

Week 9 Project Presentation

Character creator API

INTRODUCTION

- The purpose of this project was to create a functional application that allows you to create characters with CRUD functionality.

MoSCoW Prioritisation

Editing Week-9-Hobby-Project-A

CRUD CharacterCreator

Week 9 Hobby Project - Roadma

Week-9-Hobby-Project-API/MoSC

MoSCoW Prioritisation Week 9 P

File | C:/Users/bmill/Downloads/MoSCoW%20Prioritisation%20Week3%209%20Project.pdf

MoSCoW Prioritisation Week 9 Project.pdf

1 / 3 | 100%

1

2

3

Visual Paradigm Online Free Edition

Must Have

Should Have

Could Have

Won't Have

1. A Kanban board with full expansion on user stories.

2. A record of issues experienced throughout the project.

3. A relational database used to store data persistently for the project. 3 tables to demonstrate understanding.

4. A functional application

5. Unit tests and integration tests for validation of the application. (Strive to reach coverage of 80%)

6. Code fully integrated into a Version Control System (GitHub) using the feature/branch/model.

7. A risk assessment which outlines the issues and risks faced during the project time frame.

7. Fully designated test suites for the application, as well as automated validation of the application.

8. The project as a

1. simple design provided meeting the agreed standards of patterns

2. Structured README.md with relevant content

3. Software is fully functional in all necessary areas (MVP) but still has small bugs and/or errors

4. Software has been tested in the most important areas, (judged by test coverage) with basic functions being tested. Best practises were adhered to for most areas of the project

5. Tools and workflows discussed in documentation are implemented throughout the project at an acceptable level, but more exploration could have occurred.

6. An acceptable level of

1. More extensive designs provided on both a system level and component level.

2. In depth README.md that explores the application of a high level.

3. Software has been tested in most relevant areas; best practises were consistently adhered to throughout the project.

4. Software is functional in all major areas but still has small bugs/errors.

5. Tools and workflows discussed in documentation are implemented throughout the project at a high level.

6. Analysis as to what was tested, what was out of scope and the reasons behind these decisions.

7. Summary of overall results is present.

8. A good level of test coverage met, utilising a broad range of tests.

9. A build server has been installed and used to produce a software

1. Due to time constraints, the project may not be 100% completed

2. Fully functioning application in the highest specification criteria

3. All bugs and/or errors completely removed from the code

4. In depth details within some parts of documentation.

5. A vast range of colours and different shapes and size customisation of buttons and boxes.

6. Multiple tables interlink through the back end.

Visual Paradigm Online Free Edition

MoSCoW Prioritisation Week 9 Project.pdf

100%

Show all

Began adding content to initial README.md

github.com/BillyMillsGitHub/Week-9-Hobby-Project-API

Search or jump to...

Pull requestsIssuesMarketplaceExplore

BillyMillsGitHub / Week-9-Hobby-Project-API

Private

Unwatch1Fork0Star0

<> CodeIssuesPull requests3ActionsProjectsSecurityInsightsSettings

main5 branches0 tags

Go to fileAdd fileCode

Your main branch isn't protected

Protect this branch from force pushing, deletion, or require status checks before merging.

Protect this branch

BillyMillsGitHub Update README.md

569aads yesterday4 commits

README.md

Update README.md

yesterday

README.md

Week-9-Hobby-Project-API

This github repository will store all documentation for my Week 9 Hobby Project.

This project consists of creating a fully functional API from scratch. This must be inclusive of a well presented and readable front end, a back end database with full CRUD functionality integration to the front end.

All applications used to develop this application within this project are:

- Version Control System ~ Git
- Source Code management ~ Github
- Kanban Board ~ Jira (Atlassian)
- Database ~ MySQL Server 5.7+ (local or cloud hosted)
- Back-end Development Platform ~ Java
- API Development Platform ~ Spring
- Front-End Web Technologies ~ HTML, CSS & JavaScript
- Build tool ~ Maven

The additional dependencies that were required for unit testing were:

- JUnit
- Mockito

About

No description, website, or topics provided.

Readme

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Example of all branches being developed on Github account

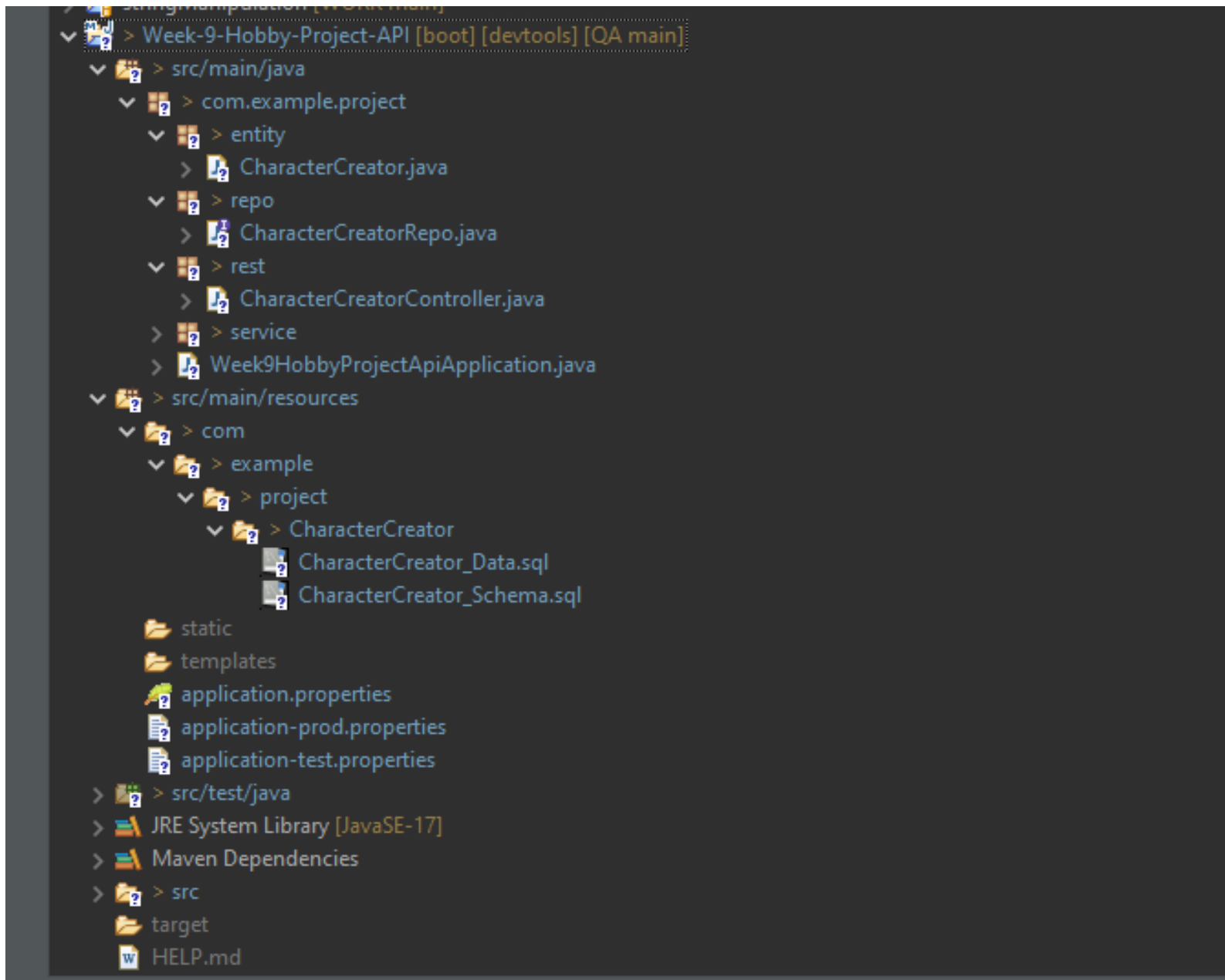
The screenshot shows the GitHub interface for the repository 'BillyMillsGitHub / Week-9-Hobby-Project-API'. The repository is private and has 5 branches and 0 tags. The 'documentation' branch is currently selected. A dropdown menu is open, showing a list of branches: 'main' (default), 'Back-end', 'Front-end', 'dev', and 'documentation' (checked). The dropdown also includes a search bar 'Find or create a branch...' and a link 'View all branches'.

The repository description states: 'This github repository will store all documentation for my Week 9 Project. This project consists of creating a fully functional API from scratch. This must be inclusive of a well presented and readable front end, a back end database with full CRUD functionality integration to the front end. All applications used to develop this application within this project are:'

- Version Control System ~ Git
- Source Code management ~ Github
- Kanban Board ~ Jira (Atlassian)
- Database ~ MySQL Server 5.7+ (local or cloud hosted)
- Back-end Development Platform ~ Java

The right sidebar shows the 'About' section with the text 'No description, website, or topics provided.' and the 'Releases' section with the text 'No releases published' and a link 'Create a new release'.

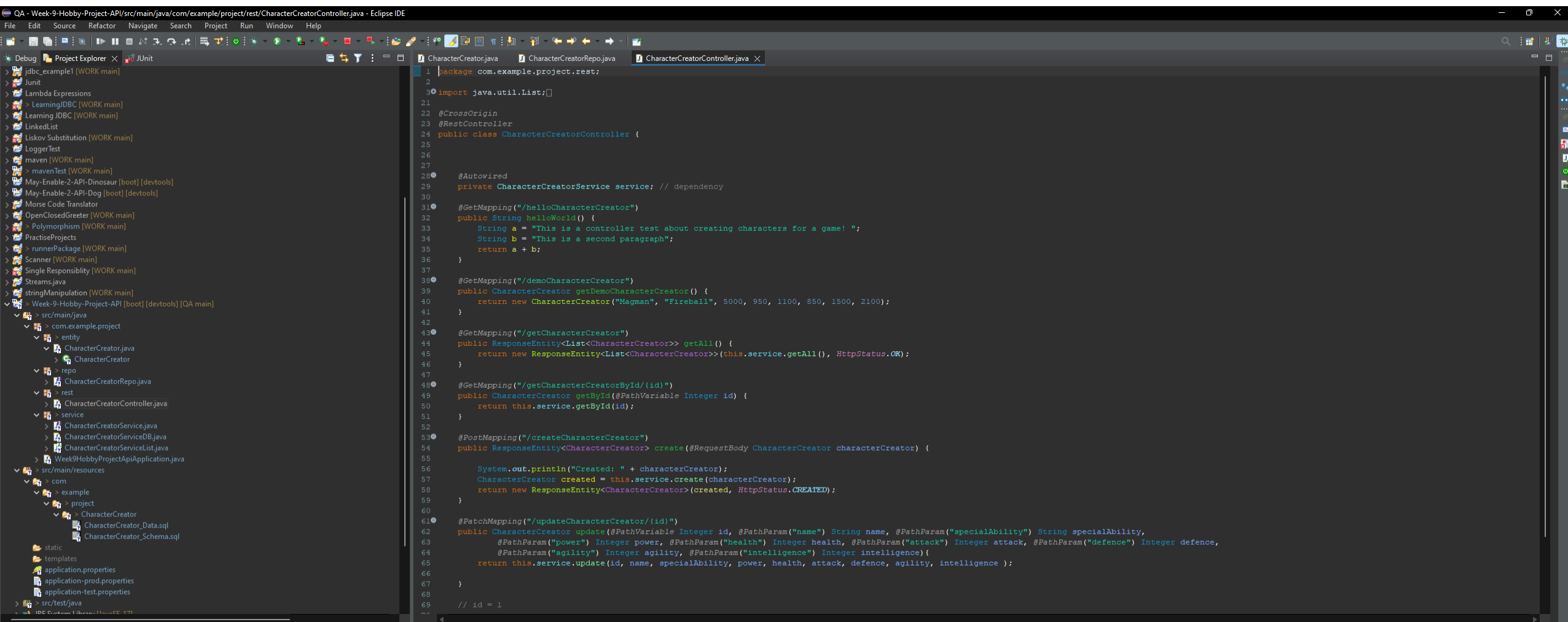
Created all classes required on Java Eclipse



Class with all attributes

```
1 package com.example.project.entity;
2
3 import javax.persistence.Column;
4
5
6
7
8
9
10 @Entity // Table
11 public class CharacterCreator {
12     @Id // PK
13     @GeneratedValue(strategy = GenerationType.IDENTITY) // AI
14     private Integer id;
15
16     @Column(name = "CharacterCreatorName", nullable = false, unique = true)
17     private String name;
18
19     private String specialAbility;
20
21     private Integer power;
22
23     private Integer health;
24
25     private Integer attack;
26
27     private Integer defence;
28
29     private Integer agility;
30
31     private Integer intelligence;
32
33     public CharacterCreator() {
34         super();
35         //Required superclass constructor
36     }
37
38     public CharacterCreator(String name, String specialAbility, Integer power, Integer health, Integer attack,
39         Integer defence, Integer agility, Integer intelligence) {
40         super();
41         this.name = name;
42         this.specialAbility = specialAbility;
43         this.power = power;
44         this.health = health;
45         this.attack = attack;
46         this.defence = defence;
47         this.agility = agility;
48         this.intelligence = intelligence;
49         //constructor without ID
50     }
51
52     public CharacterCreator(Integer id, String name, String specialAbility, Integer power, Integer health,
53         Integer attack, Integer defence, Integer agility, Integer intelligence) {
54         super();
55         this.id = id;
56         this.name = name;
57         this.specialAbility = specialAbility;
58         this.power = power;
59         this.health = health;
60         this.attack = attack;
61         this.defence = defence;
62         this.agility = agility;
63         this.intelligence = intelligence;
64     }
65 }
```

Controller class



Testing with coverage of 75.2%

The screenshot displays the Eclipse IDE interface. The main editor shows a Java test file, `CharacterCreatorControllerIntegrationTest.java`, with the following code:

```
63
64 @Test
65 void testReadById() throws Exception {
66     CharacterCreator readCharactercreator = new CharacterCreator (1, "Magman", "Fireball", 5000, 950, 1100, 850, 1500, 2100);
67     String createdCharactorCreatorAsJSON = this.mapper.writeValueAsString(readCharactercreator);
68
69     this.mvc.perform(get("/getCharacterCreatorById/1")).andExpect(content().json(createdCharactorCreatorAsJSON)).andExpect(status().isOk());
70 }
71
72
```

The left sidebar shows the Package Explorer with the project structure. The bottom status bar indicates the test run results: `Runs: 4/4`, `Errors: 0`, and `Failures: 0`. The bottom right pane shows the Coverage report for the test run.

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
Week-9-Hobby-Project-API	83.1 %	417	85	502
> src/main/java	75.2 %	245	81	326
> src/test/java	97.7 %	172	4	176

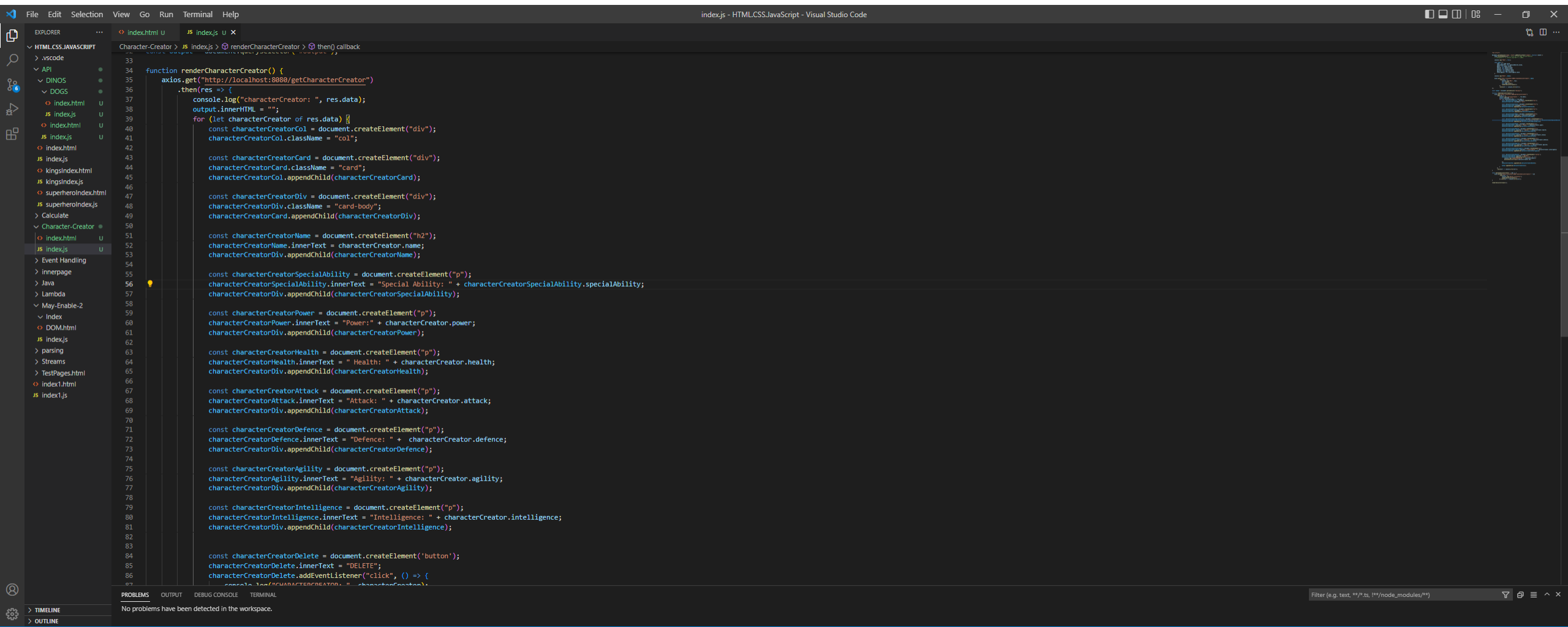
Front end styling in index.js

The screenshot shows the Visual Studio Code editor with the following components:

- Explorer Panel:** Displays the file structure of the project, including folders like .vscode, API, DINOS, DOGS, and files like index.html, index.js, kingsIndex.html, kingsIndex.js, superheroIndex.html, superheroIndex.js, Calculate, Character-Creator, Event Handling, innerpage, Java, Lambda, May-Enable-2, Index, DOM.html, parsing, Streams, TestPages.html, index1.html, and index1.js.
- Editor Panel:** Shows the content of the selected file, index.html. The code is as follows:

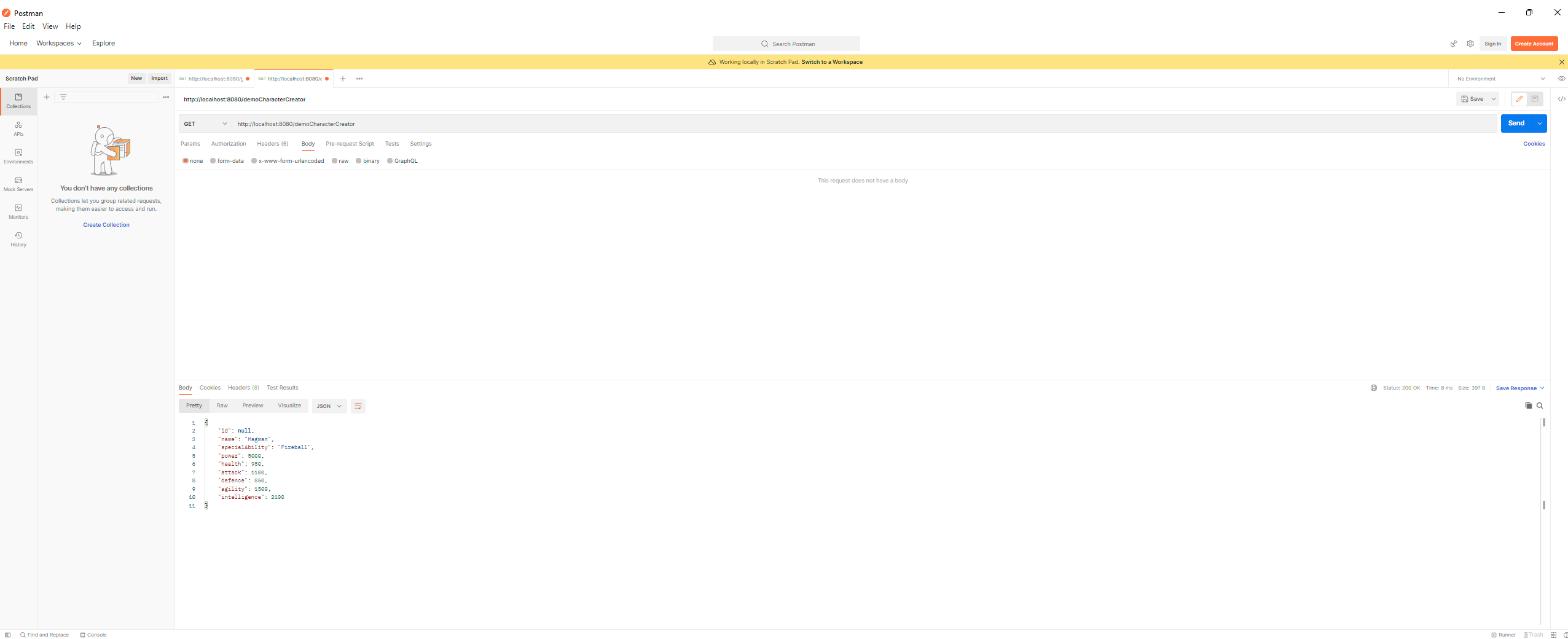
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>CRD CharacterCreator</title>
8   <script src="https://cdn.jsdelivr.net/npm/axios@1.0.0-alpha.1/axios.min.js" integrity="sha512-xIPqrrfVUAC/Cspuj7Bq0UtHw/5qkdyngx6Vwt+mbvTLDszzM0G6c91LXmGrRx8KEPuIT+Af00ez+TeVylg=="
9   crossorigin="anonymous" referrerpolicy="no-referrer"></script>
10 </head>
11 <body>
12
13
14   <form>
15     <label for="characterCreatorName">Name:</label>
16     <input type="text" name="name" id="characterCreatorName" required>
17     <br>
18     <label for="characterCreatorSpecialAbility">Special Ability:</label>
19     <input type="text" name="specialAbility" id="characterCreatorSpecialAbility" required>
20     <br>
21     <label for="characterCreatorPower">Power:</label>
22     <input type="text" name="power" id="characterCreatorPower" min="0" required>
23     <br>
24     <label for="characterCreatorHealth">Health:</label>
25     <input type="text" name="health" id="characterCreatorHealth" min="0" required>
26     <br>
27     <label for="characterCreatorAttack">Attack:</label>
28     <input type="text" name="attack" id="characterCreatorAttack" min="0" required>
29     <br>
30     <label for="characterCreatorDefence">Defence:</label>
31     <input type="text" name="defence" id="characterCreatorDefence" min="0" required>
32     <br>
33     <label for="characterCreatorAgility">Agility:</label>
34     <input type="text" name="agility" id="characterCreatorAgility" min="0" required>
35     <br>
36     <label for="characterCreatorIntelligence">Intelligence:</label>
37     <input type="text" name="intelligence" id="characterCreatorIntelligence" min="0" required>
38     <br>
39     <button type="reset">RESET</button>
40     <button type="submit">SUBMIT</button>
41
42
43   </form>
44   <div id="output"></div>
45   <script src="//index.js"></script>
46 </body>
47 </html>
```
- Problems Panel:** Shows the message "No problems have been detected in the workspace."

Front end index.html



MySQL Data

Created on postman



API before styling

Name:

Special Ability:

Power:

Health:

Attack:

Defence:

Agility:

Intelligence:

RESET

SUBMIT

Supernova

undefined

15000

28000

11000

7800

8000

23000

DELETE

Elements

Console

Sources

Network

Performance

Memory

Application

Security

Lighthouse

top

Filter

characterCreator: ▶ []

index, 1s:17

THIS: ▶ <form>.</form>

index, 1s:17

DATA: ▶ {name: 'Supernova', specialAbility: 'Sonic boom', power: '15000', health: '28000', attack: '11000', ...}

index, 1s:20

RES: ▶ {data: {...}, status: 201, statusText: '', headers: {...}, config: {...}, ...}

index, 1s:24

characterCreator: ▶ [{}]

index, 1s:37

H2 Database with entity

Max rows: 1000

Auto complete

Off

Auto select

On

?

Run

Run Selected

Auto complete

Clear

SQL statement:

SELECT * FROM CHARACTER_CREATOR

SELECT * FROM CHARACTER_CREATOR;

ID	AGILITY	ATTACK	DEFENCE	HEALTH	INTELLIGENCE	CHARACTER_CREATOR_NAME	POWER	SPECIAL_ABILITY
1	8000	11000	7800	28000	23000	Supernova	15000	Sonic boom

(1 row, 1 ms)

Edit

Initial ERD

The screenshot displays the MySQL Workbench environment. The top menu bar includes File, Edit, View, Arrange, Model, Database, Tools, Scripting, and Help. The main workspace is titled 'Diagram' and shows an Entity-Relationship (EER) diagram for a 'character_creator' table. The table's attributes are listed as follows:

- id** INT (Primary Key)
- agility INT
- attack INT
- defence INT
- health INT
- intelligence INT
- character_creator_name VARCHAR(255)
- power INT
- special_ability VARCHAR(255)

The left sidebar contains the 'Catalog Tree' showing a database named 'mydb' with sub-items for Tables, Views, Routine Groups, and a specific table 'may-enable-2'. Below the catalog tree is a 'Description Editor' showing 'No Selection'. The right sidebar, 'Modeling Additions', lists three options: 'timestamps' (create_time, update_time), 'user' (username, email, password), and 'category' (category_id, name). The bottom status bar indicates '1 object(s) deleted.' and the bottom right corner shows 'Templates'.

Completion of all sprints on Jira – Back End

Editing Week-9-Hobby-Project-A

CRUD CharacterCreator

Week 9 Hobby Project - Roadmap

Week-9-Hobby-Project-API/Week

← → ↺ ↻ 🏠 🔒

billymillsqa.atlassian.net/jira/software/projects/W9HP/boards/2/roadmap?selectedIssue=W9HP-2

🔗 ☆ ⚙️ 🗖️ 👤 B ⋮

Jira Software

Your work ▾

Projects ▾

Filters ▾

Dashboards ▾

People ▾

Apps ▾

Create

Week 9 Hobby Project

Software project

PLANNING

Roadmap

Backlog

Board

DEVELOPMENT

Code

Project pages

Add shortcut

Project settings

You're in a team-managed project

Learn more

Projects / Week 9 Hobby Project

Roadmap

🔍

B 👤 👤

Status category ▾

	JUN	JUL	AUG	SEP	OCT	NOV	
Sprints		W9HP Sprint 2 ...					
W9HP-1 Front End							
W9HP-2 Back End							
W9HP-3 Documentation							
+ Create Epic							

Today

Weeks

Months

Quarters

↗️

W9HP-2

🔒

👁️ 1

👍

🔗

⋮

✕

Back End

🔗 👤 🔗 ⋮

To Do ▾

Description

Add a description...

Child issues

Order by ▾ ⋮ +

100% Done

W9HP-16

Place all content in r...

10 👤

DONE ▾

W9HP-19

Fully designed test s...

10 👤

DONE ▾

W9HP-21

Use Github to devel...

10 👤

DONE ▾

W9HP-12

CRUD functionality

10 👤

DONE ▾

Details

⬆️

B

Add a comment...

Pro tip: press **M** to comment

Show all

✕

Completion of all sprints on Jira –Documentation

Editing Week-9-Hobby-Project-A x

CRUD CharacterCreator x

Week 9 Hobby Project - Roadma x

Week-9-Hobby-Project-API/Wee x

+

← → ↺ ↻

billymillsqa.atlassian.net/jira/software/projects/W9HP/boards/2/roadmap?selectedIssue=W9HP-3

🔗 ☆ ⚙️ 🗖️ B ⋮

Jira Software

Your work ▾

Projects ▾

Filters ▾

Dashboards ▾

People ▾

Apps ▾

Create

Week 9 Hobby Project

Software project

PLANNING

Roadmap

Backlog

Board

DEVELOPMENT

Code

Project pages

Add shortcut

Project settings

You're in a team-managed project

Learn more

Projects / Week 9 Hobby Project

Roadmap

Q

B

Status category ▾

	JUN	JUL	AUG	SEP	OCT	NOV	
Sprints		W9HP Sprint 2 ...					
W9HP-1 Front End							
W9HP-2 Back End							
W9HP-3 Documentation							
+ Create Epic							

Today

Weeks

Months

Quarters

↗

W9HP-3

🔒

👁️ 1

👍

🔗

⋮

✕

Documentation

🔗

👤

🔗

⋮

To Do ▾

Description

Add a description...

Child issues

Order by ▾ ⋮ +

100% Done

W9HP-26

Create a wireframe fo...

5

👤

DONE ▾

W9HP-14

Create PowerPoint fo...

7

👤

DONE ▾

W9HP-4

Create MoSCoW Priors...

5

👤

DONE ▾

W9HP-6

Create Risk Matrix

5

👤

DONE ▾

W9HP-7

Create a project usin...

10

👤

DONE ▾

W9HP-5

Create Risk Assessment

5

👤

DONE ▾

B

Add a comment...

Pro tip: press **M** to comment

Show all ✕

Final product

Create your very own character with attributes!

The function of this application is to allow you to customise all attributes of a character of your choice. The basis of this is similar to globally recognised board games such as:

- [Dungeons and Dragons](#)
- [Warhammer](#)
- [Lord of The Rings Roleplaying Game](#)

If you are unfamiliar with attributes in a role playing game, then please click this link for more information.

Enter new character details below

Name:

Special Ability:

Power:

Health:

Attack:

Defence:

Agility:

Intelligences:

Reset

Submit

The results for your newly created character will be displayed below:

If you wish to update any of the attributes, then you can change them in the section below.

S

Special Ability: Sonic boom

Power:4

Health: 4

Attack: 4

Defence: 4

Agility: 4

Intelligence: 4

DELETE

UPDATE

Time flies when playing role-playing games! Press the button below to find out what the time and date is.

-->

Time and date is.

The time is...!

