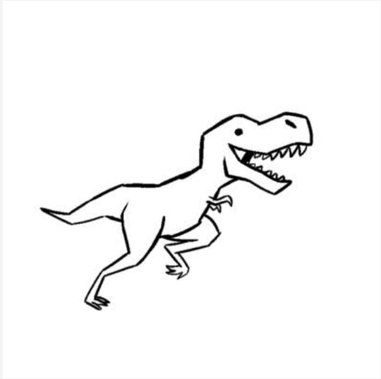


DINO CHASE

PLAY . LEARN . SUCCEED

TABLE OF CONTENTS

Joe F.

Elliot C.

HELLO!

We are Binary Bananas.

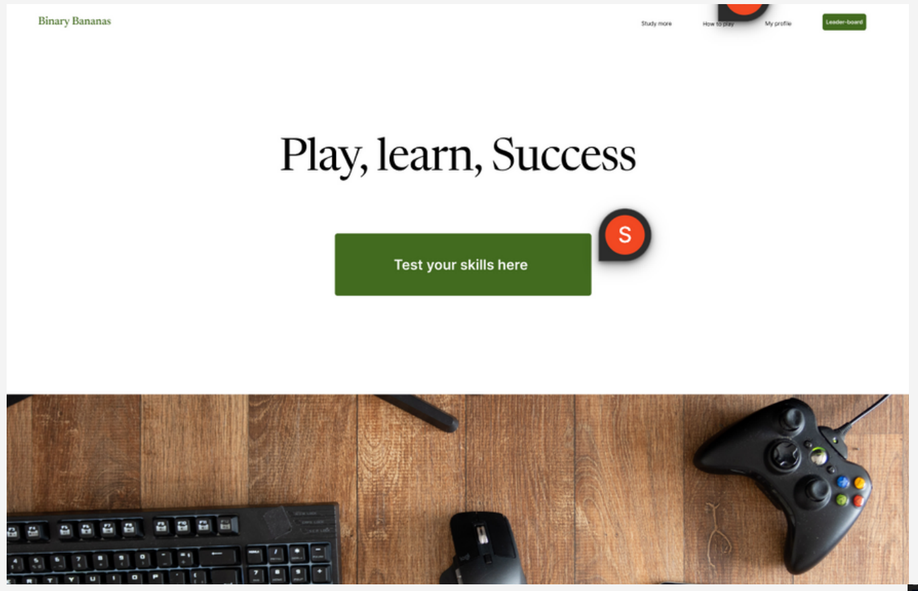
Sumeet C.

Hasan C.

VISION

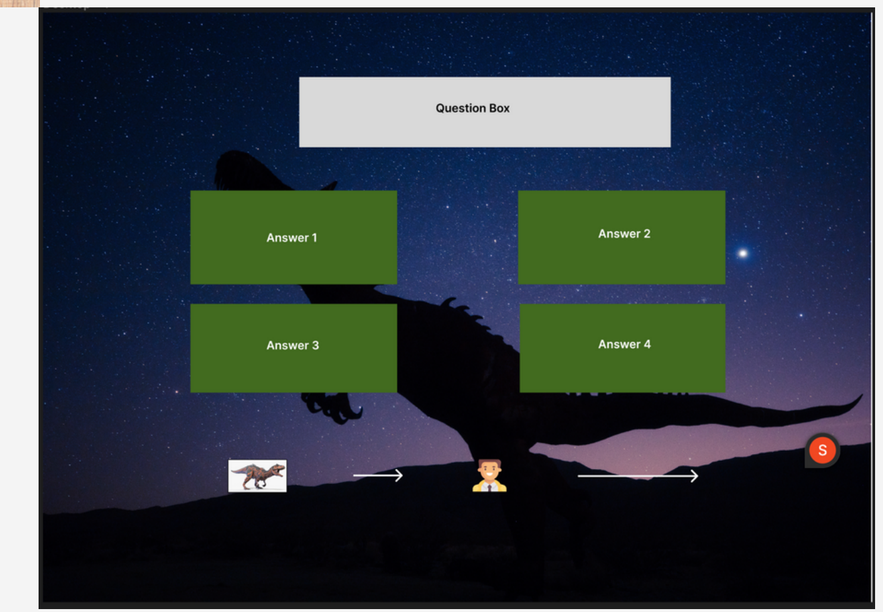
With a greater uptake in STEM subjects for GCSE students, it has left other classic topics only taken as last choice. Dino Chase aims to hit back and give support to equally important subjects. We want to make learning non-STEM that much more enjoyable.

For parents and teachers in particular, this game will act as a revision tool either at the end of class or as an after school activity. Ultimately this will bridge the knowledge of our youth!



1) Understand the problem
2) Crazy 8 design
3) Kanban
4) Wireframe

PLANNING



Step one, surprisingly was not the Kanban board. We wanted to really digest problem and find different motivations for what would be really useful.

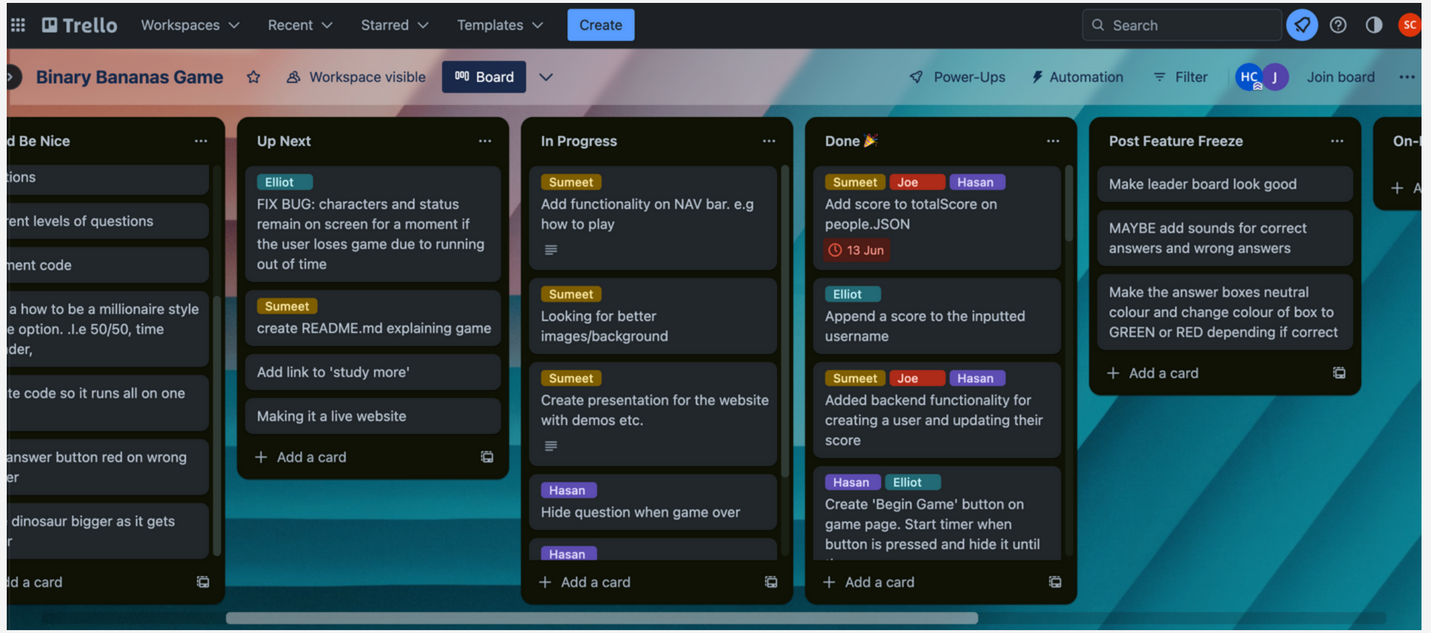
All good designs come from the back of a napkin. As did our Crazy 8 designs!

Wireframe Design solutions such as Figma gave us an immediate understanding of how our game would look.

We opted for a minimalistic design style with a gameified approach to make this tool more inviting.

BINARY BANANAS

DINO CHASE

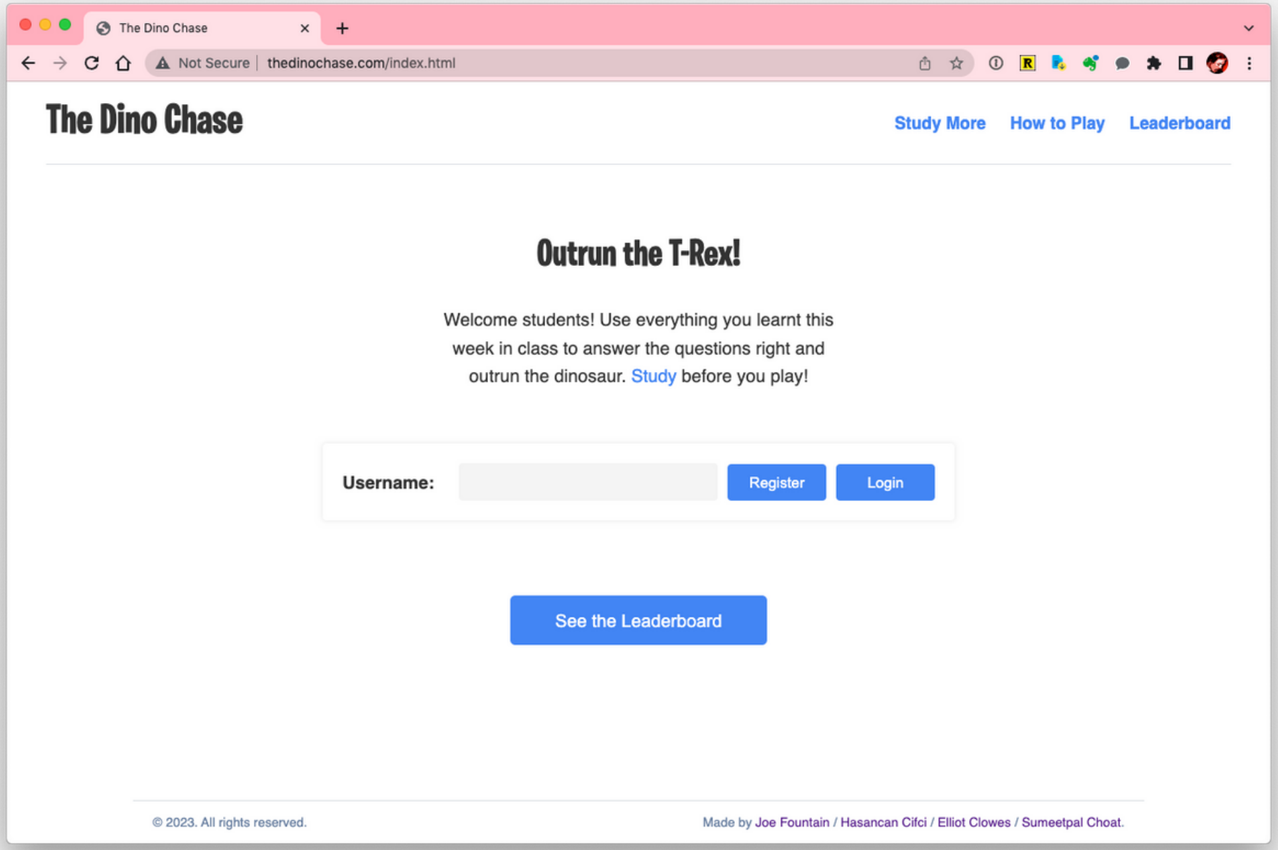


KANBAN

Started off with a Waterfall project management and tried to adjust to an Agile methodology. The Kanban work-flow greatly improved productivity and kept us focused.

In the future we could work on how errors can effect the workflow especially after a merge.

LIVE DEMO TIME!



BINARY BANANAS

DINO CHASE

TECHNOLOGY

Front-End-Dev

Made the game board and the home page integrate into one.

Animation of moving pieces

Back-End-Dev

API creation for users and questions

Dependencies that are required. Express, cors and env

SIGNIFICANT CODE

front-end/game.js

This code changes HTML elements and CSS on the fly as users get answers right and wrong.

It required a lot of work because when you're changing things due to a user clicking a button and because of a countdown running out it's very easy for bugs to develop or for the layout of the page to break and not work correctly.

```
// If time runs out they lose a position
if (userPosition === dinoPosition) {
  endGame();
} else if (countdown < 0) {
  clearInterval(countdownInterval);
  incorrectAnswers++;

  // Move the dinosaur one place to the right. '++'
  document.getElementById("player").style.gridColum

  // Update the status text to indicate an incorrec
  document.getElementById("status").innerText =
    "You ran out of time. New question!";

  // ELLIOT Decrement user score on locally and on
  currentScore--;
  //updateScore(username, currentScore);
```

front-end/home.js

home.js is the JavaScript that powers the homepage. It registers users, logs them in and creates the leaderboard when the 'Show the Leaderboard' button is clicked.

The code needed to be efficient as it generates the leaderboard in real time when the 'See the Leaderboard button is clicked.

```
// Function to retrieve and order data for lea
const userData = async (username) => {
  try {
    const resp = await fetch("https://thedinoc
    if (resp.ok) {
      const data = await resp.json();
      // Set a counter to work out people's po
      let positionCount = 1;
      // For loop to append each data element
      for (let i = 0; i < data.length; i++) {
        if (i != 0 && data[i].totalScore != da
          positionCount++;
        }

        // add information to 3 spans in a li
```

back-end/app.js

Our backend API lives here. We used a simple .JSON file instead of SQL/NoSQL as we lacked experience with databases, didn't need too many security features and we wanted a very simple, fast API.

This is deployed on Render.com – separate to our front-end – as we wanted our back-end to be deployed automatically from GitHub when new changes were made.

```
// Posting usernames to leaderboard.
app.post("/add-user/:user", (req, res) => {
  //check whether a username has been entered
  const userName = req.params.user;
  if (!userName) {
    return res.status(400).json({ error: "Please ent
  }
  //look for that username in the json file, if it a
  const checkUser = leaderboard.find(
    (element) => element.username === userName
  );
  if (checkUser) {
    return res.status(409).json({ error: "Username a
  }

  //if its a new user, push the new user to leaderbo
  const userData = { username: userName, totalScore:
```

BINARY BANANAS

DINO CHASE

CASCADING CHALLENGES

Making a moving, dynamic game with JavaScript, CSS and HTML presented challenges. These technologies don't lend themselves to making a game.

Abandon a tricky feature or double down on the problem?

Sometimes small tasks present big problems.

Allocating tasks in large or small chunks?

Debug process. Testing saves you time.

Time management.

Git *is* a git.

IF WE HAD MORE TIME

SMART PHONE FRIENDLY

INCREASED SECURITY VIA PASSWORD SUPPORT

HTTPS

AUTO DEPLOYING FRONT-END

A PAGE FOR TEACHERS TO ADD QUESTIONS

DIFFICULTY LEVELS AS USER ADVANCES

QUESTIONS PLEASE

ALSO, FEEL FREE TO TRY IT YOURSELF (IT'S FAST →)

THEDINOCHASE.COM