes : 5**  **DELETE**  **/heroes**  **/{user\_id}**  **/{hero\_id}**  **200** |  |
| 54 | **DELETE**  **/heroes**  **/{user\_id}**  **/{hero\_id}**  **404** |  |
| 55 | **DELETE**  **/heroes**  **/{user\_id}**  **/{hero\_id}**  **404** |  |
| 56 | **Section B : UserInventory : 1**  **GET**  **/inventory**  **/{user\_id}**  **200** |  |
| 57 | **GET**  **/inventory**  **/{user\_id}**  **404** |  |
| 58 | **GET**  **/inventory**  **/{user\_id}**  **404** |  |
| 59 | **Section B : UserInventory : 2**  **DELETE**  **/inventory**  **/{user\_id}**  **/{inventory\_id}**  **204** |  |
| 60 | **DELETE**  **/inventory**  **/{user\_id}**  **/{inventory\_id}**  **404** |  |
| 61 | **DELETE**  **/inventory**  **/{user\_id}**  **/{inventory\_id}**  **404** |  |
| 62 | **Section B : UserInventory : 3**  **PUT**  **/inventory**  **/{user\_id}**  **/{inventory\_id}**  **200** |  |
| 63 | **PUT**  **/inventory**  **/{user\_id}**  **/{inventory\_id}**  **400** |  |
| 64 | **PUT**  **/inventory**  **/{user\_id}**  **/{inventory\_id}**  **404** |  |
| 65 | **PUT**  **/inventory**  **/{user\_id}**  **/{inventory\_id}**  **404** |  |
| 66 | **PUT**  **/inventory**  **/{user\_id}**  **/{inventory\_id}**  **404** |  |
| 67 | **PUT**  **/inventory**  **/{user\_id}**  **/{inventory\_id}**  **409** |  |
| 68 | **PUT**  **/inventory**  **/{user\_id}**  **/{inventory\_id}**  **409** |  |
| 69 | **PUT**  **/inventory**  **/{user\_id}**  **/{inventory\_id}**  **409** |  |
| 70 | **PUT**  **/inventory**  **/{user\_id}**  **/{inventory\_id}**  **409** |  |
| 71 | **Section B : level : 1**  **GET**  **/level**  **200** |  |
| 72 | **Section B : level : 2**  **GET**  **/level**  **/{map\_id}**  **200** |  |
| 73 | **GET**  **/level**  **/{map\_id}**  **404** |  |
| 74 | **Section B : UserLineup : 1**  **POST**  **/lineup**  **/{user\_id}**  **/{hero\_id}**  **200** |  |
| 75 | **POST**  **/lineup**  **/{user\_id}**  **/{hero\_id}**  **404** |  |
| 76 | **POST**  **/lineup**  **/{user\_id}**  **/{hero\_id}**  **404** |  |
| 77 | **POST**  **/lineup**  **/{user\_id}**  **/{hero\_id}**  **409** |  |
| 78 | **POST**  **/lineup**  **/{user\_id}**  **/{hero\_id}**  **409** |  |
| 79 | **Section B : UserLineup : 2**  **DELETE**  **/lineup**  **/{user\_id}**  **/{hero\_id}**  **200** |  |
| 80 | **DELETE**  **/lineup**  **/{user\_id}**  **/{hero\_id}**  **400** |  |
| 81 | **DELETE**  **/lineup**  **/{user\_id}**  **/{hero\_id}**  **404** |  |
| 82 | **DELETE**  **/lineup**  **/{user\_id}**  **/{hero\_id}**  **404** |  |
| 83 | **Section B : UserLineup : 3**  **GET**  **/lineup**  **/{user\_id}**  **200** |  |
| 84 | **GET**  **/lineup**  **/{user\_id}**  **404** |  |
| 85 | **Section B : Battle : 1**  **PUT**  **/battle**  **/fight\_level**  **/{user\_id}**  **200**  **(battle won)** |  |
| 86 | **PUT**  **/battle**  **/fight\_level**  **/{user\_id}**  **400** |  |
| 87 | **PUT**  **/battle**  **/fight\_level**  **/{user\_id}**  **404** |  |
| 88 | **PUT**  **/battle**  **/fight\_level**  **/{user\_id}**  **404** |  |
| 89 | **PUT**  **/battle**  **/fight\_level**  **/{user\_id}**  **404** |  |
| 90 | **PUT**  **/battle**  **/fight\_level**  **/{user\_id}**  **400**  **(battle lost)** |  |
|  | **Section B : battle : 2**  **PUT**  **/battle**  **/fight\_player**  **/{user\_id}**  **200**  **(battle won)** |  |
|  | **PUT**  **/battle**  **/fight\_player**  **/{user\_id}**  **400** |  |
|  | **PUT**  **/battle**  **/fight\_player**  **/{user\_id}**  **404** |  |
|  | **PUT**  **/battle**  **/fight\_player**  **/{user\_id}**  **404** |  |
|  | **PUT**  **/battle**  **/fight\_player**  **/{user\_id}**  **400**  **(battle lost)** |  |