

SPAT GROUP REPORT

KRM

GROUP 9

**Elise Baxter
Thomas Constantine
Max Manoehoetoe
Quincy Majekadegbe**

Akila Munasinghe - Contributed to 1 Sprint (Sprint 2)

No contribution: - Muhammad Patel
• Sufyan Kotwal

<http://hc22-9.poseidon.salford.ac.uk>

Table Of Contents

1. Introduction
 2. Description of the development process, for each sprint produce some evidence of:
 - Product Backlog
 - Sprint backlog
 - Work item estimates
 - Work item selection by group members
 - Daily stand-up meetings
 - Sprint burndown chart
 - Customer Demonstration meeting
 - Sprint retrospective
 3. Requirements analysis and system design
 - User stories and product backlog
 - Overall system architecture
 - System designs such as UML diagrams, database design, etc.
 4. Description of the solution implementation
5. Details of the solution testing and evaluation
 - Acceptance test cases
 - Unit tests
 - Test results
6. Conclusions, summary and evaluation of the achieved results
7. References
8. Appendices
 - Detailed evidence of daily SCRUM stand-up meetings
Date, time, attendees, decisions, actions

Introduction

During this assignment we underwent the participation of Hackcamp 2022. Within this assignment Group 9 decided to follow a agile approach using a scrum technique.

This was followed by iterating the following stages: - Plan, Requirements, Design, implement, test, Deploy.

These stages were iterated across 3 sprints over the course of 2 weeks.

By using this Agile technique it allows us to gain constant feedback from the client and to make sure the final product is aligned with their ideas.

Group 9 got assigned to the company KRM who provided a brief on what system they would like creating. The brief outlined that they required an onboarding system

SPRINT 1

Product Backlog

The screenshot shows a Trello board with the following structure:

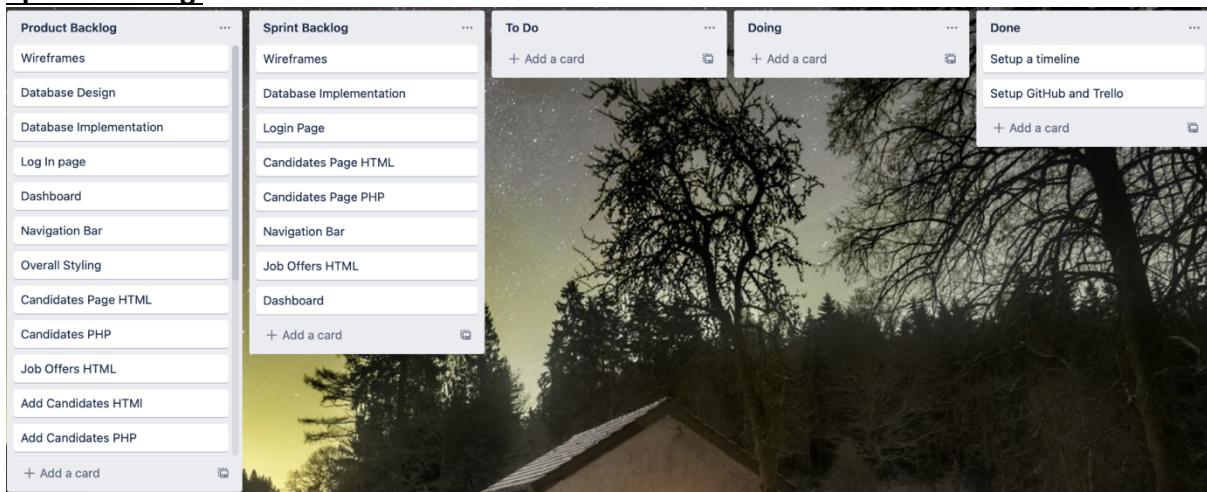
- Product Backlog:** Wireframes, Database Design, Database Implementation, Log In page, Dashboard, Navigation Bar, Overall Styling, Candidates Page HTML, Candidates PHP, Job Offers HTML, Add Candidates HTML, Add Candidates PHP, + Add a card.
- Sprint Backlog:** Wireframes, Database Implementation, Login Page, Candidates Page HTML, Candidates Page PHP, Navigation Bar, Job Offers HTML, Dashboard, + Add a card.
- To Do:** + Add a card.
- Doing:** + Add a card. (One card: "Setup GitHub and Trello").
- Done:** Setup a timeline, Setup GitHub and Trello, + Add a card.

The screenshot shows a Trello board with the following structure:

- Product Backlog:** Job Offers HTML, Add Candidates HTML, Add Candidates PHP, Candidates Trying Page HTML, Candidates Training Page Styling, Admin Page HTML, Admin Page PHP, To Do list HTML, To do List PHP, Individual Candidates Profiles, Checklist for candidates HTML, Checklist for candidates PHP, + Add a card.
- Sprint Backlog:** Wireframes, Database Implementation, Login Page, Candidates Page HTML, Candidates Page PHP, Navigation Bar, Job Offers HTML, Dashboard, + Add a card.
- To Do:** + Add a card.
- Doing:** + Add a card. (One card: "Setup GitHub and Trello").
- Done:** Setup a timeline, Setup GitHub and Trello, + Add a card.

We decided to use a trello board as our product backlog so that everyone could refer back to the same task board to ensure we all have the right frame of mind. As this the first sprint, We created the majority of the product backlog which included everything which we wanted to create over the course of the 2 weeks. We decided that we would break down all the requirements using decomposition into smaller, more manageable task. As prior to the first sprint we did a skills check on all members of our team and we felt as though it would be unfair giving PHP tasks to someone who did not feel confident in this area. By breaking down the tasks into HTML, PHP and SQL, it ensure that everyone within the team would be included no matter their ability and what they felt more confident in.

Sprint backlog



Within the first sprint we decided that we wanted to get the overall template of the system completed and we decided to focus on a select few pages to ensure we had a good amount of progress to deliver to the client at the end of the sprint during the demonstration. We decided to focus on implementing the database, login system, Navigation bar, Dashboard, Candidates list and job offers page.

We felt as though through talking to the client these were the main focus of the system so our thought to get them implemented first ensured that we would receive the most feedback and development on these features to ensure that matched the needs of the client.

We decided to include the sprint backlog within trello so that the team members were able to move their given tasks along the timeline so that we could all stay up to date with the progress everyone was making and to ensure no element would get left behind.

Work item estimates and Work item selection by group members

During the first sprint, We agreed as a team that the best way to tackle the first sprint would be to each to chose a wireframe in which we had created and to each implement a different wireframe, whilst having someone focus on the styling and Someone focusing on the implementation of the database.

We decided it would be fair to allow each team member to select a task in which they felt confident with due to some of our team members feeling weaker on php than html, although support was always offered by the more experienced members of the team.

The following Tasks were selected by the following group members:

Max – Dashboard and Overall Styling - Estimated 3 days

Thomas – Database Implementation - Estimated 4 Days

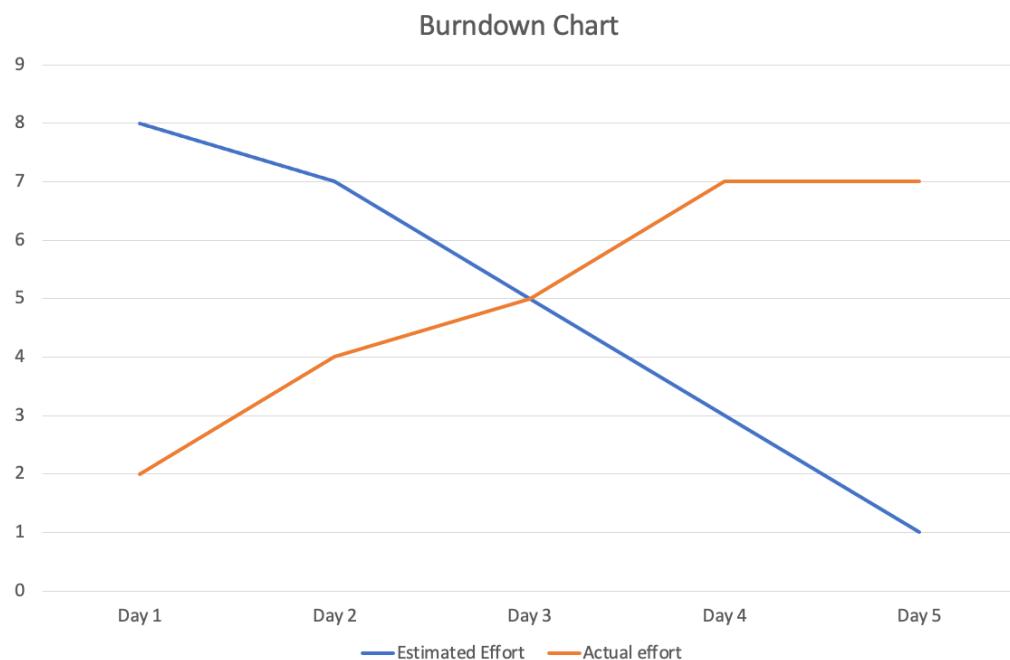
Elise – Jobs Offers page - Estimated 2 days

Quincy – Candidates Page – Estimated 3 days

Daily stand-up meetings

Throughout the Daily Stand up meetings it became clear that our group of 7 was quickly becoming a group of 4 due to the lack of attendance from certain team members. This proved as a difficult issue for our team as we always wanted to include everyone in the decisions for this project due to the nature of the team-working assignment but they also agreed to complete work which they didn't end up completing which set us back.

Sprint burndown chart



We expected the effort form our team to decrease as time went on, however we experienced the opposite with this. On the first day we witnessed little effort from within the team, lack of communication being a key issue here. However, as time went on and fellow teammates picked up work which other people missed, as per the dip in the middle of the graph, we witnessed an increase in team effort from other people within the team as they realised they would have to work harder to maintain the same effort from fewer members.

Customer Demonstration meeting

Feedback from Client

On Friday 14th January we ended our first sprint with a demonstration to the client. Through this demonstration we demonstrated a walkthrough of the system step-by step.

We gain very positive feedback from the client demonstrating that she liked how we did not take styling as an after though and we had though about it from the front set. We cleared up our ideas with the client on what she expect from the training page as we had the wrong understanding on what she required from this page and after discussing the client, we cleared this up as being a page where she can establish whether there is any more training a candidate feels like they require to be suitable for their given job offer.

Reflection for next sprint

After gaining the insight that we were on the right track, we got together as a team to gather our thoughts on how we will tackle the next sprint through the feedback provided. From reflecting on the feedback we established that we needed to match our system to the colour screen of the clients website, we needed to create an admin page where the higher users would be able to create users with different access levels and we needed to create our training page.

Sprint retrospective

What went well?

During our Last daily meeting we reviewed what we all thought went well during our first sprint. The main factor which we though went well was the speed and efficiency we had in creating a basic overall systems and we put that down to the effort and hard work of all the individuals who decided to take part in the first sprint.

What needs Improving for the next sprint?

A major downfall within the first sprint was the communication between the team. We decided during our last daily stand-up meeting of the sprunt that we would all be honest and review how it went. Majority of people said the main issue within our team was communication. This was seen as an issue due to work then being completed incorrectly as people didn't attend the meetings or stay in contact so they didn't have a clear understanding of the brief.

To improve this for the next sprint, we decided to try a new form of contact by creating a WhatsApp group instead of messaging through apps. By creating this form of contact we believe that everyone will be able to communicate better throughout the next sprint.

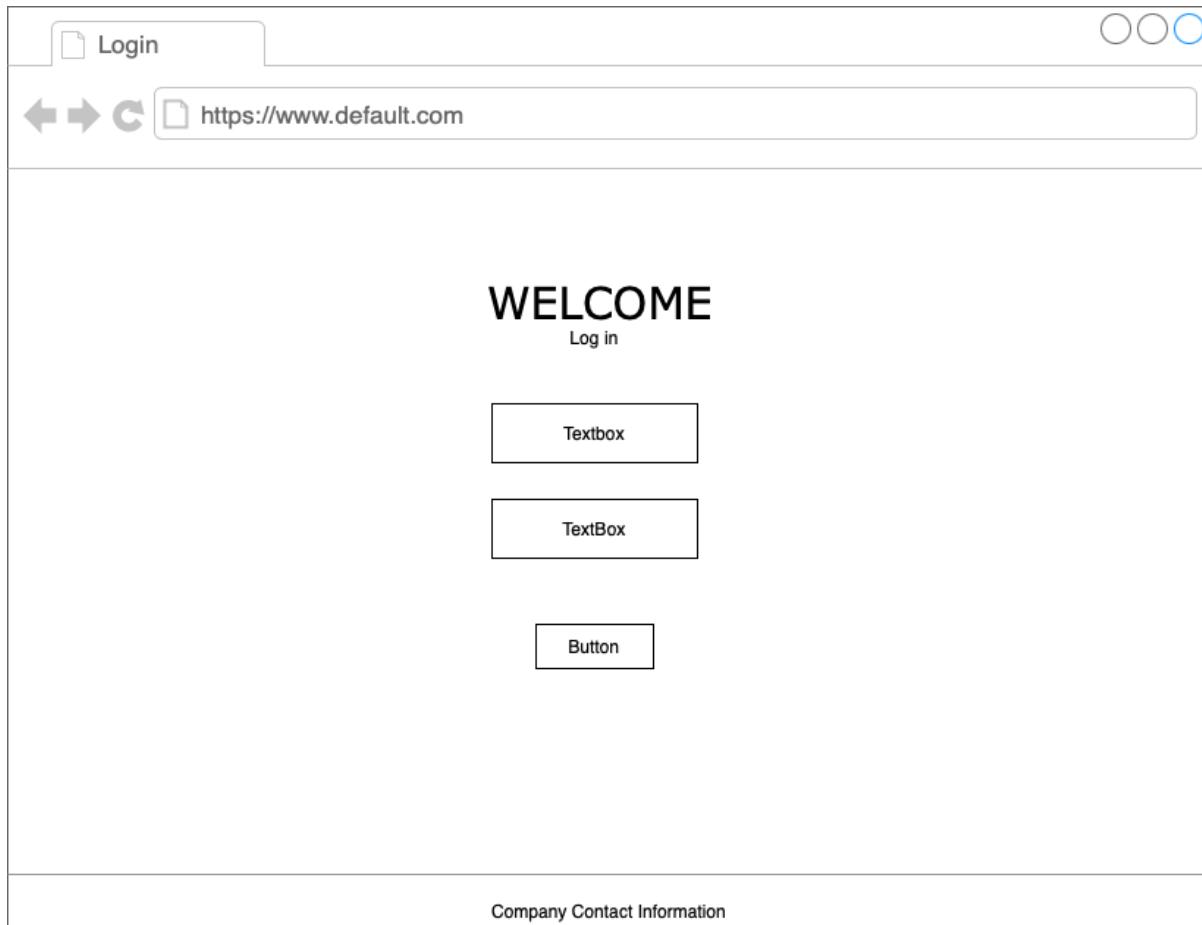
Another issue we faced during the first sprint, was working being evenly separated between the team and trust being put into those working from home and not communicating. This proved as an issue as it would get to the last day of the sprint when the demonstration was due in a matter of hours and they would reveal they hadn't done the work. This proved very stressful for other team members as they to complete additional work which needed to be completed before the client demonstration to ensure we stayed on schedule. To improve this for the next sprint, I believe we should not delegate the work equally but instead make sure there are enough tasks available for everyone and let people select which task they wanted to complete if they were committed to complete them. This would hopefully ensure we wouldn't fall short at the next customer demonstration.

Sprint 1: Requirements and User Stories

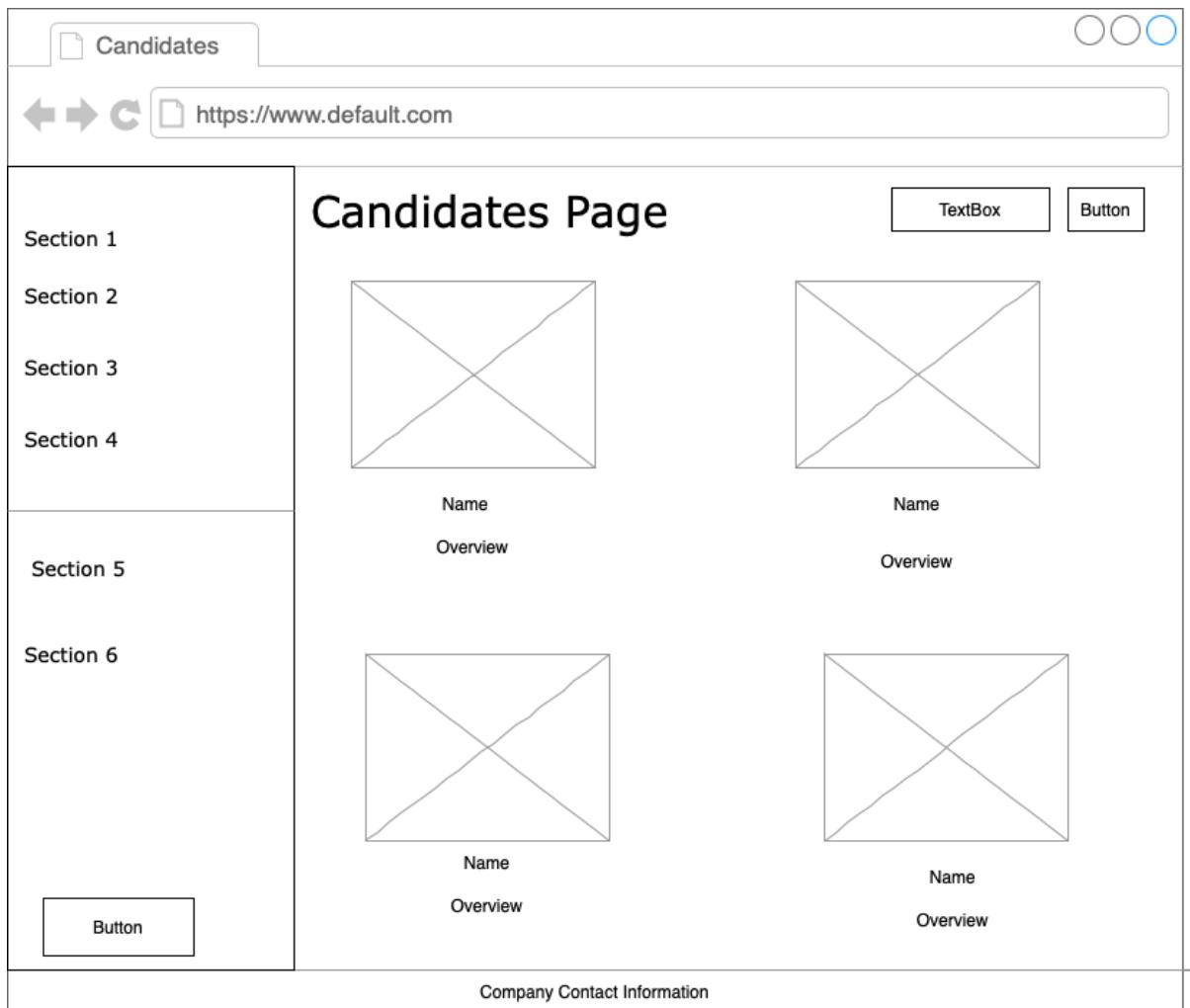
1. As a company user, I want to be able to view how many people are onboarding at that current time so that I can see how much my team is expanding.
2. As a company user, I want to be able to view the Monthly statistics for the onboarding so that I can compare the monthly progress and check we are meeting the targets.
3. As a company user, I want to be able to view the training each candidate has received so that I can see if there is more course I could provide to them.
4. As a company user, I want to be able to add a candidates information so that I can keep track of all of their details and jobs which have been agreed.
5. As a company user, I want to be able to edit candidate information so that I can ensure the data we keep is up to date.
6. As a company user, I want to be able to delete candidates so that I can ensure we do not keep data for longer than we should do If they suddenly stop onboarding.
7. As a company User, I want to be able to view an overall dashboard to ensure I can quickly get up to date with all the information
8. As a company user, I want to be able to view candidates information so that I know who I will be meeting at the interview and know what job they have been offered.
9. As a company User, I want to be able to view Job offers so that I know what offers have been agreed.
10. As a company User, I want to be able to update Job offers so that they are up to day and valid.
11. As a company user, I want to be able to log into this website so that users who do not have permission to access this data can not view it.
12. As a company user, I want to be able to create a checklist for each candidate so I can keep track of whether they need a contract creating or any work related items such as software's and laptops sorting out.

Sprint 1: Design Process

Wireframes:



This wireframe is the first in which the user would see when accessing the system. To make sure the company abides by GDPR we wanted to make sure the data contained within this system is kept secure. Therefore, we wanted to create a log in screen to make sure only the authorised users had access. To create this, we will include 2 textboxes, one for username and one for password. We would also include a button for once pressed it would compare the user inputs to what is stored within the database.



Within the brief, the company wanted to be able to view all the candidates which were in the onboarding process and to be able to see a quick overview of these. We therefore decided to create a page devoted to the candidates. The Textbox and button in the top right corner would create a search function. When the button is pressed, it would compare the user input to the database and narrow down the candidates to those with the same name as searched.

Database implementation:

Setting up the database:

After designing the database, I decided that the best way to manage and set up the database was to use a web-based client called phpMyAdmin, this allows me to easily manage the database and presents all the data in a way that is easy to read. To setup each of the tables I used the code that is shown below.

```
Table users {  
    userID int [pk, increment] // auto-increment  
    firstName varchar [not null]  
    lastName varchar [not null]  
    username varchar [not null]  
    password varchar [not null]  
    adminLevel int [not null]  
}  
}
```

```
Table candidates {  
    candidateID int [pk, increment]  
    firstName varchar [not null]  
    lastName varchar [not null]  
    phoneNumber varchar [not null]  
    email varchar [not Null]  
}  
}
```

```
Table JobOffers {  
    jobID int [pk, increment]  
    candidateID varchar [not null]  
    jobType varchar [not null]  
    Status varchar [not null]  
    dateStarted date [not null]  
}  
}
```

```
Table Interviews {  
    interviewID int [pk, increment]  
    jobOfferID varchar [not null]  
    dateStarted date [not null]  
    assignedTo varchar [not null]  
}  
}
```

ref: JobOffers(candidateID < candidates(candidateID

ref: Interviews(jobOfferID > JobOffers.jobID

ref: Interviews(assignedTo < users(userID

Each attribute to the table is followed by its corresponding data type and then any extra parameters that are needed. Each attribute has the “not null” to ensure that the table retains integrity and there is no missing data. At the bottom are the references that link foreign keys to corresponding primary keys in the original table.

```

<?php
class Database {
protected static $_dbInstance = null; // static instance
protected $_dbHandle;

public static function getInstance() {
    $username = 'hc22-9'; $password = 'xsWco02LRllYP5S';
    $host = '51.141.126.36'; $dbName = 'hc22_9_populated';

    if(self::$_dbInstance === null) { //checks if the PDO exists
        // creates new single instance if not, sending in connection info
        self::$_dbInstance = new self($username, $password, $host, $dbName);
    }
    return self::$_dbInstance;
}

private function __construct($username, $password, $host, $database) {
try {
    $this->_dbHandle = new PDO("mysql:host=$host;dbname=$database", $username, $password);
    // creates the database handle with connection info
}
catch (PDOException $e) { // catch any failure to connect to the database
    echo $e->getMessage();
}
}

public function getdbConnection() {
    return $this->_dbHandle; // returns the PDO handle to be used elsewhere
}

public function __destruct() {
    $this->_dbHandle = null; // destroys the PDO handle when no longer needed
}
}

```

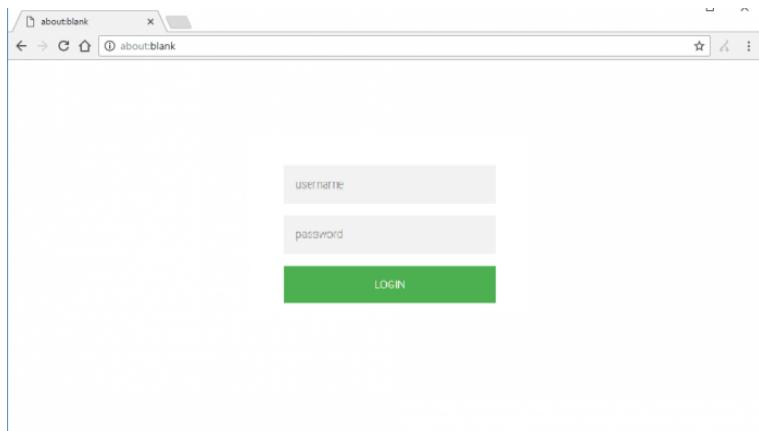
PHP implementation:

When linking the database to the pages I followed an MVC (Model-View-Controller) style of coding. To begin with this, I created a standalone file that contained the connection details to the database. This way I could just include the file in any of the pages that need it. This file uses a class model that contains the details of the SQL server. Inside the class there are three functions the `__construct` which handles creating a link to the database. `getdbConnection()` this just returns the PDO handle to be used elsewhere and then the last function is the `__destruct` function which destroys the connection when it is no longer needed.

Graphical Designs

Once we had developed our wireframes we then moved on to a more graphical design. This ensured all team members could gain a real insight into what we wanted to create and ensure that we were all on the same page. Max took a lead on creating the graphical designs, making sure they all followed the same format. During this stage, we created a colour scheme to follow from the website, however in the first sprint our main focus was to get the over all layout and design implemented before we took a focus on the colours and finer detail.

Home



Candidate List

A screenshot of a web browser window titled 'about:blank'. The left side features a dark sidebar with a list of items: 'Item 1', 'Item 2', 'Item 3', 'Item 4', 'Item 5', 'Link', 'Link', 'Link', 'Link', 'Link', and 'Link'. The right side is titled 'Candidates' and displays four identical profile cards. Each card contains a small blue placeholder icon for a user picture, followed by the name 'Elsie Ferguson' and the title 'Senior Designer'.

Job offers

A screenshot of a web browser window titled "about:blank". The main content area displays two identical job offer cards. Each card has a header "Active job offers", followed by a section for "Job title : xxxxxxxx", "Salary: £xxxxxxxx", and "Info:" with placeholder text "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx". To the left of the cards is a sidebar containing a list of items: "Item 1", "Item 2", "Item 3", "Item 4", "Item 5", "Link", "Link", "Link", "Link", "Link", and "Link".

Active job offers

Job title : xxxxxxxx

Salary: £xxxxxxxx

Info:

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Job title : xxxxxxxx

Salary: £xxxxxxxx

Info:

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

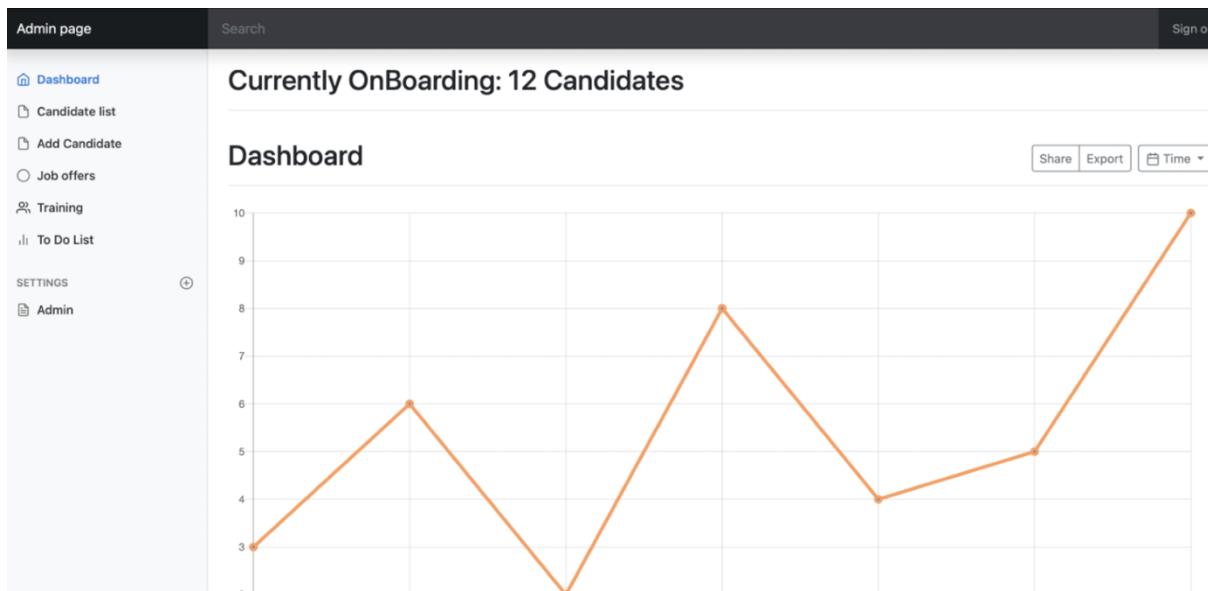
Description of the solution implementation

A screenshot of a login form titled 'Login' under 'Administrator Login'. It contains fields for 'Username' and 'Password', and a 'Login' button at the bottom. A red box highlights the entire form area.

This login screen was created by firstly creating a html form which could take a user input. Once we had the html we then added functionality through php by forming a session which would be launched when the log in button is hit. We used sql to check whether the user input matched the database details. If so they log in screen would continue to the system, however if the details were incorrect an error message would pop up.

A screenshot of an admin dashboard titled 'Admin page'. The left sidebar includes links for Dashboard, Candidate list (which is currently selected), Add Candidate, Job offers, Training, To Do List, and Admin. The main content area is titled 'Candidate list' and displays a table of candidate data. The table has columns: candID, firstName, LastName, phoneNumber, and email. The data is as follows:

The candidate list was first created by using HTML and CSS where the table was created and then styled to fit into the website after creating the HTML and CSS template the PHP could be started which involved using a SELECT query from the SQL database to select all the candidates and then display them in the relevant sections of the HTML and CSS template.



The dashboard seen above was created using HTML CSS and JavaScript. The HTML and CSS styling was done for the website first and then the JavaScript needed for the graph was then added

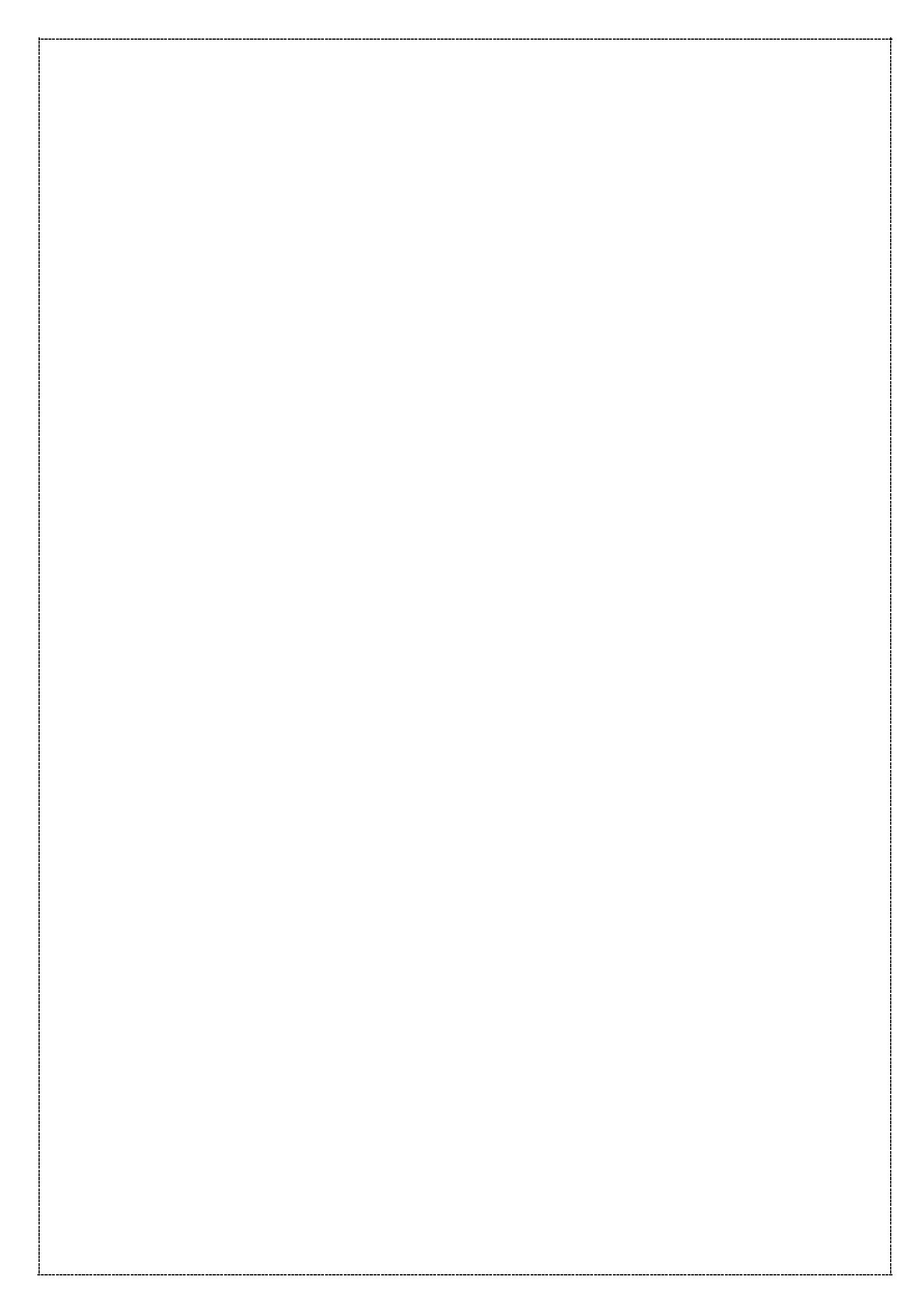
Admin page		Job offers		
#	Job Title	Number of candidates	Last Updated	
1	Software engineer	5	12th January 2021	
2	Programmer	2	10th January 2021	
3	HR	3	15th January 2021	

The job offers page was created by first using HTML and CSS to create a basic template of what we want it to look like after doing this the PHP was then started which would check the job offers table in the SQL database and then display the number of candidates for each job type after that it would then get a timestamp of when this page was last updated so you know whether the information is up to date or not.

Test	Expected Result	Actual result	Evidence	Fixes (If needed)
Test the login functionality	Login and display dashboard	Password was incorrect	<div style="border: 1px solid #ccc; padding: 10px; width: fit-content;"> Password is incorrect <div style="text-align: right; margin-top: 10px;"> Close </div> </div>	Issue was fixed. There was a syntax error

				when checking the password against the one in the database					
Test the login functionality pt.2	Login and display the dashboard	Logged in and dashboard was displayed	password is correct	Close					
Test login functionality pt.3	Incorrect username	Incorrect username	Username is Incorrect	Close					
Test login functionality pt.4	Incorrect Password	Incorrect Password	Password is incorrect	Close					
Candidate List functionality	Display all candidates and all the relevant information	Displayed an error	<p>Candidate list</p> <table> <thead> <tr> <th>Error</th> <th>3629095426</th> <th>Error</th> <th>Error</th> <th>Error</th> </tr> </thead> </table>	Error	3629095426	Error	Error	Error	Issue with the SQL statement that retrieves the candidate information
Error	3629095426	Error	Error	Error					

Candidate List pt.2	Displays all candidates and all the relevant information	Displays all candidates and all the relevant information	<p>Admin page</p> <ul style="list-style-type: none"> Dashboard Candidate list Add Candidate Job offers Training To Do List <p>SETTINGS</p> <ul style="list-style-type: none"> Admin 	<p>Search</p> <h2>Candidate list</h2> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #f08080;"> <th>candid</th><th>firstName</th><th>lastname</th><th>phoneNumber</th><th>email</th></tr> </thead> <tbody> <tr><td>2</td><td>Robinetta</td><td>Guslon</td><td>7901778439</td><td>rguslon0@who.int</td></tr> <tr><td>3</td><td>Madelene</td><td>Whitehall</td><td>3814662038</td><td>mwhitehall1@myspace.com</td></tr> <tr><td>4</td><td>Hewie</td><td>Town</td><td>1574999969</td><td>htown2@amazon.co.jp</td></tr> <tr><td>5</td><td>Solomon</td><td>Pringer</td><td>3011364033</td><td>springer3@google.cn</td></tr> <tr><td>6</td><td>Theodore</td><td>Dunsire</td><td>2765645240</td><td>tdunsire4@yelp.com</td></tr> <tr><td>7</td><td>Vittorio</td><td>Rachuig</td><td>3994215840</td><td>vrachuig5@wp.com</td></tr> <tr><td>8</td><td>Baryram</td><td>Draayer</td><td>8169077567</td><td>bdraayer6@youtu.be</td></tr> <tr><td>9</td><td>Barrie</td><td>Lowle</td><td>6124375950</td><td>blowle7@baidu.com</td></tr> <tr><td>10</td><td>Pryce</td><td>Guitton</td><td>8372632178</td><td>pguitton8@paypal.com</td></tr> <tr><td>11</td><td>Amil</td><td>Boothby</td><td>5486032732</td><td>aboothby9@mozilla.com</td></tr> <tr><td>12</td><td>Pietra</td><td>Hallut</td><td>8881265801</td><td>phalluta@slashdot.org</td></tr> <tr><td>13</td><td>Anna-diana</td><td>Caw</td><td>3194042559</td><td>acawb@mediafire.com</td></tr> <tr><td>14</td><td>Norean</td><td>Rockhill</td><td>4002500695</td><td>nrockhillc@mozilla.org</td></tr> <tr><td>15</td><td>Nobie</td><td>Syde</td><td>8446356033</td><td>nsyded@pcworld.com</td></tr> <tr><td>16</td><td>Josepha</td><td>Harcarse</td><td>3629095426</td><td>jharcasee@berkeley.edu</td></tr> </tbody> </table> <p style="text-align: right;">Search</p>	candid	firstName	lastname	phoneNumber	email	2	Robinetta	Guslon	7901778439	rguslon0@who.int	3	Madelene	Whitehall	3814662038	mwhitehall1@myspace.com	4	Hewie	Town	1574999969	htown2@amazon.co.jp	5	Solomon	Pringer	3011364033	springer3@google.cn	6	Theodore	Dunsire	2765645240	tdunsire4@yelp.com	7	Vittorio	Rachuig	3994215840	vrachuig5@wp.com	8	Baryram	Draayer	8169077567	bdraayer6@youtu.be	9	Barrie	Lowle	6124375950	blowle7@baidu.com	10	Pryce	Guitton	8372632178	pguitton8@paypal.com	11	Amil	Boothby	5486032732	aboothby9@mozilla.com	12	Pietra	Hallut	8881265801	phalluta@slashdot.org	13	Anna-diana	Caw	3194042559	acawb@mediafire.com	14	Norean	Rockhill	4002500695	nrockhillc@mozilla.org	15	Nobie	Syde	8446356033	nsyded@pcworld.com	16	Josepha	Harcarse	3629095426	jharcasee@berkeley.edu
candid	firstName	lastname	phoneNumber	email																																																																																
2	Robinetta	Guslon	7901778439	rguslon0@who.int																																																																																
3	Madelene	Whitehall	3814662038	mwhitehall1@myspace.com																																																																																
4	Hewie	Town	1574999969	htown2@amazon.co.jp																																																																																
5	Solomon	Pringer	3011364033	springer3@google.cn																																																																																
6	Theodore	Dunsire	2765645240	tdunsire4@yelp.com																																																																																
7	Vittorio	Rachuig	3994215840	vrachuig5@wp.com																																																																																
8	Baryram	Draayer	8169077567	bdraayer6@youtu.be																																																																																
9	Barrie	Lowle	6124375950	blowle7@baidu.com																																																																																
10	Pryce	Guitton	8372632178	pguitton8@paypal.com																																																																																
11	Amil	Boothby	5486032732	aboothby9@mozilla.com																																																																																
12	Pietra	Hallut	8881265801	phalluta@slashdot.org																																																																																
13	Anna-diana	Caw	3194042559	acawb@mediafire.com																																																																																
14	Norean	Rockhill	4002500695	nrockhillc@mozilla.org																																																																																
15	Nobie	Syde	8446356033	nsyded@pcworld.com																																																																																
16	Josepha	Harcarse	3629095426	jharcasee@berkeley.edu																																																																																
Dashboard functionality	Displays the graph correctly	Displays the graph correctly	<p>Admin page</p> <ul style="list-style-type: none"> Dashboard Candidate list Add Candidate Job offers Training To Do List <p>SETTINGS</p> <ul style="list-style-type: none"> Admin 	<p>Search</p> <p>Currently OnBoarding: 12 Candidates</p> <p>Dashboard</p> <p style="text-align: right;">Share Export Time ▾</p>																																																																																
Job offers functionality	Displays jobs offered	Displays jobs offered	<p>Admin page</p> <ul style="list-style-type: none"> Dashboard Candidate list Add Candidate Job offers Training To Do List <p>SETTINGS</p> <ul style="list-style-type: none"> Admin 	<p>Search</p> <h2>Job offers</h2> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>#</th> <th>Job Title</th> <th>Number of candidates</th> <th>Last Updated</th> </tr> </thead> <tbody> <tr><td>1</td><td>Software engineer</td><td>5</td><td>12th January 2021</td></tr> <tr><td>2</td><td>Programmer</td><td>2</td><td>10th January 2021</td></tr> <tr><td>3</td><td>HR</td><td>3</td><td>15th January 2021</td></tr> </tbody> </table>	#	Job Title	Number of candidates	Last Updated	1	Software engineer	5	12th January 2021	2	Programmer	2	10th January 2021	3	HR	3	15th January 2021																																																																
#	Job Title	Number of candidates	Last Updated																																																																																	
1	Software engineer	5	12th January 2021																																																																																	
2	Programmer	2	10th January 2021																																																																																	
3	HR	3	15th January 2021																																																																																	



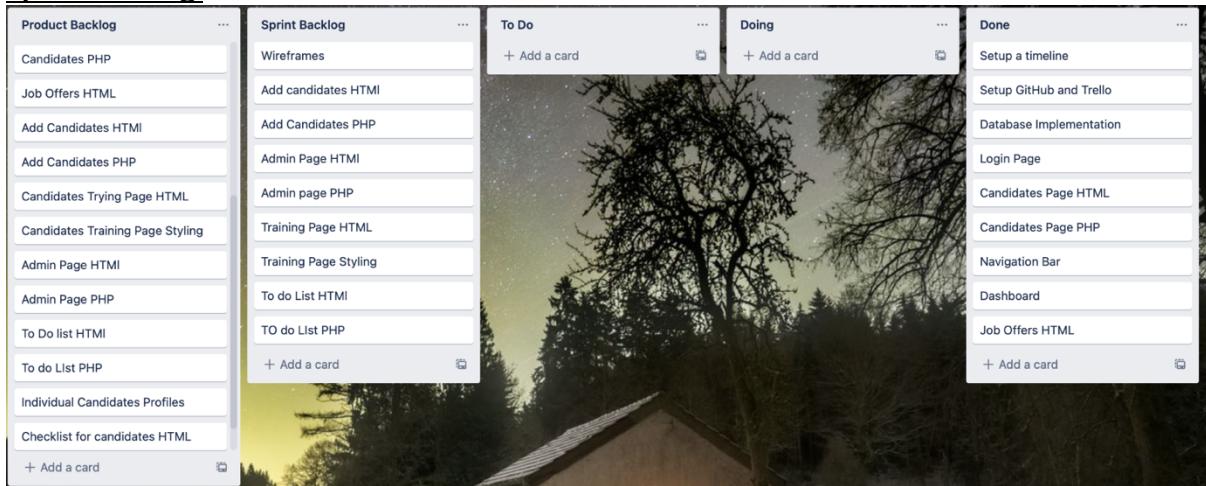
SPRINT 2

Product Backlog

During sprint 2, we altered the product backlog to include everything which it included in sprint 1 and then made some additions to include the clients feedback from the first sprint within our final product. Although the client didn't have any constructive feedback towards she trusted that our next intentions were along the right line and cleared up some doubts we had about the brief.

By finishing this client demonstration we were able to add to alter the product backlog to show more precise tasks to provide more clarity for our team members.

Sprint backlog



Within our second sprint, we felt as though we would be able to take on more tasks within the sprint. We therefore decided to spend this sprint implementing 4 more pages to our system, these being, Add candidates, Admin page, Training page and To do list. We decided to separate these tasks into smaller tasks by separating the functionality from the html to provide task for those who feel more confident with html or php.

Work item estimates & selection

Within the sprint backlog we imagined that the training page would require the majority of time for this sprint due to the individual profiles which required to be implemented. Within these individual profiles we wanted to include a wheel chart to represent the amount of training which the candidates. The following sprint estimations were predicted by the teams:

Wireframes - $\frac{1}{2}$ Days - Assigned to Elise

Add candidates HTML - 1 Day - Assigned to Elise

Add candidates PHP - 1 Day - Assigned to Elise

Admin Page HTML - 1 Day - Assigned to Thomas

Admin Page PHP - 1 Day - Assigned to Thomas

Training Page HTML - 3 day - Assigned to Max

Training Page Styling - 3 Day - Assigned to Max

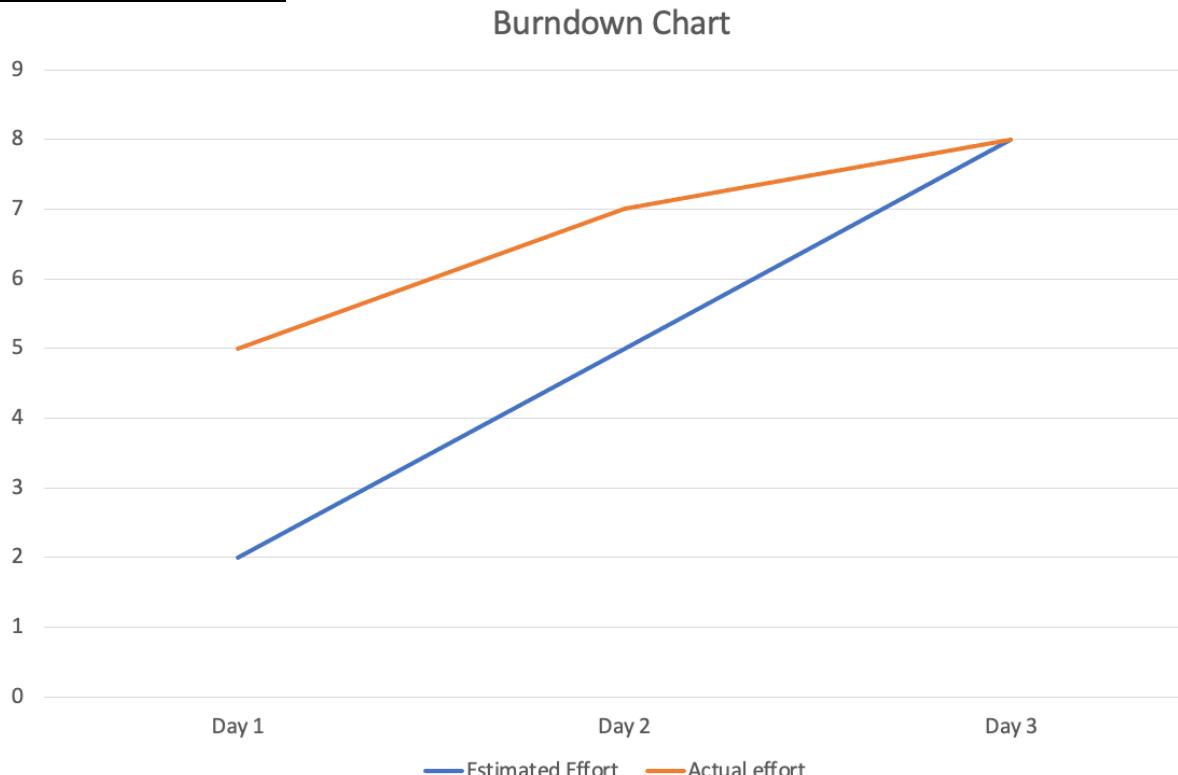
To do list HTML - 2 Day - Assigned to Akila

To do list PHP - 2 Day - Assigned to Akila

Daily stand-up meetings

During this sprint we found these stand-up meetings extremely helpful. During these meetings we were able to catch up on

Sprint burndown chart



We expected the estimated effort to reflect the effort portrayed within the last sprint, but what we actually witnessed was us entering sprint 2 at a much higher effort level than we entered sprint 1. Upon reflection we think this is due to us knowing the amount of effort we would have to put in each day to achieve the outcome as this could had been unclear at the start of sprint 1. Overall, with the graph being a positive correlation, starting at a higher point, this has a positive effect on what we were able to achieve.

Customer Demonstration meeting

Feedback from client

On Wednesday 19th January, we ended our second sprint with a client demonstration.

Within this demo we walked through our new elements with the client, including the training page and admin page which were the new additions within this sprint.

From gaining feedback from the client, she really appreciated our thought of including a graph to portray the amount of training a candidate has had.

We also found through feedback that we had gained the wrong approach with the to-do list.

She wanted the to do list to connect to individual candidate profiles and not to just be a separate to do list.

Reflection for next Sprint

On reflection from the feedback, we received we held a team meeting to discuss our next steps. We believed to make the most of the last sprint that we should firstly create separate candidate profiles in which we could connect the to-do lists to and then focus on neatening up the overall styling.

Sprint retrospective

What went well?

Within this Sprint the amount of research which was conducted to make the training page functional, and a high level of styling was a great factor contributing towards this sprint. It showed a high level of dedication from our team members and a true passion was shown towards achieving the expected outcome for the client. This was later appreciated from our client who really liked the ideas we implemented for this page.

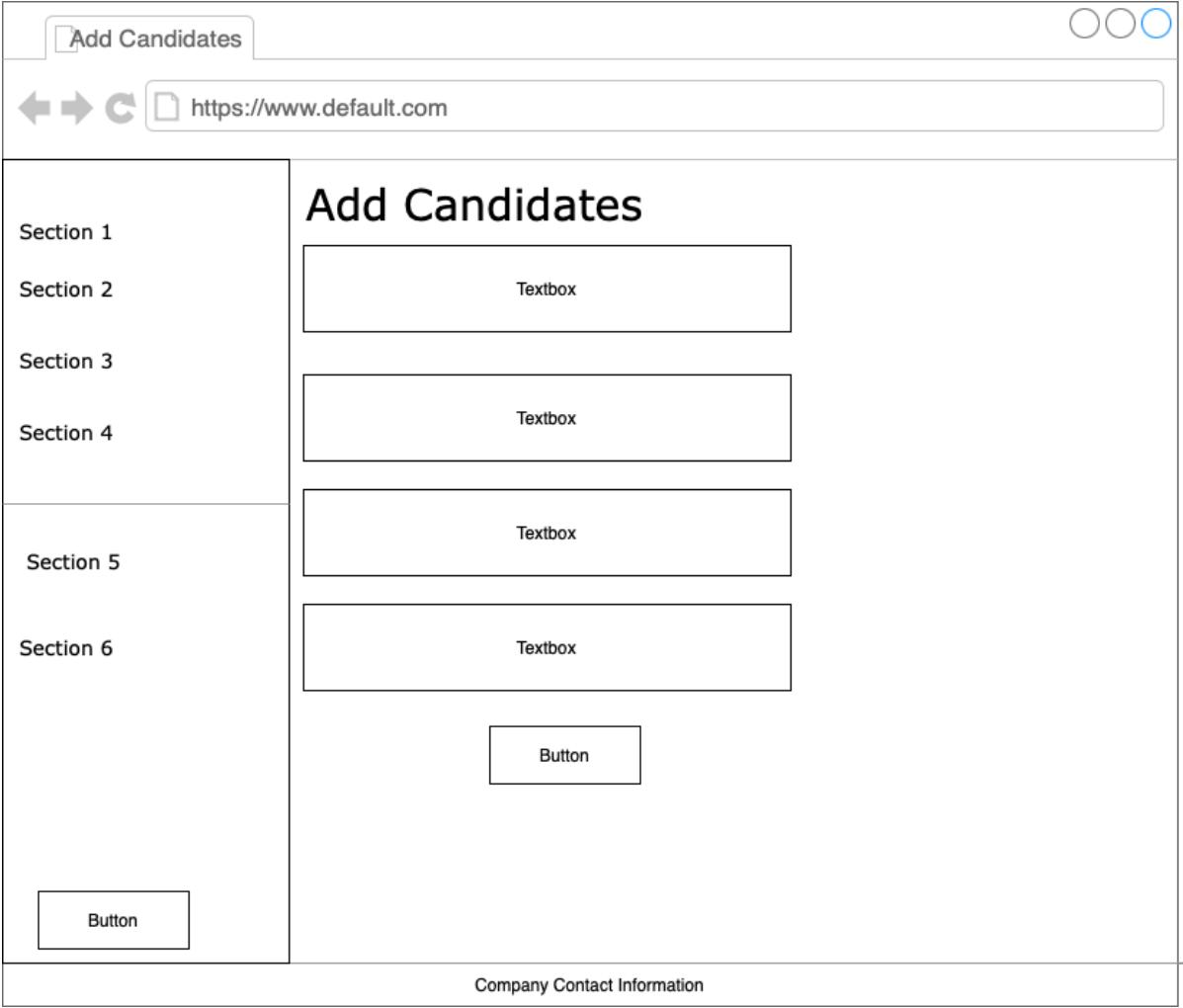
What needs improving?

Unfortunately, the new idea of creating the WhatsApp group did not work and there remained 3 people who still did not remain in contact and work as proactively as the other team members. To avoid the team from missing out over the last 3 days of coding we decided the best idea would be to press on and if they reached out to contact us, as we were proactively messaging in a group chat containing all members of the team, then we would help them in anyway way we could get them involved in the assignment.

Requirement's analysis and system design

1. As a user, I want to be able to add new candidates to the system so that I can keep the system up to date with new onboarding employees
2. As a user I want to be able to create a to do list so that I can keep track of tasks which I need to complete for each candidate.
3. As a user, I want the graph on the dashboard to match the styling of our company so that it has incorporated our brand colours
4. As an admin user I want to be able add different staff members so that they can access the information if I am away
5. As an admin user I want to be able to remove staff members incase they no longer work at KRM so that they will no longer have access to the data
6. As a user I want to be able to view the candidates training progress so that I know whether I can provide any more training to them.

Design

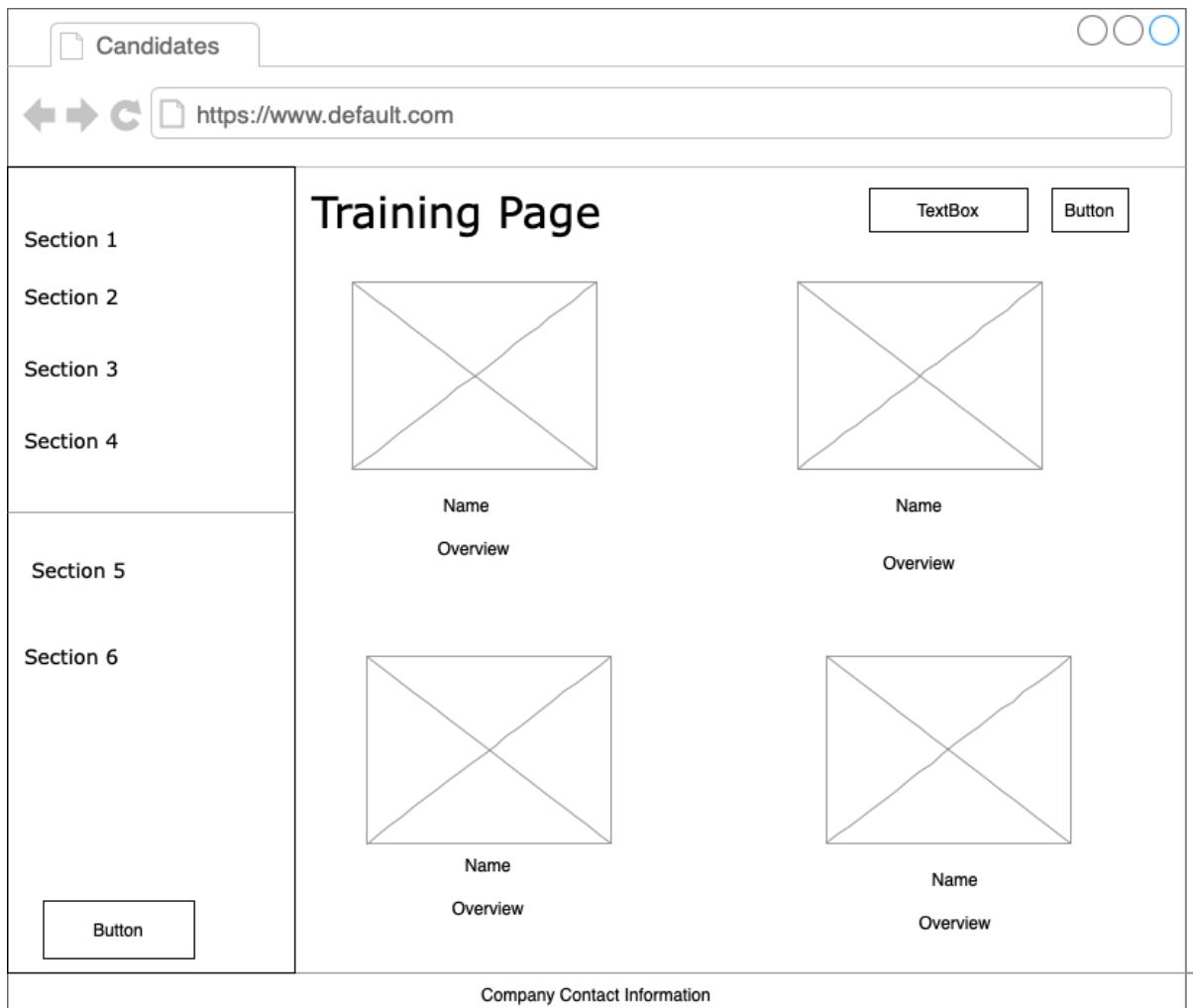


The wireframe illustrates the 'Add Candidates' page. At the top, there's a header bar with 'Add Candidates' and three blue circular icons. Below it is a navigation bar with back, forward, and refresh buttons, and a URL field showing 'https://www.default.com'. The main content area has a sidebar on the left containing 'Section 1' through 'Section 6'. To the right of the sidebar, the title 'Add Candidates' is centered above four 'Textbox' input fields. A single 'Button' is positioned below these textboxes. At the bottom of the page, a footer bar contains the text 'Company Contact Information'.

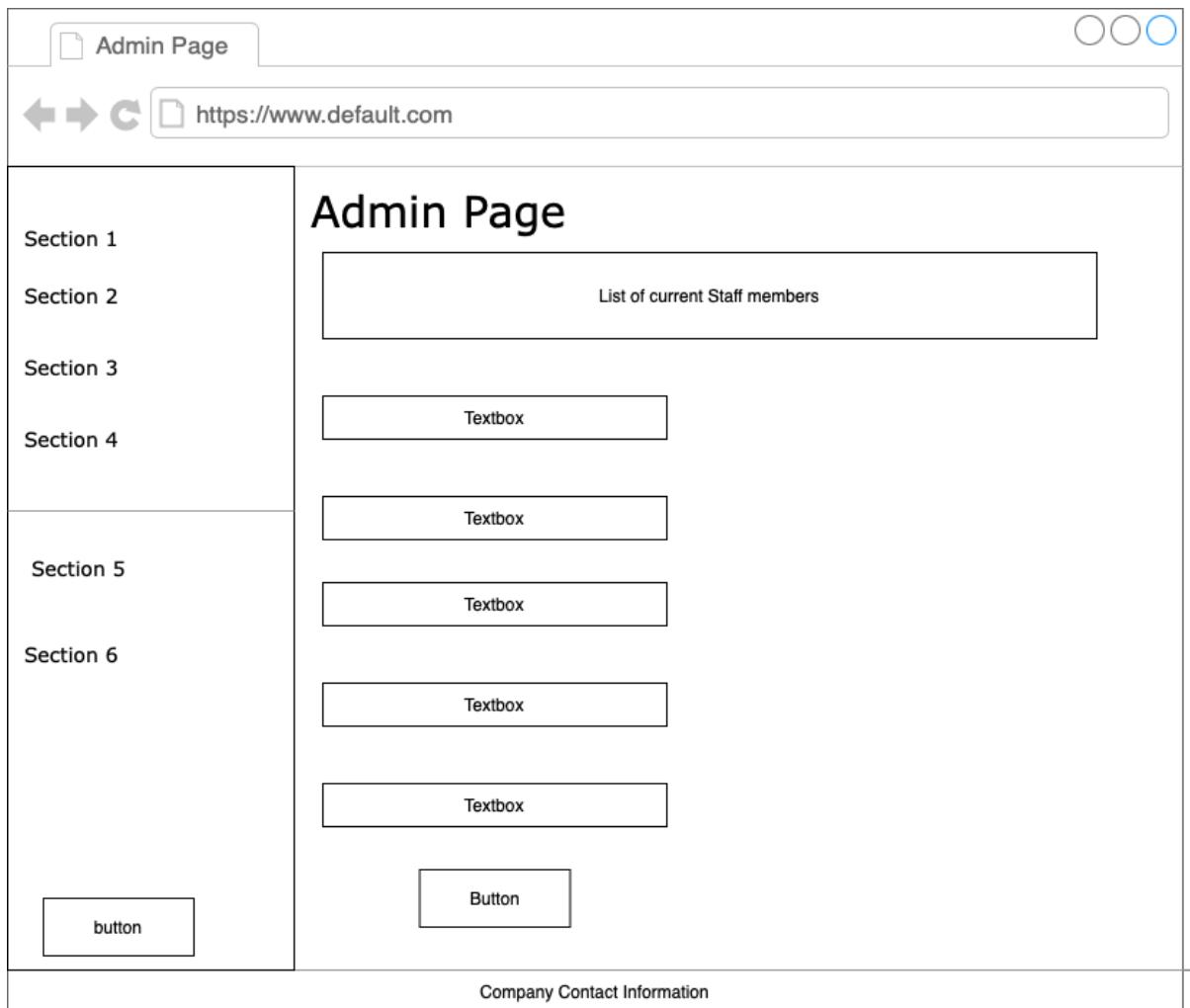
From the User stories we established that we needed to create a place within the system in which the user would be able to add candidates into the system which would then be stored within the database. We decided that we would implement this as a separate page in which would be accessible through the sidebar menu. Within this page there would be a title to provide the user with clarity on the current page and it would also include a form with a series of textbox answers. We decide to include 4 textboxes to collect the following information about a candidate: First name, Last Name, Email and phone number. There would then be a button underneath the form where the user can send this information into the database.



From the discussion with our client at the end of our first sprint we discussed the importance of implementing a to do list in the system to ensure that all tasks are completed and not forgotten about. We decided to implement this as a new page which could be accessed through the navigation bar at the side of the page. In the centre of this page there would be a box which would contain the elements of the to do list. Inside their would be a textbox where the user could enter new task and these would be added to the list by hitting the button and also any task which had already been added would be displayed within this box.



After clearing up the concept of the Training page with the client during the sprint 1 demo, we decided it would be implemented during sprint 2. We decided to get our thoughts drawn up into a wireframe. We decided to have a similar thought process as the individual client profiles but to use graphs where the photo boxes are indicated to portray how much training that individual candidate has gained prior to their new position at KRM. We also decided to have this page accessible by the navigation bar at the side.



As we had already built the majority of the system within the first sprint we decided as a team as we were going to have a shorter sprint, we should save time by not creating the graphical designs as we could picture the graphical from what we had designed within the first sprint.

Description of the solution implementation



We first started implementing the training page by creating a basic template for the candidate cards using HTML and CSS. We also created a progress wheel which works through the use of HTML and CSS as well. After creating this basic template we could then create the backend functionality using PHP, this involved pulling the candidate information from the database and then sending them to be displayed on the webpage. After creating the functionality with the database we then worked on displaying a template for each of the candidates in the database, this was done with PHP.

The screenshot shows the 'Add Candidate' section of the admin interface. The sidebar is identical to the previous one. The main area has a title 'Add Candidate' and a form with the following fields:

Please enter the following details for the candidate...	
First name	Last name
Phone Number	Email Address

Below the form is a 'Submit Candidate' button.

The add candidate page was created by first creating a basic template for the form using a combination of HTML and CSS, after creating this template we then added functionality to this by using PHP to connect to the database and insert the data submitted into the database.

The screenshot shows the Admin page interface. On the left, there's a sidebar with links: Dashboard, Candidate list, Add Candidate, Job offers, Training, To Do List (which is currently selected), SETTINGS, and Admin. The main area is titled "To Do List". It features a search bar, a text input field with placeholder "Add your new task" and a red "+" button, and a message "You have 0 pending tasks" with a "Clear All" button.

The To-Do list, like all the other pages, was first designed using HTML and CSS and then after the styling and html was done the actual functionality was added which is a combination of JavaScript and PHP which allows for this to work.

Details of the solution testing and evaluation

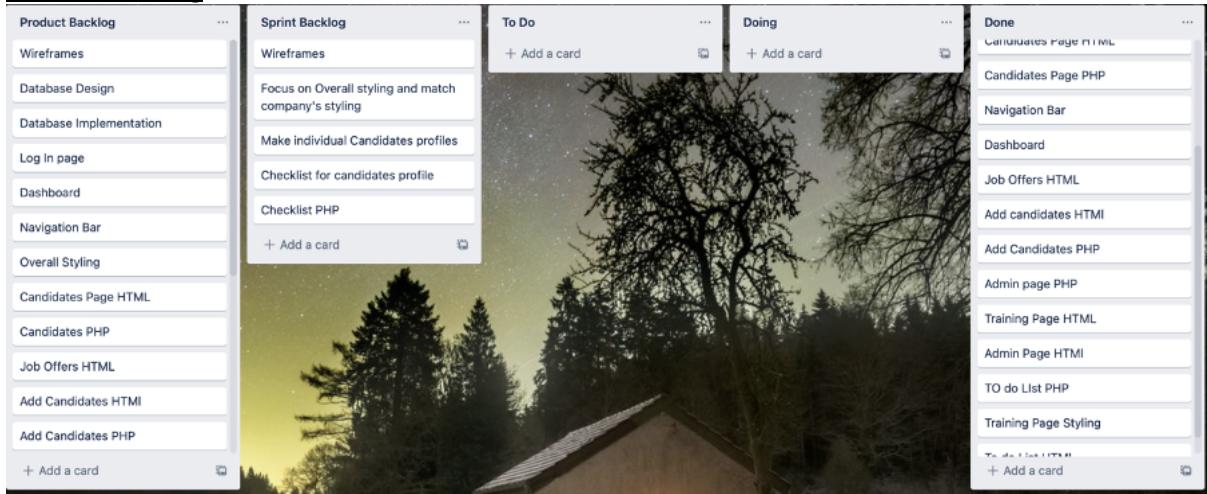
Test	Expected result	Actual result	Evidence	Fixes (If needed)
Add candidate	Expected to add candidate to the database	Error while adding candidate to the database	Insert Query failed: Incorrect data types	Data was being inputted into the wrong fields in the query. Swapping variables order fixed this
Add candidate pt.2	Expected to add candidate to the database	adds candidate to the database	Candidate sucessfully added to the database	Close
Admin page	Expected to display all users	Displayed an error	SQL ERROR: Cannot retrieve values	Error within the SQL query when retrieving the data. Wrong table selected

Admin page pt.2	Expected to display all users	displays all users	<p>Admin Page</p> <table border="1"> <thead> <tr> <th>#</th><th>First</th><th>Last</th><th>Username</th><th>Admin Level</th></tr> </thead> <tbody> <tr> <td>1</td><td>Aggie</td><td>Tuke</td><td>test</td><td>1</td></tr> </tbody> </table> <p>Username: <input type="text"/></p> <p>Password: <input type="password"/></p> <p>First name: <input type="text"/></p> <p>Last name: <input type="text"/></p> <p>Admin level: <input type="text"/></p> <p><input type="button" value="Submit"/></p>	#	First	Last	Username	Admin Level	1	Aggie	Tuke	test	1
#	First	Last	Username	Admin Level									
1	Aggie	Tuke	test	1									
Admin page pt.3	Expected to delete users	delete users	<p>User deleted</p> <p>Close</p>										
Admin page pt.4	Expected to add new users	add new users	<p>User Added</p> <p>Close</p>										
To-Do list	Expected to add new items to the to do list	add new items to the to do list	<h2>To Do List</h2> <p>Add your new task <input type="text"/> +</p> <p>New task</p> <p>You have 1 pending tasks Clear All</p>										
Training	Displays all the candidate profiles	Only displayed one candidate profile	<p>Training</p> <p>70%</p> <p>Robinette 7901778439 rguskind@who.int</p> <p>We switched from a while loop to a for each loop in order to fix this error</p>										

Training pt.2	Displays all the candidate profiles	Displays all the candidate profiles	 ID:2 Robinetta Guslon 7901778439 rguslon0@who.int	 ID:3 Madeline Whithall 3814662038 mwhithall1@myspace.com	 ID:4 Hewie Town 157499969 htown2@amazon.co.jp	 ID:5 Solomon Pringer 3011364033 springer3@google.cn	 ID:6 Theodore Dunsire 2765645240 tdunsire4@yelp.com
			 ID:7 Vittorio Rachulig 3994215840 vrachulig5@wp.com	 ID:8 Baryram Drayer 8169077567 bdrayer6@youtu.be	 ID:9 Barrie Lowle 6124375950 blowle7@baidu.com	 ID:10 Pryce Guilton 8572632178 pguilton8@paypal.com	 ID:11 Amil Boothby 5486032732 aboothby9@mozilla.com

SPRINT 3

Product Backlog



Sprint Backlog

For this sprint we decided on 5 different tasks that we needed to work on which are wireframes, overall styling, individual candidate profile's, checklist design, checklist functionality. The reason we did these in the last sprint is because we decided that these tasks were of least importance to the overall functionality of the website as well as this the overall styling was a suggestion that was put forward by the client. Again, we thought it would be the best idea to split certain tasks such as the checklist into a design and functionality task which would allow us to assign the design task to group members who have a good knowledge of html & CSS but not PHP and then assign the PHP to group members who have a good knowledge of PHP.

Work item estimates

Checklist PHP – 1 day – The php for the checklist we estimated would only take 1 day of the sprint as the code was simple as it just needed to edit the database and pull data from the database.

Overall styling – 2 days – The overall styling we estimated would take 2 days since there are a lot of different pages, and the CSS files are now quite large so changing different aspects could take a while.

Wireframes – 1 day – We estimated the wireframe would only take 1 day as they are quite simple to do.

Checklist HTML & CSS – 1 day – We estimated that the checklist HTML & CSS would only take 1 day as it was being added to the individual candidate profiles, so we didn't have to style its own page.

Individual candidate profiles – 2 days – We estimated the HTML, CSS and PHP for the candidate profiles would take 2 days as this could be more complicated and it was a new page.

Work item selection by group members

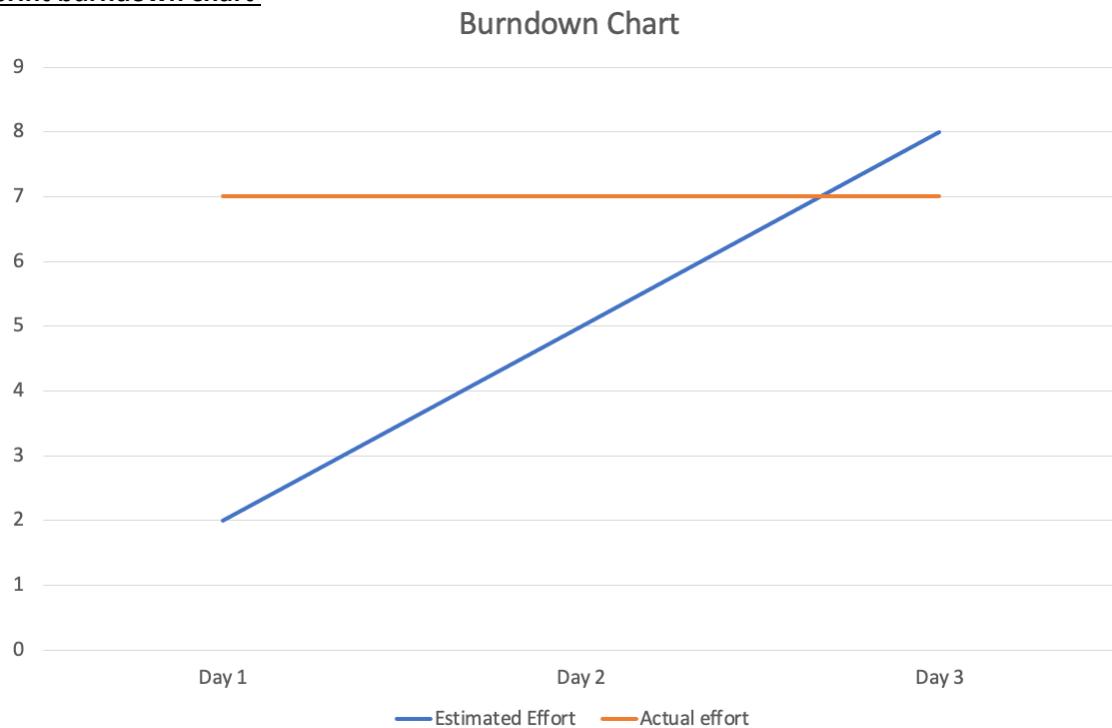
Tommy – Checklist PHP & Individual candidate profiles

Max – Overall styling based of client feedback

Quincy – Individual candidate profiles

Elise – Wireframes & Checklist HTML & CSS

Sprint burndown chart



By the time we had entered and finished sprint 3 we had maintained a steady level of high amounts of effort. Although by this point in the assignment we had lost 3 member of our team, the remaining 4 of us had gained the most effort during our last sprint and managed to finish the assignment on a high note.

Customer Demonstration meeting

After showing the demonstration to the client we got very positive feedback about what we had created so far. The client requested that we send a screenshot of each page and then she would write suggestions for each page. The suggestions can be seen below:

- The logo at the top of the screen is out of proportions
- Maybe make the login screen a bit bigger, you have enough space. If you set the background on light grey, you can leave the orange border and shade. It should give the login screen a more modern look.
- You could make the login button orange
- You could leave out the orange background
- Align the text of "Login" and "Administrator Login" with the input fields and the login button so it looks vertically aligned.
- One little thing, maybe the orange bar at the top of the application could be the black colour used on our website. It will give a bit more contrast than having everything white and orange. Users will know exactly where to look if you add one more colour to it.
- I personally would get rid of the shade and border of the table.
- You could align the checklist more to the top. It's just randomly floating now for users

- You could give more margin-top to “Add Candidate” and some more spacing between the input fields and a margin-top to the submit button. You have enough space on your screen for it and the fields a little bit less width
- You could add some “line-height” to the text so there is a bit more space in height which makes it more readable
- Try to give it a uniform look & feel. It can really make or break your layout otherwise towards users.

Sprint retrospective

What went well:

During our last daily meeting we reviewed everything that we thought went well during the last sprint and we thought that although we still didn't have some members contributing we managed to make all the changes that the client wanted and implemented the new features in a swift fashion as well as this we also found that working together on some of the tasks was very useful as it allowed us to get a second opinion on design ideas as well as this it also allowed us to solve code issues quickly.

What needs improving:

The major downfall of this sprint was that we only had 4 people that were in contact with us and actually turned up to meetings which meant we had to limit new features we were adding because we didn't have enough time to maybe implement some of the ideas which were discussed in the earlier meetings. I believe that if we had a full team of 7 instead of the reduced 4 then we could have added more functions and have made a more refined product in the end. We were also not able to separate team members and assign them to the report which meant that we had to work longer hours during the day so that we could reach the same standard that other groups have done which left a lot of the team very exhausted.

Requirements analysis and system design

User stories and product backlog

1. As a user I want to be able to view an overview of each candidate which
2. displays their details
3. As a User I want to be able to have a checklist on which I can add items so that I can see what each candidate needs
4. As a user I want there to be consistent styling across the website to make the website more attractive
- 5.

Change styling according to client feedback

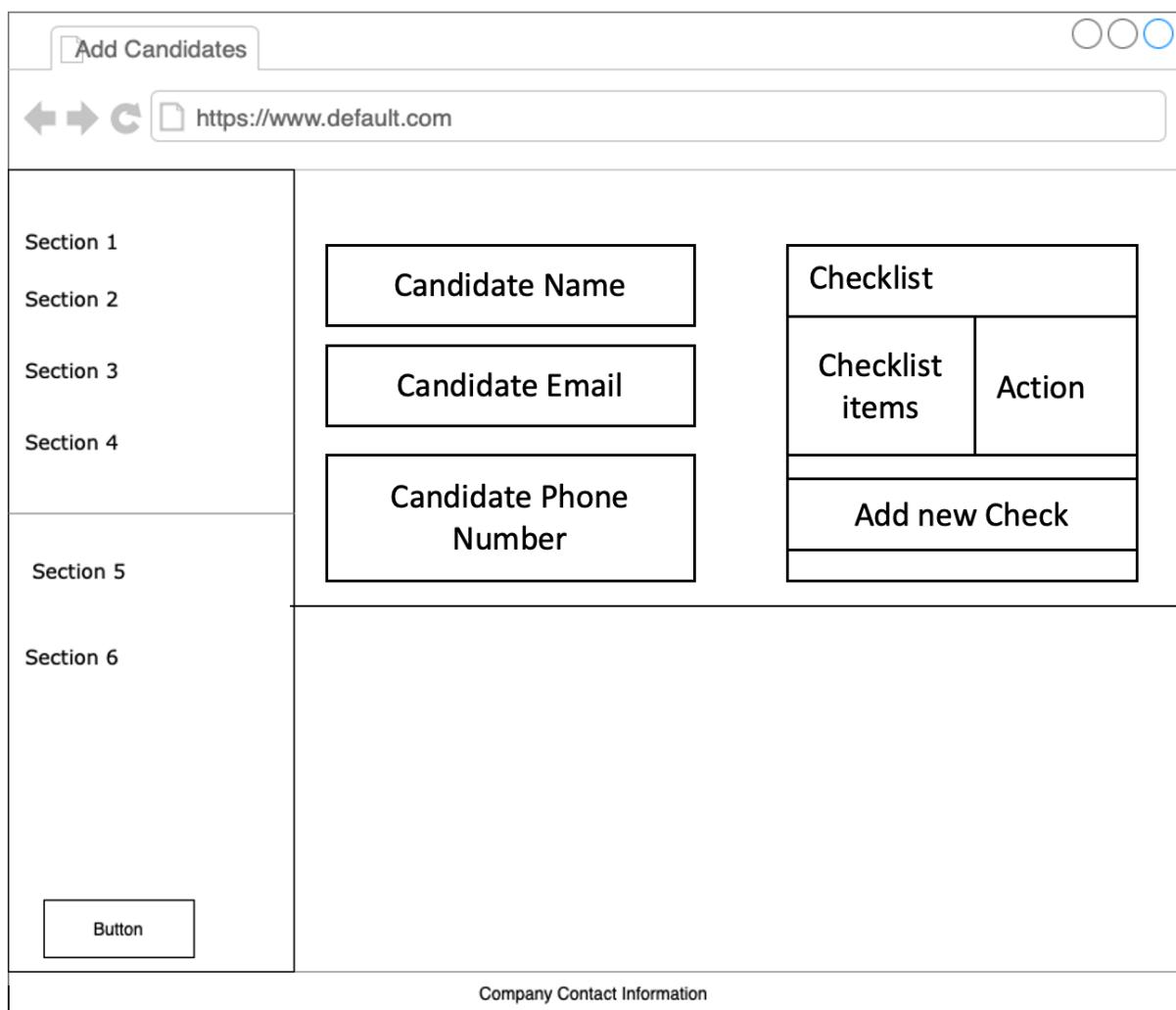
Make individual candidate profiles

Make a checklist for each candidate

Move the To-Do list onto its own page

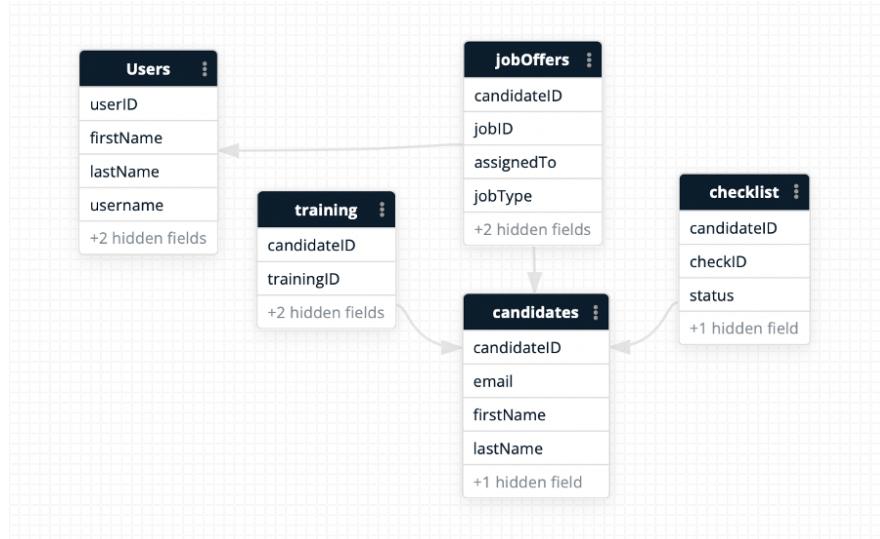
Overall system architecture

System designs such as UML diagrams, database design, etc.



Based on the user stories and feedback we received from the client we decided that we needed to create a page where each individual candidate could be viewed and edited. We decided that this would be a separate page that would display information about each candidate based off the candidate ID which would be stored in the database, this page would be accessible through the candidate list, we decided that it would be best if you could click on the candidate that you wanted to view and then that would bring you to their individual page. On this page you will be able to see all the candidate's information (Name, Email phone number) as well as having access to the candidate's personal checklist at the top of the page.

Column	Type	Null	Default	Comments
checkID (Primary)	int(11)	No		
candidateID	int(11)	No		
detail	varchar(255)	No		
status	int(1)	No		



After deciding to store the values of the checklist in a database we then started to design the database to make sure we include all the relevant values needed.

Candidate ID will link to the candidate table so that the checklist can be individual for each candidate.

Description of the solution implementation

The screenshot shows a web-based application interface. On the left is a sidebar with navigation links: Dashboard, Candidate list, Add Candidate, Job offers, Training, To Do List, SETTINGS, and Admin. The main content area displays three fields: Name (Hewie Town), Email (htown2@amazon.co.jp), and Phone number (1574999969). A red header bar at the top right contains the text 'CheckList' and 'Status', with a 'submit' button. The overall design is clean and modern.

This is the page which was designed first according to the system design. After creating a model of what the page would look like we could then start to implement the actual functionality. To do this we decided that the best way would be to include the candidate's unique ID in the link and then use PHP functions (`$_GET`) to retrieve this data from the link. Then the specific records would be retrieved from the database and displayed to the user.

The screenshot shows the 'Structure' tab of the PhpMyAdmin interface for a table named 'checklist'. The table has four columns: checkID (int(11)), candidateID (int(11)), detail (varchar(255)), and status (int(1)). The 'status' column is defined as AUTO_INCREMENT. Below the table structure, there are buttons for Check all, With selected:, Browse, Change, Drop, Primary, Unique, Index, and Spatial. At the bottom, there are buttons for Print, Propose table structure, Move columns, Normalize, Add, and Go. The 'Indexes' tab is also visible, showing two primary keys: checkID (PRIMARY, BTREE) and candidateID (PRIMARY, BTREE).

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 checkID	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 candidateID	int(11)			No	None			Change Drop More
<input type="checkbox"/>	3 detail	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	4 status	int(1)			No	None			Change Drop More

To implement the database, we used a web program called PhpMyAdmin which gives a visual representation of the sql server and allows you to manage everything with the server. After adding the new table with all the correct fields, we then linked the candidateID to the candidates table which can be seen under the indexes table in the screenshot

Details of the solution testing and evaluation

Test	Expected Result	Actual result	Evidence	Fixes (if needed)						
Individual candidate profile	Displays the candidates information	Displayed the wrong information	<p>Name rguslon0@who.int</p> <p>Email rguslon0@who.int</p> <p>Phone number 7901778439</p>	Simple fix: Error was the email was being displayed in the name section as well as the email section						
Individual candidate profile pt.2	Displays the candidates information	Displays the candidates information	<p>Name Robinetta Guslon</p> <p>Email rguslon0@who.int</p> <p>Phone number 7901778439</p>							
Checklist	Item added to the database	Error when adding	<p>Item could not be added</p> <p>Close</p>	Data was being inserted into the wrong table						
Checklist pt.2	Item added to the database	Item added to the database	<table border="1"> <thead> <tr> <th>CheckList</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>Laptop - Not Done</td> <td>Mark as Done</td> </tr> <tr> <td>laptop</td> <td>submit</td> </tr> </tbody> </table>	CheckList	Status	Laptop - Not Done	Mark as Done	laptop	submit	
CheckList	Status									
Laptop - Not Done	Mark as Done									
laptop	submit									
Checklist pt.3	Mark item as done	Marked item as done	<table border="1"> <thead> <tr> <th>CheckList</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>Laptop - Done</td> <td>Delete</td> </tr> <tr> <td>CheckList item</td> <td>submit</td> </tr> </tbody> </table>	CheckList	Status	Laptop - Done	Delete	CheckList item	submit	
CheckList	Status									
Laptop - Done	Delete									
CheckList item	submit									

Checklist pt.4	Delete item	Delete item	CheckList	Status
			CheckList item	submit

Conclusion

In conclusion our team is very happy with the system we have implemented as we think it includes all of the user requirements which the client required for their onboarding system.

Throughout the constant feedback with the client throughout the process of this development cycle our. Positive thoughts about our creation were reinstated.

This assignment has taught us all time management skills as we were following a strict iterative schedule and knew if we fell behind that schedule then this would have a negative effect on the overall product. We have also developed our communication and self-discipline through attending our daily meetings and working with people who we haven't yet met throughout university.

We appreciated that we faced some disadvantages within this assignment compared to other teams due to members not engaging within this project, however we did not want our overall product to reflect this.

Overall we feel like we have gained first hand experience of an agile approach to software development and feel as though we have developed personal skills in which we can apply to real life situations during university and beyond.

Across all three Sprints we have concluded the 3 main strengths of our team were:

- Efficiency when making the system to the highest standard to make sure we stuck to the strict timescale.
- The quality of work produced due to the amount of research conducted
- Adaptability due to being able to pick up other people's work at the last minute and be able to get working on it straight away.

We believe that all these strengths helped us produce the final product. However we also believe we had 3 main weaknesses which include:

- Communication between Team members
- Lack of engagement from some team members
- Time management of some members as they didn't meet the scrum master deadlines

We overall feel as though if we participated in hack camp again these would be the main factors to work on in the beginning as if these were solved, our team would have been able to produce a much higher quality product which included more features.

Appendices

Appendix 1: Daily stand-up meetings

Date/Time	Attendees	What was Agreed
<u>Monday 10th January</u>	<u>Elise</u> <u>Thomas</u> <u>Akila</u> <u>Muhammad</u> <u>Quincy</u> <u>Sufyan</u> <u>Max</u>	<p>Within this meeting, we decided that Elise would be scrum master.</p> <p>We faced difficulties within this meeting due to all participants turning on mute after 5 minutes, this lead to Elise talking to no one from the remaining 10 minutes, trying to determine everyone's skills.</p>
<u>Tuesday 11th January</u>	<u>Elise</u> <u>Thomas</u> <u>Max</u>	<p>Within this meeting, we agreed on the user requirements of the first sprint and we also agreed on who would design what part.</p> <p>Elise took wireframes.</p> <p>Thomas took database design.</p> <p>Max took graphical design.</p> <p>We then spent the rest of the meeting talking through our ideas of the design to make sure we were all on the same page.</p> <p>Elise also took it upon herself to develop the product backlog and sprint backlog.</p>
<u>Wednesday 12th January</u>	<u>Elise</u> <u>Thomas</u> <u>Max</u> <u>Quincy</u>	<p>Within this meeting, we picked apart our designs and separated tasks amongst the group. Although the rest of the team didn't participate in the meeting, they still took on some tasks to complete.</p> <p>Elise took Jobs offers page</p> <p>Thomas implemented database</p> <p>Max implemented dashboard and layout</p> <p>Quincy implemented candidates page</p>
<u>Thursday 13th January</u>	<u>Elise</u> <u>Thomas</u> <u>Max</u> <u>Quincy</u> <u>Akila</u>	<p>Within this meeting as we were in the implementation phase we just had a catch up to see where everyone was up to and whether anyone needed any help upon their sections.</p> <p>At this point no one required any help and everyone was on track to have their sections completed for the demo on Friday.</p>
<u>Friday 14th January</u>	<u>Elise</u> <u>Thomas</u> <u>Max</u> <u>Quincy</u>	The customer demonstration would take place later today so we decided to meet to make sure everyone was prepared with their sections and so that we could start combining all the separate elements.

		Today we found out that sufyam, Muhammad and akila has produce no work towards the sprint, therefore we had a lot of work to catch up on.
<u>Monday 17th January</u>	<u>Elise</u> <u>Thomas</u> <u>Max</u> <u>Quincy</u>	Today we started our second sprint, therefore to keep our agile approach me started with taking the user requirements and design elements in the morning and then we started to implement it in the afternoon as we had a much shorter sprint during sprint 2. We also reviewed our client feedback with out meeting.
<u>Tuesday 18th January</u>	<u>Elise</u> <u>Thomas</u> <u>Max</u> <u>Quincy</u>	Today was a day devoted towards the implementation of our second sprint. Therefore todays meeting was checking whether anyone needed help with their section and whether we were all on track for the demonstration on Wednesday. Quincy revealed he needed some help so max jumped to help him straight away and managed to keep us on schedule, which has a positive effect on our outcome.
<u>Wednesday 19th January</u>	<u>Elise</u> <u>Thomas</u> <u>Max</u> <u>Quincy</u>	Today was our last day of our second sprint so within the meeting today we assessed how everyone was getting on and if everyone was on track for the demonstration in the afternoon. We then decided at 2pm we would combine all elements of the second sprint before the demonstration. We then started our 3rd sprint in the evening going over the design process.
<u>Thursday 20th January</u>	<u>Elise</u> <u>Thomas</u> <u>Max</u> <u>Quincy</u>	Today was devoted to the implementation process of Sprint 3. Max and Quincy took charge in the overall styling and Elise and Thomas would finish implementing the next pages within the Sprint.
<u>Friday 21th January</u>	<u>Elise</u> <u>Thomas</u> <u>Max</u> <u>Quincy</u>	Today was the last day of our 3rd sprint so in the meeting we check everyone's progress and made sure we were on track for the demo which was happening in the afternoon. We agreed again we would combine all elements at 2pm and request feedback later in the afternoon.

Appendix 2: Final Product

Currently On boarding
12

Total Candidates
20

First Name	Last Name	Email	Phone number
Robinetta	Guslon	rguslon0@who.int	7901778439
Madelene	Whitehall	mwhitehall1@myspace.com	3814662038
Hewie	Town	htown2@amazon.co.jp	1574999969
Solomon	Pringer	springer3@google.cn	3011364033

Candidate list

Person	phoneNumber	email
Robinetta Guslon	7901778439	rguslon0@who.int
Madelene Whitehall	3814662038	mwhitehall1@myspace.com
Hewie Town	1574999969	htown2@amazon.co.jp
Solomon Pringer	3011364033	springer3@google.cn
Theodore Dunsire	2765645240	tdunsire4@yelp.com
Vittorio Rachig	3994215840	vrachig5@wp.com
Baryram Draayer	8169077567	bdraayer6@youtu.be
Barrie Lowle	6124375950	blowle7@baidu.com
Pryce Guitton	8372632178	pguitton8@paypal.com
Amil Boothby	5486032732	aboothby9@mozilla.com
Pietra Hallut	8881265801	phalluta@slashdot.org
Anna-diana Caw	3194042559	acawb@mediafire.com
Norean Rockhill	4002500695	nrockhillc@mozilla.org
Nobie Syde	8446356033	nsyded@pcworld.com
Josephha Harcarse	3629095426	jharcarsee@berkeley.edu

Name
Robinetta Guslon

Email
rguslon0@who.int

Phone number
7901778439

CheckList	Status
CheckList item	submit

Add Candidate

Please enter the following details for the candidate...

First name
Last name
Phone Number
Email Address

Submit Candidate

Job offers

#	Job Title	Number of candidates	Last Updated
1	Software engineer	5	12th January 2021
2	Programmer	2	10th January 2021
3	HR	3	15th January 2021



The screenshot shows a user interface for a 'To Do List' feature. At the top right is a 'Sign out' link. On the left, a sidebar menu includes 'Dashboard', 'Candidate list', 'Add Candidate', 'Job offers', 'Training', and 'To Do List' (which is currently selected). Below the sidebar are 'SETTINGS' and 'Admin' links. The main content area is titled 'To Do List' and contains a text input field with placeholder 'Add your new task' and a red '+' button. Below the input field, a message says 'You have 0 pending tasks' next to a red 'Clear All' button.

The screenshot shows an 'Admin Page'. At the top right is a 'Sign out' link. On the left, a sidebar menu includes 'Dashboard', 'Candidate list', 'Add Candidate', 'Job offers', 'Training', and 'To Do List' (which is currently selected). Below the sidebar are 'SETTINGS' and 'Admin' links. The main content area is titled 'Admin Page' and features a table with one row of data:

#	First	Last	Username	Admin Level
1	Aggie	Tuke	test	1

Below the table is a red 'Delete' button. Underneath the table, there are five input fields labeled 'Username', 'Password', 'First name', 'Last name', and 'Admin level'. At the bottom is a 'Submit' button.