IMAGE GANG

LAP 1 - Group Project
Art History Quiz

João Patacas
Angela Demiri
Francesca Adu
George Hill
Tobi Akinola

PLAN FOR PRESENTATION

- Purpose of Solution
- Planning & Delivery
- Technologies
- Significant Code
- Live Demonstration
- Future Features
- What We've Learned

User Persona

Name: Eren Yeager

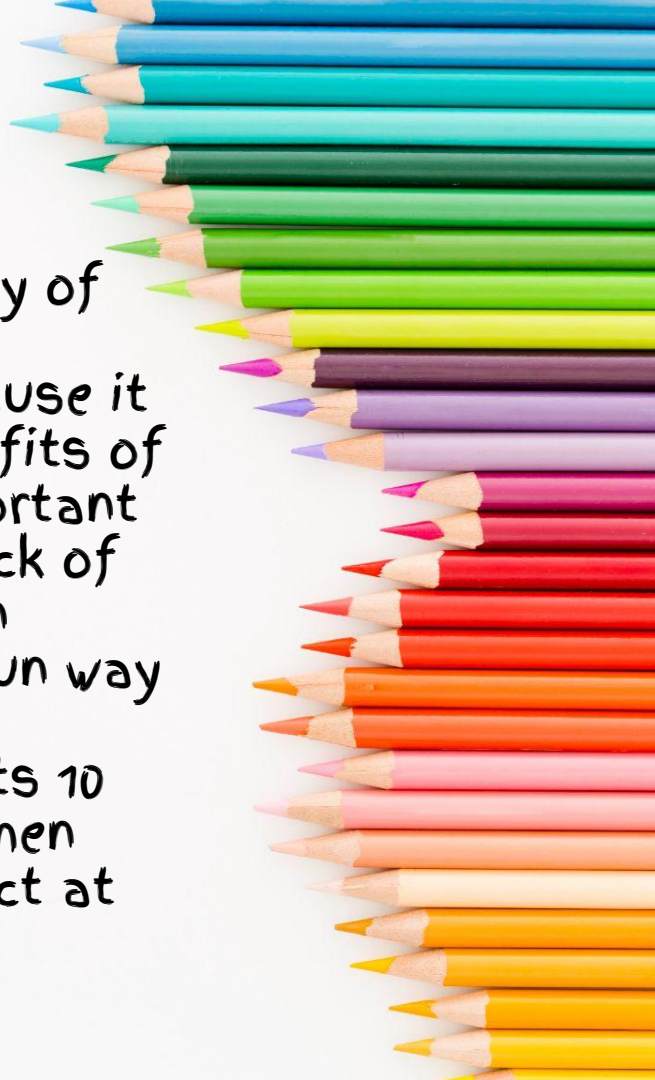
Occupation: Student

Age: 15

Interests/Hobbies: spending time with friends, video games

Purpose of Solution

- Purpose: educate students about the History of Art.
- Reasoning: We based this game on Art because it is commonly forgotten. It has positive benefits of creativity and self-expression which is important for self-worth and confidence. Due to the lack of engagement in non-STEM subjects, our main priority was to make it an interactive and fun way of learning.
- Overview: The game includes asking students 10 questions based on their art knowledge. It then concludes with their score and a random fact at the end.



Planning and Delivery

Following the Agile approach

Phase 1 - Concept

- Our initial task as a team was to conceptualise the requirements for the game, this was done using Miro.

Phase 2 - Inception

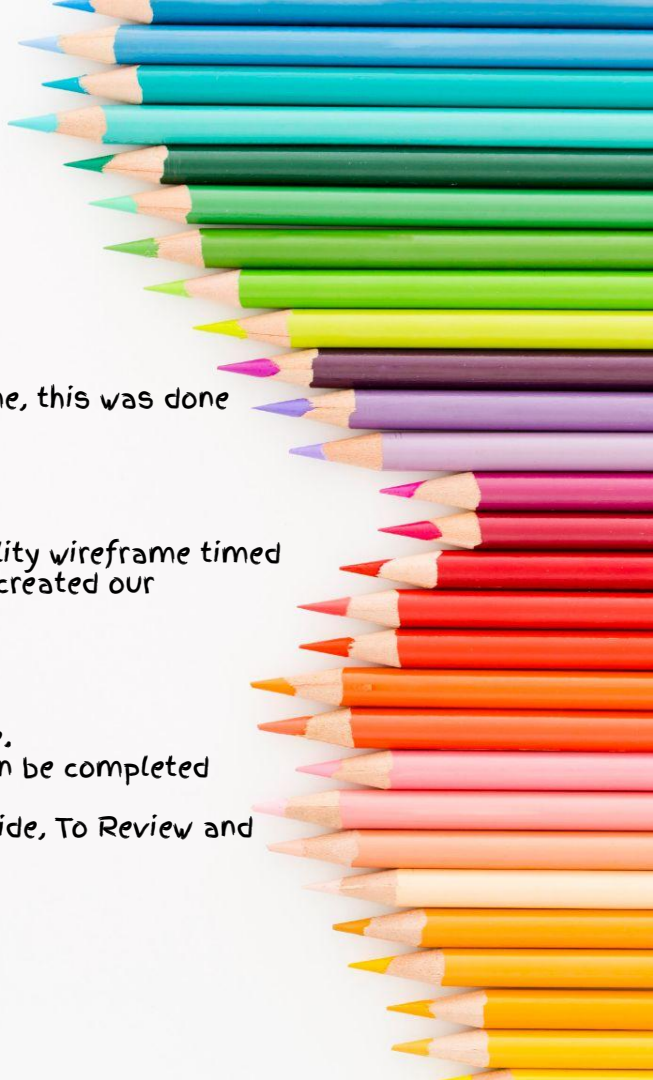
- Once the requirements were established we moved on to creating our low-fidelity wireframe timed sketches (which can be seen on slide 7) to get a rough idea of the UI, we then created our high-fidelity wireframe for a more realistic UI.

Phase 3 - Iteration

- Using Trello we created a SCRUM board to build the functionality of the game.
- Each team member assigned themselves tasks and discussed what tasks can be completed together
- We broke each task down into To-Do and In Progress on both Server & Client side, To Review and the Backlog.

Phase 4 - Development

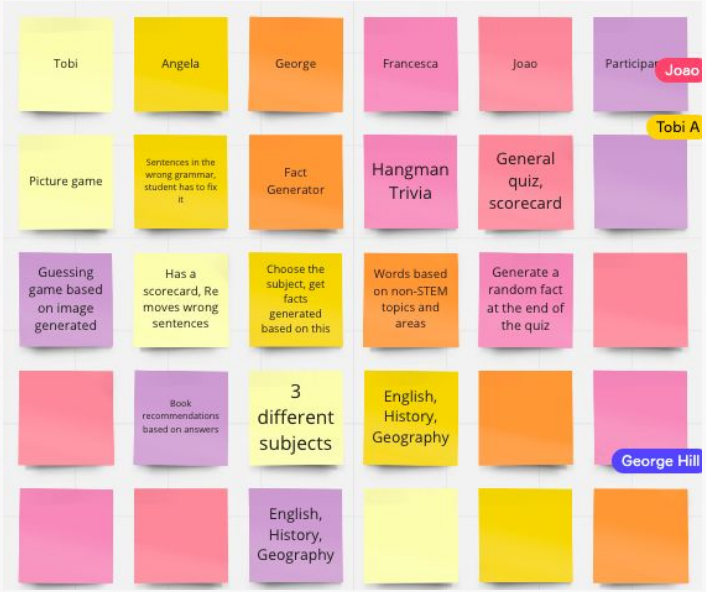
- We pulled our assigned tasks from Trello and started to develop the game

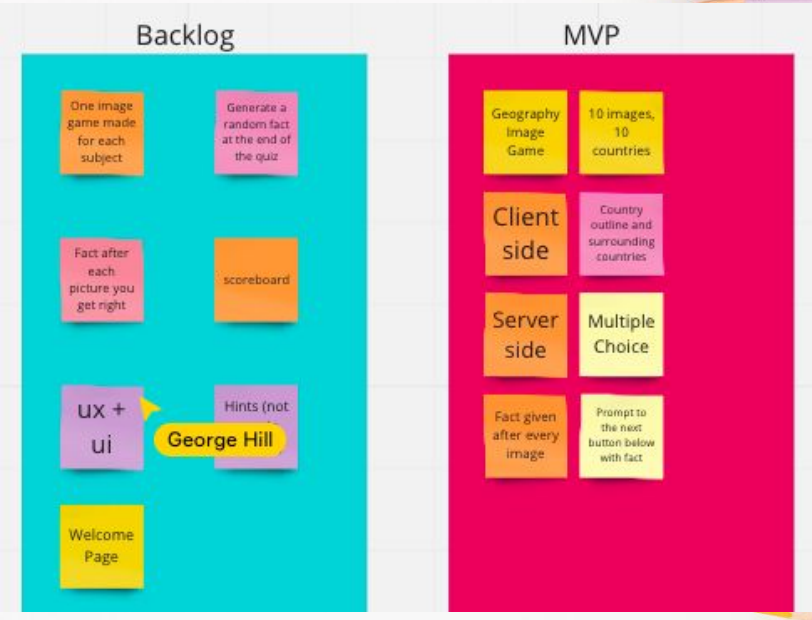


Scope

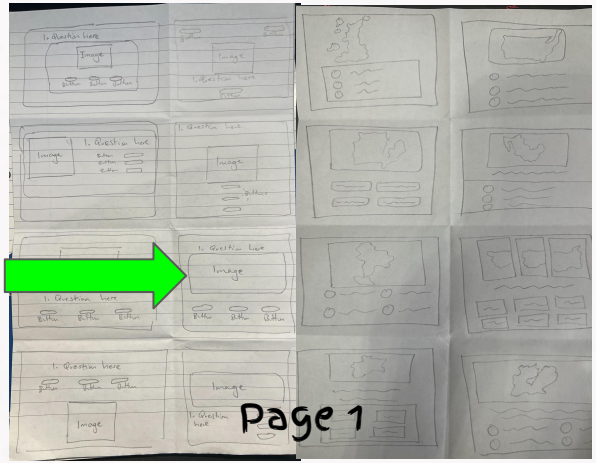
Our starting point was to brainstorm ideas for different games, we then discussed the best features and merged these ideas for our chosen game

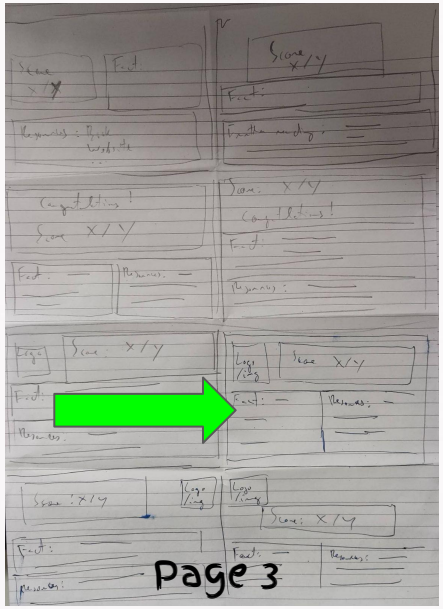
We then decided on the main features we wanted to implement into our MVP.

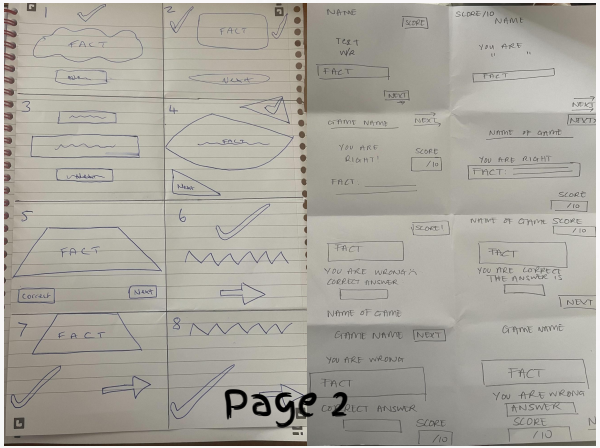




Low Fidelity Wireframe - StoryBoarding

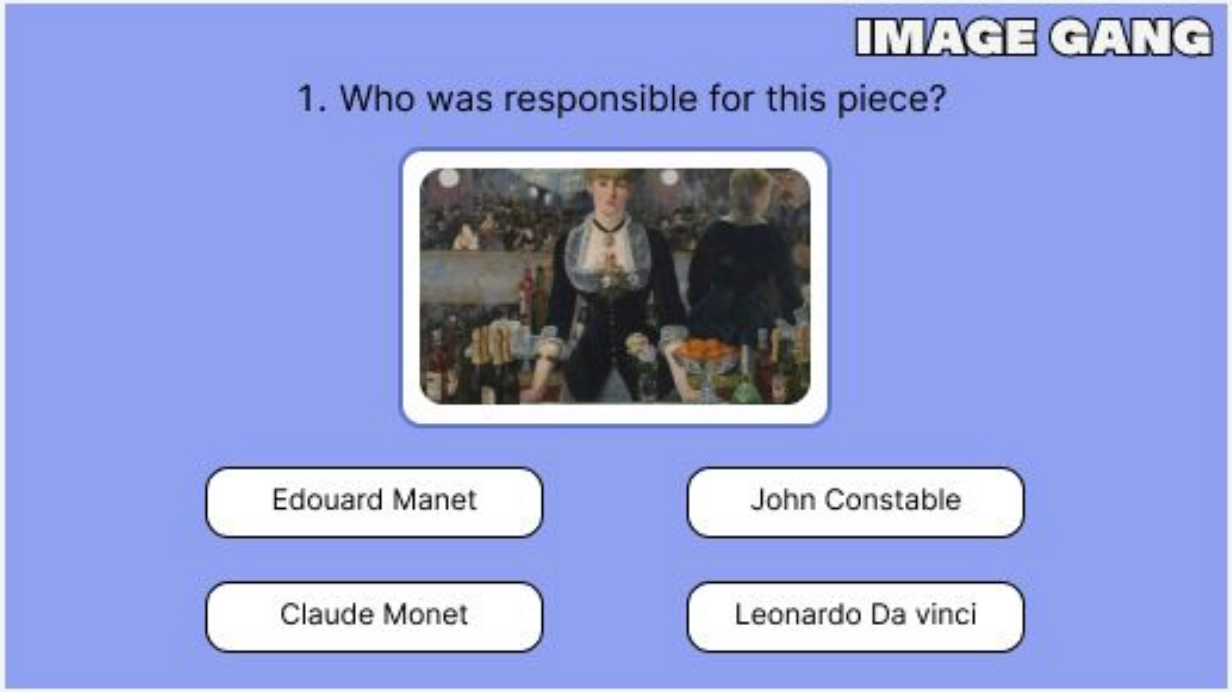






Cut out page 2 since it was not required for our wireframe. Further development of this would include page 2.

High Fidelity Wireframe



High Fidelity Wireframe - Continued

Challenges and Solutions

1) Object to store Questions and Answers data

```
const data = [ {
    id: 1,
    image: "/client/assets/VanGogh.jpeg",
    question: "What painting style was used to create this piece?",
    answers: [
        {id: 1, ans: "Watercolour", correct: false },
        {id: 2, ans: "Pastel", correct: false },
        {id: 3, ans: "Oil", correct: true},
        {id: 4, ans: "Acrylic", correct: false }
    ]
},
{...}
]
```

Challenges and Solutions

Displaying and checking the buttons input

- Joao created functions to update the buttons text with data from the API and check if answer is correct using onclick events

Displaying random facts

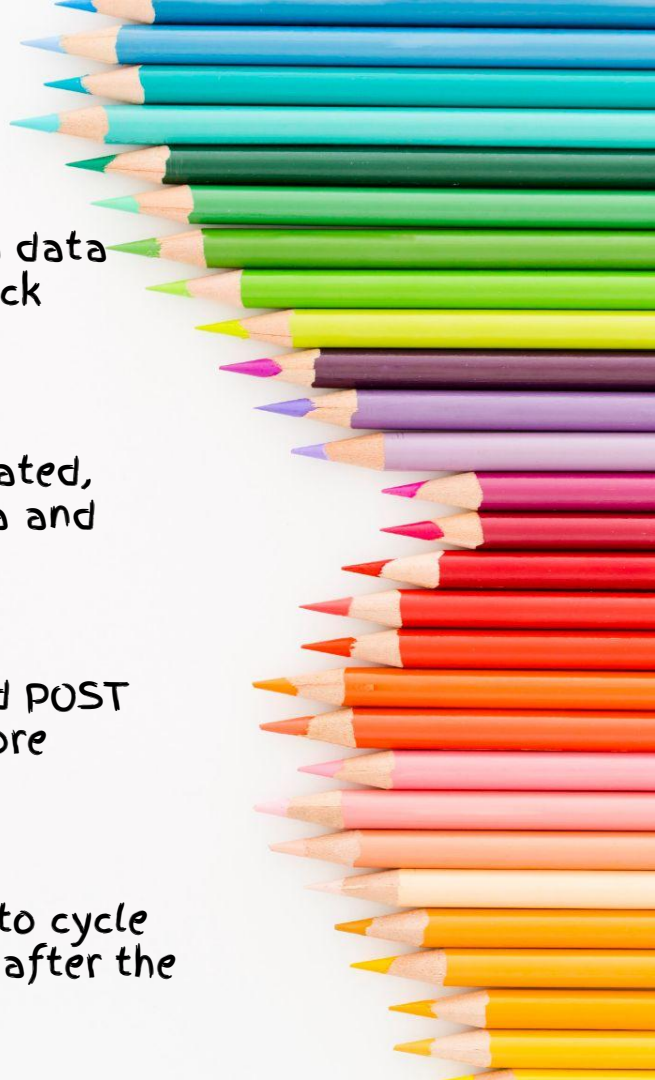
- Angela created a GET method to access the API we created, then used the async await function to retrieve the data and display it on the front-end

Displaying the final score

- Joao created a score object on server side and GET and POST methods to update and show the final score on the score results page

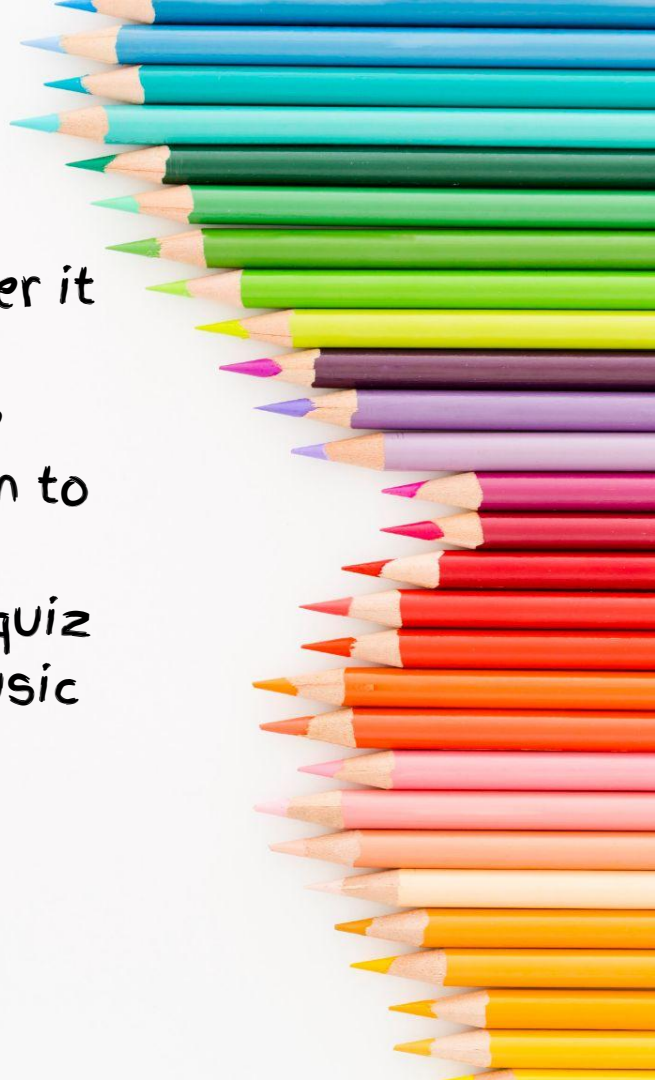
Auto rolling

- Joao combined promises with the setTimeOut method to cycle through the list of questions and go to the score page after the last question



Future Features

- Adding a fact after every response (whether it be right or wrong)
- Display current score on the question page
- Visual message after answering a question to notify whether chosen answer is correct.
- Expanding the webpage to include similar quiz games for other subject areas, namely music and english topics



What we've learned

- Connecting the back end and front end together using async and await functions in JavaScript
- Getting and posting data to an API
- Interesting: planning and working out what to do for the project
- A deeper understanding of GitHub and its communication levels needed.