

COMP202-2202628

By Alex Purser

Contents

1. Back end

- A. Setup
- B. MVC architecture
- C. Endpoints
- D. Networking

2. Databases

- A. Projects page
- B. Employment page
- C. Why use JSON?
- D. Handlebars
 - i. Handlebars - issue
- E. Contact me page
 - i. contacts.db
- F. Query management
 - i. Security
- G. Edge cases

3. Front end

- A. Contrast checker
 - i. All checks
 - ii. Read more
 - iii. Header hover

B. Feedback

- i. Overview
- ii. Main page
- iii. Main page – About me
- iv. Projects page
- v. Employment page
- vi. Employment page - animations
- vii. Contact page
- viii. Final feedback

C. Before and after

- i. Main page
- ii. Projects popup
- iii. Employment page

D. Mobile usage

4. Optimisation

A. Lighthouse

- i. Accessibility
- ii. SEO
- iii. Performance

B. Mobile optimisation

5. References

Back-end

Worksheet 1

Setup

```
// ----- Setup ----- //

// handlebars
const hbs = require('hbs')
app.set('view engine', 'hbs')

// register partials folder
hbs.registerPartials(path.join(__dirname, "/public/views/partials"));

// Set template folder
app.set('views', 'public/views')

// setup stylesheets
app.use(express.static(path.join(__dirname, 'public')));
app.use('/stylesheets', express.static('stylesheets'));
```

- The first worksheet of the project was to initialise a backend for the project. This includes endpoints and routes.
- This was the setup for the backend. In the picture, this is where I initialised my components like handlebars and stylesheets.

MVC architecture



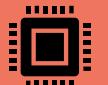
MVC Architecture is the method of organising a website into three main components: Model-View-Controller (MVC).



The Model is the database(s).



The View is the data the users can see.



The Controller is the component that handles the inputs and displays the correct data.

I have used MVC a lot throughout this project. In the back-end, this can be seen with my endpoints.

Endpoints

1

2

3

4

- 1 ○ When the user selects an option in the header, they are taken to the desired webpage. This happens because they trigger an endpoint in app.js, e.g href="/projects".
- 2 ○ App.js notes this trigger and calls the app.get function that contains ('/projects'). In the function, the desired partial (webpage) is selected. Since I wanted each page to have the same layout, I also implemented a layout, that the partial is injected into.
- 3 ○ In the layout, there is a {{body}}. This is the variable where the partial is injected into, outputting the partial's content within the layout.

Projects Employment CV Contact Me

```
<div class="Nav_buttons">
  <a href="/projects">Projects</a>
  <a href="/employment">Employment</a>
  <a href="/cvDownload" download="Alex_Purser_CV.pdf">CV</a>
  <a href="/contact">Contact Me</a>
</div>
// injecting thank you page into layout
app.get('/thanks', (req, res) => {
  res.render('partials/thanks', { layout: 'layouts/layout' });
})
```

```
// injecting projects into layout
app.get('/projects', (req, res) => {
  res.render('partials/projects', {
    layout: 'layouts/layout',
    projectData: projectData
  });
})
```

```
<body>
  {{body}}
</body>
```

Networking

```
const express = require('express')
const app = express()
var path = require('path');
const sqlite3 = require('sqlite3')
```

- For my networking feature, I am using Express.js and Node.js. Express controls all communication between the website and the back-end. I have used Express to allow my website to access my databases, both requesting and responding where required.

Databases

Worksheet 2

Projects page

- The Model is my database called projectData. This is a JSON that allows handlebars to access the data.
- The View is the cards or popup the user sees. This is managed by handlebars.
- The Controller is the read more button which the user clicks to see more data. This is done using a function in the javascript file: readmore.js

- On my website, I wanted to implement a projects page which showcases all my current and past projects that I have worked on.

1

[Read More](#)

2

```
<a class="readMore" onclick="ReadMore({{index}})">Read More</a>
```

3

```
// Make the popup visible and get which button was clicked
function ReadMore(index) {
```

{

4

```
"projects": [
{
  "title": "Cormoran",
  "startDate": "2022",
  "endDate": "2023",
  "information": "Cormoran is the title of one",
  "link": "https://fugii.itch.io/cormoran",
  "picture": "../../pics/Cormoran.png",
  "popup_pics": [
    "../../pics/Cormoran_popup1.png",
    "../../pics/Cormoran_popup2.png",
    "../../pics/Cormoran_popup3.png",
    "../../pics/Cormoran_popup4.png"
  ],
},
```

5

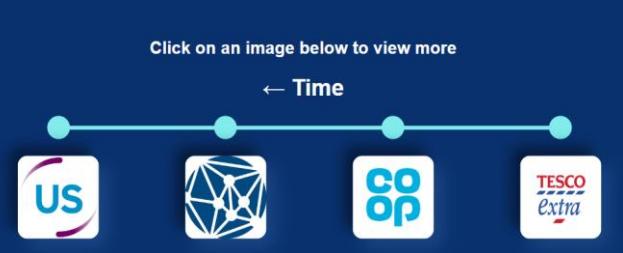


Employment page

- The Model is my database called jobData.json.
- The View is the employment info tab, which displays the data.
- The Controller is the interactive timeline, that the user selects a job on.

On my website, I wanted to implement an employment history page to show users my experience within the work environment. This information shows users my levels of understanding and commitment to the working world and allows me to show what I have gained from them. I wanted this to be like the projects page but have a different layout. The layout I went for was a timeline with interactive events that, when clicked, opened information in a box to the right of the timeline.

1



2

```
<a onclick="select_event({{pic_index}})">
```

3

```
// show data for selected job  
function select_event(index){
```

4

```
{  
    "title": "IT Service Work Placement",  
    "company": "MASS Consultants Ltd.",  
    "startDate": "01/2022",  
    "endDate": "07/2022",  
    "information": "For almost half on my year 13 in education,  
    "logo": ".../pics/logos/mass_logo.png"
```

5

IT Service Work Placement

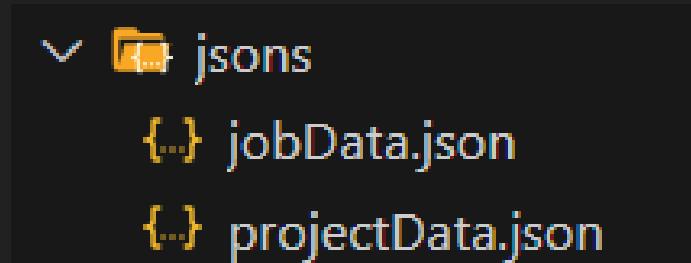
MASS Consultants Ltd.

01/2022 - 07/2022

Info:

For almost half on my year 13 in education, I was employed by MASS Consultants Ltd. This was one of my favourite jobs as it was very tech heavy and gave me insight into a day in the life of IT workers. In this job, I was tasked with taking on client requests (tickets) and working in a team or individually to solve them. Some of the jobs I was tasked with was building computer systems / setups, repairing / upgrading computer systems, and learning how to use the IT management system.

```
{  
  "projects": [  
    {  
      "title": "Cormoran",  
      "startDate": "2022",  
      "endDate": "2023",  
      "information": "Cormoran is the title of one  
      "link": "https://fugii.itch.io/cormoran",  
      "picture": "../../pics/Cormoran.png",  
      "popup_pics": [  
        "../../pics/Cormoran_popup1.png",  
        "../../pics/Cormoran_popup2.png",  
        "../../pics/Cormoran_popup3.png",  
        "../../pics/Cormoran_popup4.png"  
      ]  
    },  
  ],
```



Why use JSON?

- I chose to use JSON files for this because JSONs are a very user-friendly and easy-to-implement database. Typically, JSONs do not support queries since they are mainly used for representing data. But this can be changed using handlebars.
- JSONs are perfect for storing my project and job data. Being able to add data to an array within the JSON allows my website to be flexible and easily changeable, something that drastically improves maintainability.

Handlebars

In this project I used handlebars to maximise efficiency and functionality. In the past I usually stuck with html, but after being shown that handlebars made website data management reactive and flexible, I wanted to give it a go. Handlebars was quite hard to learn at the start, but after a bit of experimenting, became quite understandable.

Handlebars is excellent at managing JSON data. Handlebars automatically parses JSON data, allowing queries to be made. This saved me time from implementing my own JSON parser. Using handlebars' helpers, loops such as #each and #if can be used to retrieve and output data from JSON arrays. In the partial 'projects.hbs' I used an #each helper to display each object in the JSON array titled: "projects". The helper runs through each object in the projects array within the JSON, and extracts the required info, injecting them into the variables such as {{title}} and {{date}}.

#each helper

```
{#each projectData.projects as |project index|}
<div class="card_container">
  <div class="card_content" id="card_content">
    <figure>
      {{project.title}}</h2>
    <h2 class="card_date">{{endDate}}</h2>
    <p class="card_info">{{information}}</p>
    <a class="readMore" onclick="ReadMore({{index}})">Read More</a>
  </div>
</div>
{{/each}}
```

JS function

```
// Make the popup visible and get which button was clicked
function ReadMore(index) {

  // finding the correct popup
  var popup_box = document.getElementById('popup_box' + index);
  var page_content = document.getElementById('page_content');

  page_content.classList.add('blur');
  popup_box.classList.add('popup_box_active');

  // setup the starting slide
  showSlide(index, 0);
}
```

Handlebars - Issue

One problem I had when implementing this was trying to get the correct data across. When the user presses the ReadMore button, the program was unable to output the correct information for this. This caused a roadblock for me for a while.

To fix this, I added the {{index}} value to the id for each project. So, in the each loop, each project is saved in a div with the id: "x + {{index}}" making them all unique. Then, when the user clicks the ReadMore button, the {{index}} value is sent to readmore.js. In here, the index value is taken, and the program gets the current project using getElementById(x + index). This allows the script to find which id aligns with which job, and then make it active.

This problem ended up occurring multiple times throughout the project, but this solution proved very effective at resolving this every time.

Readmore

```
<a class="readMore" onclick="ReadMore('{{index}})">Read More</a>
```

JS function

```
// Make the popup visible and get which button was clicked
function ReadMore(index) {
    // finding the correct popup
    var popup_box = document.getElementById('popup_box' + index);
```

Contact me page

Finally, I wanted an easy way for any user to be able to contact me. I decided to implement a contact me page which users can fill in and submit to me. These messages are stored in a database called "contacts.db". To start with, app.js connects to the database. This is vital as otherwise the program won't know where to send the data.

Next, if needed, a table is created called "messages", which has 4 rows titled: name, email, subject, message. When the user fills in the required inputs and presses submit, the program takes their input, sorts them into each row accordingly, and stores it in the database for later access.

```
// Connect to the Database
const db = new sqlite3.Database('./contacts.db', sqlite3.OPEN_READWRITE, (err) => {
  if (err) return console.log(err.message);
});

// Create table if table doesn't exist
let sql_queryall = `SELECT * FROM messages`;
db.all(sql_queryall, [], (err, rows) => {
  if (err) {
    sql_create_table = `CREATE TABLE messages(id INTEGER PRIMARY KEY, name, email, subject, message)`;
    db.run(sql_create_table);
  }
})
```

contacts.db

'contacts.db' is the name given to my 'contact me' database. When a form is filled out and submitted, it triggers a function in 'app.js' called 'app.post'.

In this function, it gathers each input from the form, and inserts them into the messages table. It starts by breaking up the input into each category: name, email, subject, message. If there are no errors, it adds in all the submitted data, and renders the thanks partial.

This partial says thanks to the user and offers a return home button.

Submit

```
<form action="/" method="POST">

// input from contact me form
app.post('/', (req, res) => {

    // add the data to the db
    let sql_insert_user = `INSERT INTO messages(name, email, subject, message) VALUES (?,?,?,?,?)`;
    db.run(
        sql_insert_user,
        [req.body.name, req.body.email, req.body.subject, req.body.message], // insert into db rows
        (err) => {
            if (err) return res.status('500').send(err.message)

            // render thank you page if there isn't an error
            db.all(sql_queryall, [], (err, rows) => {
                if (err) return console.log(err.message);
                console.log(rows)
                res.render('partials/thanks', { layout: 'layouts/layout' })
            })
        }
    )
})
```

Alex Purser



Projects Employment CV Contact Me

Thank you for contacting me.

Your message has been sent and you can expect a response within a few days.

[Return to home page.](#)

Query management

- As of right now, I have very little query management in place. I have 2 query management options:
 - taking and storing the data
 - Checking if emails already exist
- In the future, I plan to implement an auto-emailer system that sends the messages to my main email account. However, I haven't implemented this yet as I cannot protect against hacker attacks. Attacks such as DDOS and SQL injections are still vulnerabilities of the site, something which I will need more time and research to be able to achieve.
- Currently, the messages get stored in the database, but the data is never taken or removed.

```
// check if email is already in database
let sql_check_email = `SELECT * FROM messages WHERE email = ?`;
db.get(sql_check_email, [req.body.email], (err, row) => {
  if (err) return res.status('500').send(err.message);
  if (row) return res.status('500').send('Email address already exists');
```

```
{
  id: 20,
  name: 'test',
  email: 'test@test.test',
  subject: 'test',
  message: 'test'
},
{
  id: 21,
  name: 'test',
  email: 'test@test.test',
  subject: 'test',
  message: 'test'
},
{
  id: 22,
  name: 'test',
  email: 'test@test.test',
  subject: 'test',
  message: 'test'
}
```

Security

- One other optimisation I wanted to include was my attempt at preventing DDOS attacks in the contact form. I found out that once the user submitted a message, if they refreshed the page, it would send the message again, leading to a very exploitable weakness.
- To fix this, I added an if statement that checks if the newly entered email exists with the email row. If it does, it returns the message “Email address already exists”.
- Unfortunately, this does also mean that users can only send one message per email, but in theory I should have responded to their original message before the need for this ever arises.

Edge cases

- I have also considered edge cases in the back-end of the project: using error catches and correct outputs.
- For example, in my email checker, I have added an if (err) check. This catches any mistakes made and exits the function, returning the error message.

```
if (err) return res.status('500').send(err.message);
```

Front-end

Worksheet 3

- The next step in completing the website was to check the colour contrast. This is a crucial feature that determines if the colour contrast between 2 elements meets the WCAG requirements. Without implementing this check, some users might have a harder time using this website, and so might be deterred from using it. To do this, I used the website: WebAIM. [1]

Contrast checker

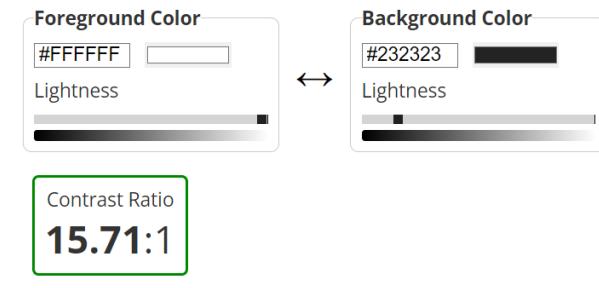
All checks

- Using the WebAIM colour contrast checker, I tested all my colours across the website. These are a few of the passes I got.

Contact Input field

Contrast Checker

[Home](#) > [Resources](#) > Contrast Checker



Normal Text

WCAG AA: **Pass** WCAG AAA: **Pass** The five boxing wizards jump quickly.

Large Text

WCAG AA: **Pass** WCAG AAA: **Pass** The five boxing wizards jump quickly.

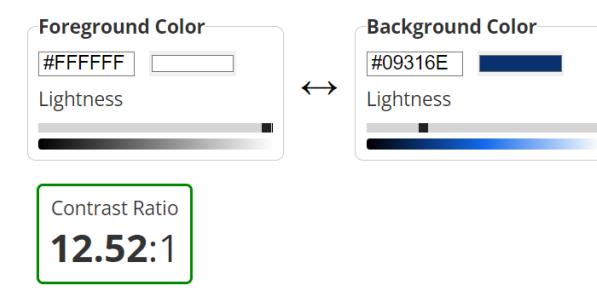
Graphical Objects and User Interface Components

WCAG AA: **Pass** Text Input ✓

Background with text

Contrast Checker

[Home](#) > [Resources](#) > Contrast Checker



Normal Text

WCAG AA: **Pass** WCAG AAA: **Pass** The five boxing wizards jump quickly.

Large Text

WCAG AA: **Pass** WCAG AAA: **Pass** The five boxing wizards jump quickly.

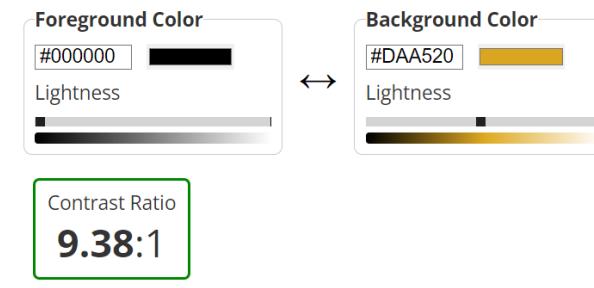
Graphical Objects and User Interface Components

WCAG AA: **Pass** Text Input ✓

Submit button

Contrast Checker

[Home](#) > [Resources](#) > Contrast Checker



Normal Text

WCAG AA: **Pass** WCAG AAA: **Pass** The five boxing wizards jump quickly.

Large Text

WCAG AA: **Pass** WCAG AAA: **Pass** The five boxing wizards jump quickly.

Graphical Objects and User Interface Components

WCAG AA: **Pass** Text Input ✓

Read More before

Contrast Ratio
5.74:1

[permalink](#)

Normal Text

WCAG AA: **Pass**
WCAG AAA: **Fail**

The five boxing wizards jump quickly.

Large Text

WCAG AA: **Pass**
WCAG AAA: **Pass**

The five boxing wizards jump quickly.

Graphical Objects and User Interface Components

WCAG AA: **Pass**

Text Input ✓



Cormoran 2023

Cormoran is the title of one of the video games I helped produce in my first year of university. In GAM190, the name given to this module, we were...

[Read More](#)

Read More after

Contrast Ratio
7.51:1

[permalink](#)

Normal Text

WCAG AA: **Pass**
WCAG AAA: **Pass**

The five boxing wizards jump quickly.

Large Text

WCAG AA: **Pass**
WCAG AAA: **Pass**

The five boxing wizards jump quickly.

Graphical Objects and User Interface Components

WCAG AA: **Pass**

Text Input ✓



Cormoran 2023

Cormoran is the title of one of the video games I helped produce in my first year of university. In GAM190, the name given to this module, we were...

[Read More](#)

Read more

- Unfortunately, the Read More button located on my project cards failed the test. As you can see below, it failed by approximately 1.3 points, failing the WCAG AAA test for normal text.

This is the old hover colour. As you can see it is difficult to distinguish between the foreground and the background.

Here is the result. As you can see it is much easier to distinguish and looks more presentable than the previous colour. As a result, users should have an easier time using the header.

The image shows two side-by-side screenshots of a website's navigation bar. Both screenshots feature a dark header with white text for 'Projects', 'Employment', 'CV', and 'Contact Me'. The first screenshot on the left shows a 'Foreground Color' of #000000 and a 'Background Color' of #808080. The 'Contrast Ratio' is displayed as 5.31:1, which is highlighted with a green border. Below this, under 'Normal Text', there are WCAG AA and AAA test results: WCAG AA is a green 'Pass' button, and WCAG AAA is a red 'Fail' button. A gray box contains the text 'The five boxing wizards jump quickly.' The second screenshot on the right shows the same setup but with a different color scheme. The 'Foreground Color' is now #000000 and the 'Background Color' is #B3B3B3. The 'Contrast Ratio' is now 10.01:1, also highlighted with a green border. Below this, under 'Normal Text', both WCAG AA and AAA results are green 'Pass' buttons. A gray box contains the same text 'The five boxing wizards jump quickly.' In both cases, the text color is white or black, and the background color is a shade of gray.

Header hover

The other big colour contrast fail was in the header: when a user hovers over a navigation button or the logo.

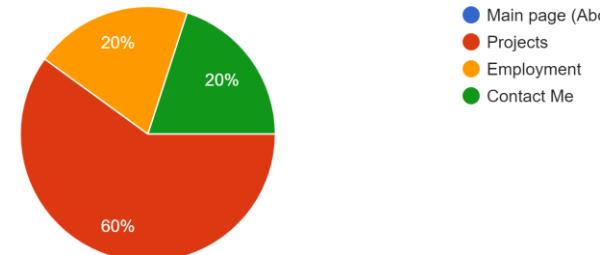
Feedback

Overview

- The next segment of the project was to implement peer feedback and reviews. This is a vital aspect as it allowed me to gather critical feedback and improvements that I should implement. In order to get feedback for my website, I made a form which had my peers fill it out. Some questions asked for a rank from 1-10 about aspects of the site such as fonts and colours, whereas other questions asked for comments / improvements. The results from these questions were very helpful in progressing my website to the next step.
- The figures on the left were my two favourite questions as they showed very simply which page needed the most focus and which the least.

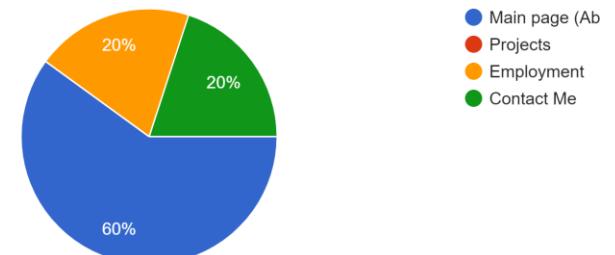
Which section needs the least improvement?

5 responses



Which section needs the most improvement?

5 responses



Main page

"More photos"

To fix this I have added a main picture of myself above my name. I might add more pictures later, but I think for now this is a decent fix to the feedback.

"No animations that I could find"

While this might be a stretch, I believed that the most interesting way to resolve this feedback was to add an animated background. I wanted my main page to stand out, grabbing the users attention. I believe that with my new animated background, that is possible. I got the animated background from Chris Smith on codepen.io [2]. I have tinkered with the design changing the colours and the animation speed.



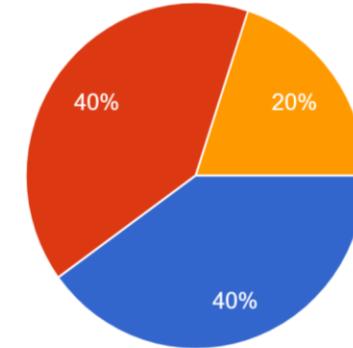
Alex Purser

Student at Falmouth University studying Computer Science BSc.

Main page – About me

Is there enough content on this page?

5 responses



- Yes
- No
- Maybe

"Add margin to the About Me paragraph."

I have implemented this feature by adding a small margin on the sides. In-turn, this looks a lot better and more presentable. I have also added more content and a mention to my projects page, giving the user some incentive to navigate to the page. This has also been colour contrast checked. In future, I will find and implement more pictures, but I believe that having a main picture and a moving background is enough to satisfy this feedback.

About Me

Hi, my name is Alex Purser. I was born and raised in Cambridgeshire but moved to Falmouth for studying. I am currently studying for a Bachelor of Science in Computer Science in order to one day achieve a dream job within the technology industry. Having grown up with technology, it has always intrigued me and eventually drove me to studying it. Throughout my childhood I would always play games, code, use the internet etc. Having access to great resources and systems of my own further enhanced this interest, leading to today. Studying at university has been great. In my first year I have studied lots about hardware, software, data analysis and game development, which have all been very fascinating and enjoyable. More information about these can be seen in [Projects](#).

While I have always been tech heavy, I do enjoy real-life stuff, such as sports and reading. I have played many sports, including football, hockey and softball, and now I have found myself focusing on two mainly: badminton and golf. At university, I have joined the badminton society and the Christian Union. These societies are great at keeping me busy and social, and are great for stress relief and enjoyment.

Projects page

"Slightly increase the font size on description of the project pop ups."

This was another useful piece of feedback stating that it was hard to read the project information in the popup. To fix this I made the paragraph font size 4px bigger, from 16px to 20px.

"Specify that group projects are group projects

New text: "Cormoran is the title of one of the video games I helped produce in my first year of university. In GAM190, the name given to this module, we as a team were tasked with creating a video game with the constraint: Local history. Cormoran is a game based on the legend of Jack the Giant Slayer, a Cornish legend turned global. The premise for this game is that the player (Jack) needs to find a way to kill Cormoran, seeking revenge for his father. The player navigates around St Micheals Mount, locating objects and eventually digging a hole for Cormoran to fall into, all while avoiding him. For this project, I was the developer behind the AI for the giant. This included navMesh pathfinding, raycasting, waypoint management, and player tracking. This was a very enjoyable game to work on and I would love to continue with it at somepoint later on."

Before

Cormoran

2022 - 2023

Info:

Cormoran is the title of one of the video games I helped produce in my first year of university. In GAM190, the name given to this module, we were tasked with creating a video game with the constraint: Local history. Cormoran is a game based on the legend of Jack the Giant Slayer, a Cornish legend turned global. The premise for this game is that the player (Jack) needs to find a way to kill Cormoran, seeking revenge for his father. The player navigates around St Micheals Mount, locating objects and eventually digging a hole for Cormoran to fall into, all while avoiding him.

Link:

<https://fugii.itch.io/cormoran>

Close

After

Cormoran

2022 - 2023

Info:

Cormoran is the title of one of the video games I helped produce in my first year of university. In GAM190, the name given to this module, we as a team were tasked with creating a video game with the constraint: Local history. Cormoran is a game based on the legend of Jack the Giant Slayer, a Cornish legend turned global. The premise for this game is that the player (Jack) needs to find a way to kill Cormoran, seeking revenge for his father. The player navigates around St Micheals Mount, locating objects and eventually digging a hole for Cormoran to fall into, all while avoiding him. For this project, I was the developer behind the AI for the giant. This included navMesh pathfinding, raycasting, waypoint management, and player tracking. This was a very enjoyable game to work on and I would love to continue with it at somepoint later on.

Link: <https://fugii.itch.io/cormoran>

Close



IT Service Work Placement

MASS Consultants Ltd.

01/2022 - 07/2022

Info:

For almost half on my year 13 in education, I was employed by MASS Consultants Ltd. This was one of my favourite jobs as it was very tech heavy and gave me insight into a day in the life of IT workers. In this job, I was tasked with taking on client requests (tickets) and working in a team or individually to solve them. Some of the jobs I was tasked with was building computer systems / setups, repairing / upgrading computer systems, and learning how to use the IT management system.

Employment page

"Use less white spacing"

"Fill the space by making the icons larger"

Both pieces of feedback showcase the significance of correctly-used white-spacing and icon-size. This was an easy fix to implement and makes the employment page look a lot nicer.

"I didn't know I could click on the Timeline at first but apart from them very beautiful"

This is a very important piece of feedback as it shows that the timeline isn't intuitive enough and should be made more obvious. This could result in users being unable to use the employment page properly and so miss its purpose and key information.

To fix this I added a textbox that tells the user what to do. Underneath this I also added a "Time" label, to help users understand it's a timeline. These fixes should increase user satisfaction and make the timeline easier to understand.

Furthermore, this fix also helps with the "Use less white spacing" piece of feedback, making the page look better.

Welcome to my employment history page. For my portfolio, I believe that showcasing my previous work experience provides good insight into my levels of experience, especially within the technology industry. Each of these jobs have taught me valuable skills and lessons.

IT Service Work Placement

MASS Consultants Ltd.

01/2022 - 07/2022

Info:

Click on an image below to view more

← Time

For almost half on my year 13 in education, I was employed by MASS Consultants Ltd. This was one of my favourite jobs as it was very tech heavy and gave me insight into a day in the life of IT workers. In this job, I was tasked with taking on client requests (tickets) and working in a team or individually to solve them. Some of the jobs I was tasked with was building computer systems / setups, repairing / upgrading computer systems, and learning how to use the IT management system.

Employment page - animations

"Animate the side card as you click on a different job."



I have implemented this feedback and fully agree that this animation was needed. The animation wasn't a hard feature to implement and adds a lot to the page. As a result, the page looks a lot more creative and interesting.

For this piece of feedback, I also wanted to add a few more just to boost interactivity. To do this I added an animation on the arrow for the timeline guide. Now, whenever the user hovers over "Time" or the arrow, the arrow moves to the left, showcasing the direction of the timeline.

Both animations I made.

Contact me

The contact me page didn't have much feedback to implement.

```
#### "add data validity checks to input fields"
```

This feedback was very important as it highlighted a major flaw within my contact form. Currently, users can enter any input into the email section. This can cause many issues, such as miss-inputs, spam attempts etc. For example, if a user sent me a form and mistyped their email, I wouldn't be able to contact them afterwards. While this fix doesn't mitigate this issue, it certainly dampens it.

Before

```
<div class="input_field">
  <label for="email">Email address *</label>
  <input type="text" name="email" placeholder="Your email address..." required>
</div>
```

After

```
<div class="input_field">
  <label for="email">Email address *</label>
  <input type="email" name="email" placeholder="Your email address..." required>
</div>
```

After

Email address *

test_test_test_test_test_test_test_test_

Subject *



Please include an '@' in the email address. 'test_test_test_test_test_test_test_test_' is missing an '@'.

Final feedback:

“motion reduced accessibility”
“Maybe some more consideration for accessibility/WCAG”

This was a very good idea for some final feedback. Including a reduced motion option is a great addition which will greatly improve the experience for some users.

This drastically improves accessibility and helps to improve overall user inclusivity.

```
// Disabling all motion
@media (prefers-reduced-motion: reduce) {
  .bg {
    display: none;
  }

  .employment_container {
  }

  .event_active {
    animation: none;
  }

  .event_active_pic {
    transform: none;
  }

  .card_container:hover {
    transform: none;
  }
}
```

Before and after

Ws3 feedback

Only showing the pages with changes*

Main page

Alex Purser



Projects Employment CV Contact Me

Alex Purser

Student at Falmouth University studying Computer Science BSc.

About Me

Hi, my name is Alex Purser. I was born and raised in Cambridgeshire where I had ... I am currently studying for a Bachelor of Science in Computer Science in order to one day achieve a dream job within the technology industry. Having grown up with technology, it has always intrigued me and eventually drove me to studying it. Having access to great resources and systems of my own further enhanced this interest, leading to today. While I have always been tech heavy, I do enjoy real-life stuff, such as sports and reading. I have played many sports, including football, hockey and softball, and now I have found myself focusing on two mainly: badminton and golf.

Alex Purser



Projects Employment CV Contact Me



Alex Purser

Student at Falmouth University studying Computer Science BSc.

About Me

Hi, my name is Alex Purser. I was born and raised in Cambridgeshire but moved to Falmouth for studying. I am currently studying for a Bachelor of Science in Computer Science in order to one day achieve a dream job within the technology industry. Having grown up with technology, it has always intrigued me and eventually drove me to studying it. Throughout my childhood I would always play games, code, use the internet etc. Having access to great resources and systems of my own further enhanced this interest, leading to today. Studying at university has been great. In my first year I have studied lots about hardware, software, data analysis and game development, which have all been very fascinating and enjoyable. More information about these can be seen in [Projects](#).

While I have always been tech heavy, I do enjoy real-life stuff, such as sports and reading. I have played many sports, including football, hockey and softball, and now I have found myself focusing on two mainly: badminton and golf. At university, I have joined the badminton society and the Christian Union. These societies are great at keeping me busy and social, and are great for stress relief and enjoyment.

Projects popup

Alex Purser

Projects Employment CV Contact Me

Projects

Cormoran

2022 - 2023

Info:

Cormoran is the title of one of the video games I helped produce in my first year of university. In GAM190, the name given to this module, we were tasked with creating a video game with the constraint: Local history. Cormoran is a game based on the legend of Jack the Giant Slayer, a Cornish legend turned global. The premise for this game is that the player (Jack) needs to find a way to kill Cormoran, seeking revenge for his father. The player navigates around St Micheals Mount, locating objects and eventually digging a hole for Cormoran to fall into, all while avoiding him.

Link:

<https://fugii.itch.io/cormoran>

Close

Alex Purser

Projects Employment CV Contact Me

Cormoran

2022 - 2023

Info:

Cormoran is the title of one of the video games I helped produce in my first year of university. In GAM190, the name given to this module, we as a team were tasked with creating a video game with the constraint: Local history. Cormoran is a game based on the legend of Jack the Giant Slayer, a Cornish legend turned global. The premise for this game is that the player (Jack) needs to find a way to kill Cormoran, seeking revenge for his father. The player navigates around St Micheals Mount, locating objects and eventually digging a hole for Cormoran to fall into, all while avoiding him. For this project, I was the developer behind the AI for the giant. This included navMesh pathfinding, raycasting, waypoint management, and player tracking. This was a very enjoyable game to work on and I would love to continue with it at somepoint later on.

Link: <https://fugii.itch.io/cormoran>

Close

Employment page

Alex Purser

[f](#) [g](#) [i](#) [in](#) [t](#) [o](#)

Projects Employment CV Contact Me

Employment History

Welcome to my employment history page. For my portfolio, I believe that showcasing my previous work experience provides good insight into my levels of experience, especially within the technology industry. Each of these jobs have taught me valuable skills and lessons.

Title:
Company:
Date:
Info:

Click on an image below to view more

← Time

IT Service Work Placement
MASS Consultants Ltd.
01/2022 - 07/2022
Info:

For almost half on my year 13 in education, I was employed by MASS Consultants Ltd. This was one of my favourite jobs as it was very tech heavy and gave me insight into a day in the life of IT workers. In this job, I was tasked with taking on client requests (tickets) and working in a team or individually to solve them. Some of the jobs I was tasked with was building computer systems / setups, repairing / upgrading computer systems, and learning how to use the IT management system.

Alex Purser

[f](#) [g](#) [i](#) [in](#) [t](#) [o](#)

Projects Employment CV Contact Me

Employment History

Welcome to my employment history page. For my portfolio, I believe that showcasing my previous work experience provides good insight into my levels of experience, especially within the technology industry. Each of these jobs have taught me valuable skills and lessons.

Title:
Company:
Date:
Info:

Click on an image below to view more

← Time

IT Service Work Placement
MASS Consultants Ltd.
01/2022 - 07/2022
Info:

For almost half on my year 13 in education, I was employed by MASS Consultants Ltd. This was one of my favourite jobs as it was very tech heavy and gave me insight into a day in the life of IT workers. In this job, I was tasked with taking on client requests (tickets) and working in a team or individually to solve them. Some of the jobs I was tasked with was building computer systems / setups, repairing / upgrading computer systems, and learning how to use the IT management system.

Mobile Usage

I want my website to be as accessible as possible, and this means mobile implementation. Nowadays, a lot of research is done on mobiles, which means my website needs to be able to do this. The main feature I added was a burger menu. This menu takes in segments of the header as the screen width decreases. I also adapted some layouts such as the employment page to fit a vertical view.

The image displays three side-by-side screenshots of a mobile website, illustrating its responsive design across different devices. All three screenshots show the same basic layout with a dark blue header and a white content area.

- Header:** The header features the name "Alex Purser" in white text. On the right side of the header is a white hamburger menu icon (three horizontal lines). Below the header is a circular profile picture of two men.
- Content Area:** The content area has a dark blue background with white text. It includes:
 - A large heading "Alex Purser" at the top.
 - A sub-heading "Student at Falmouth University studying Computer Science BSc."
 - A social media section with icons for Facebook, Instagram, LinkedIn, and GitHub.
- Mobile Navigation (Burger Menu):** In the middle screenshot (iPhone SE), the burger menu is open, revealing a vertical list of links: "Projects", "Employment", "CV", and "Contact Me".
- Tablet Navigation (iPad Air):** In the right screenshot (iPad Air), the burger menu is open, revealing a vertical list of links: "Projects", "Employment", "CV", and "Contact Me". Above the menu, there is a row of social media icons: Facebook, Twitter, Instagram, LinkedIn, GitHub, and a magnifying glass icon.

```

// header width, take out navigation
> @media only screen and (max-width: 870px) { ...
}

// header width, take out socials
> @media only screen and (max-width: 510px) { ...
}

// header width, ensure when out of burger width, contents are disabled
> @media only screen and (min-width: 870px) { ...
}

// display timeline above info box
@media only screen and (max-width: 1000px) {
    .employment_container {
        .event_info_container {
            display: block;
            width: 90%;
            margin: auto;
        }

        .timeline_container {
            display: block;
            width: 90%;
            margin: auto;
        }
    }
}

// shrink job icons
> @media only screen and (max-width: 600px) { ...
}

```

This is what the CSS behind these changes looks like. I utilised the @media only tag, which only applies these changes if the requirement is met i.e screen size.

Employment page mentioned in previous slide

Dimensions: iPad Mini ▾ 768 × 1024 50% ▾ No throttling ▾

Alex Purser

Employment History

Welcome to my employment history page. For my portfolio, I believe that showcasing my previous work experience provides good insight into my levels of experience, especially within the technology industry. Each of these jobs have taught me valuable skills and lessons.

Click on an image below to view more

← Time

US

co op

TESCO extra

Title:

Company:

Date:

Optimisation

Lighthouse

Alex Purser



Projects Employment CV Contact Me

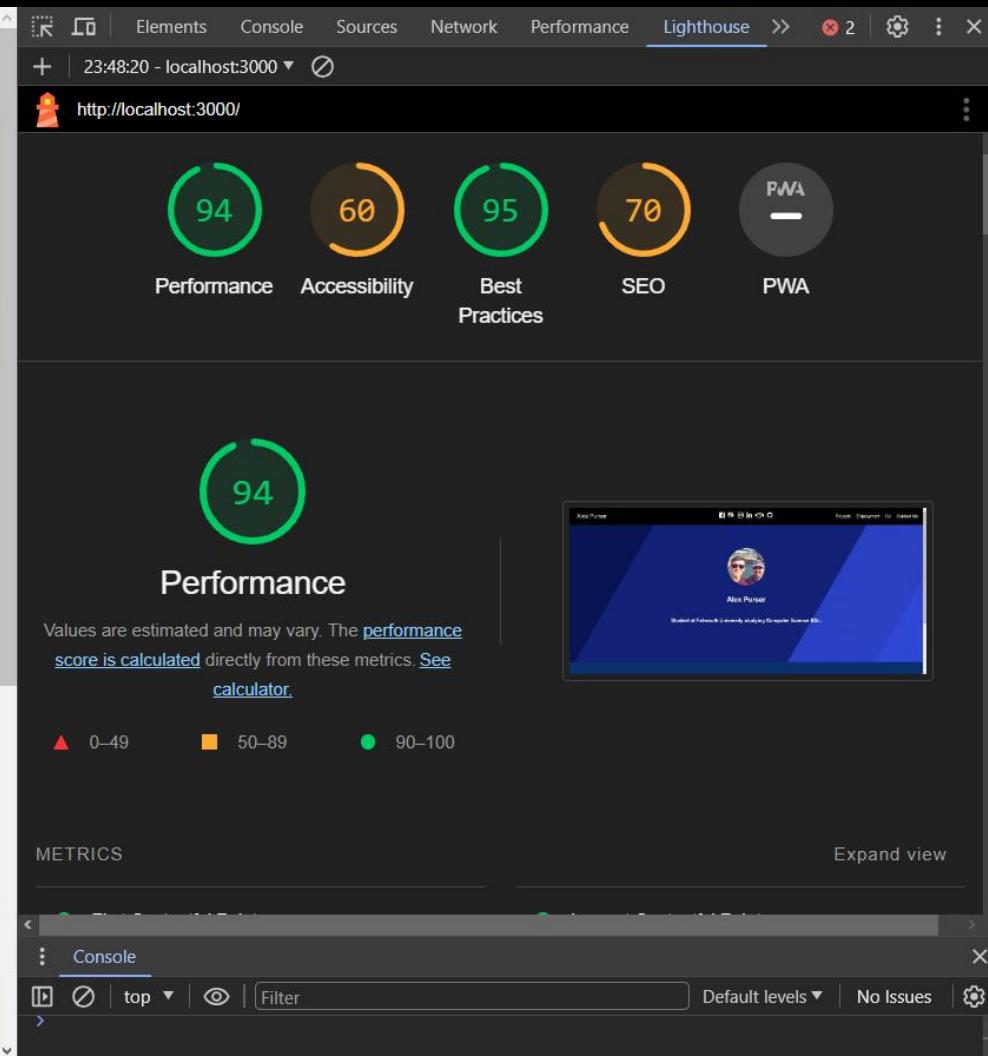


Alex Purser

Student at Falmouth University studying Computer Science BSc.

Lighthouse [3] is a website HCI and performance analyser.

When using Lighthouse, it scans the entire website and outputs any problems or issues, alongside a score. This proved very helpful in optimising my website as it highlighted issues I had completely missed.



Accessibility



Accessibility is a vital part to any website. Unfortunately, at this point my website had a score of 60, which is clearly unacceptable. Using Lighthouse, I was given a list of problems, which while relatively simple, serve a major purpose.

Accessibility

The first issue I had was that my pages did not have titles. Titles are used to help describe what page the user is looking at. This also makes the website look very unprofessional.

Here was the issue

- ▲ Document doesn't have a `<title>` element

The reason this didn't work was because I had set the variable `{{title}}` in the title but had forgotten to implement this in `app.js`.

This was the fix:

```
// injecting index into layout
app.get('/', (req, res) => {
  res.render('partials/index', {
    layout: 'layouts/layout',
    title: 'Alex Purser Portfolio'
});
```

Before fixing this, the title for each page was this:



After fixing this, the title for each page was this:



The second issue was that my html did not have a lang (language) tag. Lang tags are useful for helping search engines identify and return language specific results.

Here was the issue

- ▲ `<html>` element does not have a `[lang]` attribute

And here was the fix

```
<!DOCTYPE html>
<html lang="en"> ...
</html>
```

SEO

Aria-labels

Since I used icons for my social media links, screen readers wouldn't be able to infer what these mean. Aria-labels solve this problem by adding a text label to the icon. Now, the screen reader reads out the aria label for each icon.

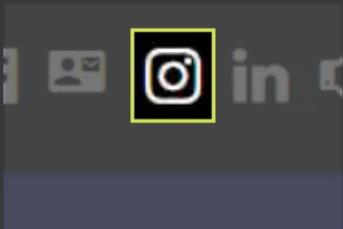
This was a crucial flaw in my accessibility, which could have dampened user experience.

This error repeated for all icons, which I have now fixed.

▲ Links do not have a discernible name

Link text (and alternate text for images, when used as links) that is discernible, unique, and focusable improves the navigation experience for screen reader users. [Learn how to make links accessible](#).

Failing Elements



`a.fa.fa-instagram`

`aria-label="Instagram">>`

SEO stands for Search Engine Optimisation. To summarise, SEO determines how likely a website is to be found on google, thanks to <meta> tags.

Before using Lighthouse, I had completely missed these tags, and had a low SEO because of it.

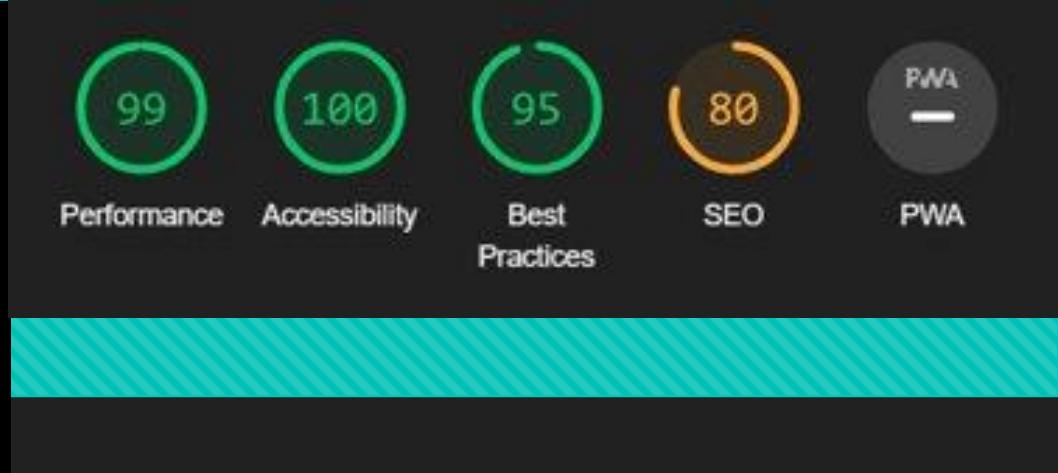
These were the issues found:

- ▲ Does not have a <meta name="viewport"> tag with width or initial-scale |
- ▲ Document does not have a meta description

These were the solutions:

```
<meta name="viewport" content="width=device-width, initial-scale=1" />
<meta name="description" content="{{description}}">
```

```
// injecting index into layout
app.get('/', (req, res) => {
  res.render('partials/index', {
    layout: 'layouts/layout',
    title: 'Alex Purser Portfolio',
    description: 'Computer Science student at Falmouth University.
  });
})
```



To continue with the flexibility of hbs, alongside making the title a variable, I made the description a variable. As a result, each page has its own unique title and description, and doesn't have to be set per page, just in app.js



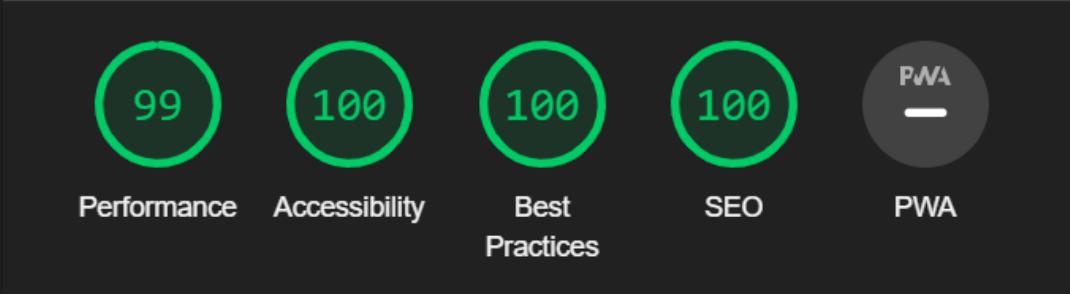
Performance

Performance

Opportunity	Estimated Savings
■ Eliminate render-blocking resources	0.94s ^
Cloudflare CDN [Cdn]	6.1 KiB 470 ms
...css/fontawesome.min.css (cdnjs.cloudflare.com)	6.1 KiB 470 ms

After testing the website, I noticed that running the Lighthouse test and then switching windows caused a drop in performance. The error suggested summed up states that the social media icons take a while to load up. This is probably because its registered as a link, which the website then needs to access, identify the required classes, and then output them to the user.

To resolve this issue, I would implement critical inline css. This is where css is generated in the head of the website. Using this, I would generate the social media icons before the rest of the page, which should improve the performance.



Mobile Optimisation

- Once again using Lighthouse, I tested the mobile quality of my website.
- Fortunately, it came out with these near perfect results, which I believe to be the effect of the previous optimisations.

References:

- [1] 'WebAIM: Contrast Checker'. Accessed: Jan. 01, 2024. [Online]. Available: <https://webaim.org/resources/contrastchecker/>
- [2] 'Sliding Diagonals Background Effect'. Accessed: Jan. 02, 2024. [Online]. Available: <https://codepen.io/chris22smith/details/RZogMa>
- [3] 'Lighthouse'. Accessed Jan. 03, 2024. [Online]. Available: <https://chromewebstore.google.com/detail/blipmdconlkpinefehnmjammfjpmpbjk>