Create an Interactive Guide to Support Primary School Teachers with Implementing the Digital Competency Framework

Author: Elinor Anne Eluned Morris

Student Number: 22040567

CMT403 Project MSc Computing | 60 Credits

Supervisor: Dr Catherine Teehan

Moderator: Dr Rhodri Morris

Acknowledgements

I would like to thank my supervisor for this project, Dr Catherine Teehan, for her unwavering support and encouragement throughout.

Abstract

The Digital Competency Framework has been introduced into primary schools as part of the new Welsh curriculum, but teachers are struggling to understand all of the relevant terminology and find relevant resources to support their teaching. We have been sent a request to build an easy to navigate interactive web-based platform that will explain the Digital Competency Framework in simplified terms, with explanations for key terms with examples of how to teach this in the classroom.

Keywords: Digital Competency Framework, Welsh Curriculum

Table of Contents

Table of Figures

Table of Tables

List of Abbreviations

DCF	Digital Competency Framework
DCT	Data and Computational Thinking
AoLE	Area of Learning and Experience

Introduction

Digital competence has become an essential skill that all young people need to succeed (Lewis, H. 2015). In Wales, the Welsh Government has recognised the importance of digital literacy in education as a fundamental component of the new Welsh curriculum, through the Digital Competency Framework (DCF) (Welsh Government 2022). This framework elevates digital literacy to the same level of importance as traditional literacy and numeracy skills (Welsh Government, 2020).

The new Welsh Curriculum 2022 has given teachers unprecedented autonomy in designing their lesson plans. The DCF is composed of four interconnected strands that span across all areas of the curriculum, fostering a holistic approach to digital literacy. Digital competency is “no longer confined to ICT classes” (Welsh Government 2018a) but is now a cross-curricular responsibility, requiring teachers to integrate digital skills into a broad range of subject areas, to ensure that students develop the digital literacy necessary to thrive in a rapidly developing digital world.

However, the effective implementation of the DCF is heavily dependent on teachers’ understanding and ability to integrate its elements into their lesson planning. This presents a myriad of challenges for teachers, such as insufficient training, limited knowledge, low confidence, inadequate resources, and resistance to change (Carroll, F. *et al.* 2023) which could in turn adversely affect students’ performance and skills development. Further to this, technology is constantly changing in education. The influx of new hardware and software tools can leave some teachers with negative attitudes towards the DCF, who believe that too many new initiatives are brought in with insufficient time to prepare (Estyn 2018). Primary school educators, who typically teach across multiple subjects without a specialised focus, may find it particularly difficult to adapt to the demands of digital literacy, as they may lack familiarity with STEM related topics and terminology. Further to this, many current educators may not have received these crucial digital skills during their own education, as they went through the old curriculum where there was no mandatory digital literacy. Consequently, many educators may have missed the chance to develop essential digital skills, especially if their own education and training neglected ICT and computing.

There is a gap to be addressed between the existing digital literacy levels of primary school teachers in Wales and the expectations of digital competency in the new curriculum. With digital competency now a cross-curricular responsibility, it is essential that teachers can integrate these skills effectively.

It is important to acknowledge the disparities in digital literacy levels among teachers, influenced by factors like age, experience, and access to professional development. It can be assumed that newly qualified teachers are likely to possess stronger digital skills and confidence compared to those who entered the profession decades ago. Some teachers may feel that they lack adequate professional development or training in digital literacy. As a result, this can lead to reliance on Digital Leads or outsourced organisations to bridge these gaps. The pressure to integrate new hardware and technologies without sufficient support or training can further undermine their confidence and ability to effectively utilise digital tools in the classroom. Despite these challenges, teachers are inherently innovative practitioners who are committed to life-long professional learning.

This dissertation project aims to address this critical issue by creating an interactive web guide specifically designed for primary school educators. The guide will enhance educators' confidence and digital literacy, equipping them with the knowledge and skills necessary to apply technology effectively in the classroom. This project will document the guide's development, including the research, design, implementation, and evaluation phases, ensuring alignment with the project's objectives.

Project Aim and Objectives

The overarching aim of this dissertation project is to develop an interactive, user-friendly, web-based guide, that simplifies the DCF, provides resources and ideas for lesson plans, to support primary school teachers in Wales, in effectively implementing the DCF into their lesson planning. As a result, it is expected that this will also enhance the digital competence and confidence of primary school teachers in delivering DCF-aligned content.

To achieve this aim, the following measurable objectives need to be met:

- Conduct a comprehensive review of the DCF and analyse the challenges it presents to educators.

- Gather user requirements to inform the web guide’s content.
- Design and develop an interactive web guide that:
  - Is user-friendly, accessible, intuitive and easy to navigate
  - Incorporates interactive features that enhance user engagement, saves time and provides tailored lesson planning support
  - Contains relevant teaching resources, activities and lesson plans aligned with the DCF, available from platforms commonly used by educators use such as the Hwb and Twinkl
  - Simplifies complex and unfamiliar DCF terminology with clear definitions and relatable examples to increase understanding and implementation of key concepts
  - Translates the DCF into practical classroom activities
- Evaluate the project against the DCF criteria as a marker of success.

Background

This chapter provides the necessary background for understanding the context of this project. I will present research highlighting the need for digital competence in the Welsh curriculum and examine how the DCF addresses this need. This will involve a brief overview of the DCF, exploring the challenges that teachers face in its implementation, and an analysis of successful case studies. Additionally, I will assess existing tools, resources and support available to teachers. The DCF will be critically analysed to understand the areas that teachers struggle with the most. Existing resources will be analysed to understand where the gaps are, as well as understanding the support that is currently available to teachers. I will look at Estyn reports, which provide valuable insights into how primary schools have approached the challenge of implementing the DCF.

The Need for Digital Competency in the Curriculum

Digital literacy is a fundamental skill for success in the modern workplace (Nania, J. 2019). The Welsh Government has recognised the importance of digital literacy, making it an essential part of the curriculum through the DCF. The DCF outlines the essential digital skills that all learners should acquire throughout their educational journey. Digital competence joins literacy and numeracy as one of the foundations for success in learning and life (Estyn 2019). Primary schools play a crucial role in not only developing students' literacy and numeracy skills but also in fostering digital competence from an early age. The DCF aims to equip learners with the essential skills to thrive in a digital world, emphasising that digital literacy is no longer optional but essential for all students (Nania, J. 2019).

The Curriculum for Wales: Context and Reform

The Welsh curriculum has undergone a significant transformation in recent years. The previous curriculum, introduced in 1988, was designed in a pre-digital era and failed to consider technological advancements. How pupils in Wales had been taught had remained largely unchanged since then. The Welsh Government requested Professor Graham Donaldson to conduct an independent review of the curriculum, resulting in the *Successful Futures* report (Donaldson, 2015). This report highlighted the need for significant curriculum reform, including:

- A streamlined curriculum with only six areas of learning, that combines core and non-core subjects.
- A more flexible approach to learning progression, replacing key stages with five progression steps, at three-year intervals, for a seamless learning experience.
- Digital competence should be introduced as a third cross-curricular framework, alongside literacy and numeracy (pg. 3 Welsh Government, 2015).

In response to these recommendations, the Curriculum for Wales 2022 was introduced, where digital competency has been given equivalent importance in legislation (Curriculum and Assessment (Wales) Act 2021) as literacy and numeracy, through the DCF. The DCF recognises that digital literacy is no longer an optional skill but an essential one to equip students with the digital skills demanded by employers (BBC 2015). Digital competency is not merely another subject, but a foundational skill that permeates all areas of learning. Embedding digital competency in school life is more than just using technology in isolated lessons, but about cultivating a digitally literate community where students are equipped and empowered to thrive in the digital age.

Following Donaldsons recommendations, the new Welsh Curriculum is made up of five progression steps and includes all six AoLE's, providing a framework for schools to design their curriculum. Schools and practitioners must use the statements of what matters to guide the development of curriculum content, ensuring that all aspects of the curriculum are covered. Digital competency has been introduced as a cross-curricular framework.

Overview of the Digital Competency Framework

The DCF is designed to provide a structured approach to developing digital skills across four key high-level strands. These key strands are split into elements, and these elements have skill statements which describes the expected skills and competence a learner should be able to demonstrate.

The DCF outlines four areas of digital competence:

Strands and elements:

I. Citizenship

- Identity, image and reputation
- Health and well-being
- Digital rights, licensing and ownership
- Online behaviour and cyberbullying

The citizenship strand helps learners develop the skills to engage positively in the digital world. Learners become responsible consumers and creators of digital content through critical thinking.

II. Interacting and Collaborating

- Communication
- Collaboration
- Storing and sharing

Learners will explore different methods of electronic communication and learn how to store and share data appropriately.

III. Producing

- Planning, sourcing and searching.
- Creating
- Evaluating and improving

Learners will work with a range of multimedia like text, images, audio and video to create digital content for specific purposes.

IV. Data and Computational Thinking

- Problem-solving and modelling
- Data and information literacy

This strand teaches learners to understand data collection, representation, and analysis (Welsh Government 2024).

While all strands of the DCF are equally important, I have narrowed the focus of this project to the Data and Computational Thinking (DCT) strand, which is essential for developing critical thinking skills, data literacy, and an understanding of algorithms. I have narrowed the scope of this project for several reasons.

Firstly, due to time limitations of this project, it is more favourable to do one part of the DCF thoroughly as opposed to touching upon lightly each of the four strands. Further to this, of the four strands of the DCF, the DCT strand is the most relevant to computing and technology. Furthermore, this strand posed the most challenges for educators as evidenced by the Estyn 2018 report, which highlights that teachers feel less confident in teaching computational thinking compared to other strands:

> *"11 Teachers say that they are more confident with the content strands of 'citizenship', 'interacting and collaborating' and 'producing', than with 'data' and 'computational thinking'. This broadly reflects many teachers' knowledge and understanding of the ICT curriculum content." (Estyn 2018).*

Focus and Scope of this Project: The Data and Computational Thinking Strand

The Data and Computational Thinking strand is divided into two elements:

1. Problem-Solving and Modelling

This element involves computational thinking techniques such as decomposition, pattern recognition, algorithmic thinking and abstraction.

2. Data and Information Literacy

This element focuses on data collection, analysis, interpretation and the various formats that data can be represented and visualised.

The DCT strand emphasises problem-solving, data literacy and the development of algorithms and models. Through this strand, learners will develop a strong foundation in computational thinking, a set of skills that combines scientific enquiry, problem-solving and critical thinking. This skill set will enable learners to use technology effectively. These concepts may be unfamiliar to many teachers who may have received their training before advancements in digital literacy. To address this, the project will provide resources that are accessible and practical, with activities and examples, to support teachers in delivering these competencies. This project specifically targets progression steps two and three (years 3 to 6) of the DCF, providing primary school educators with an accessible resource to effectively teach skills specified in these progression steps

While digital competence is intended to be integrated across the entire curriculum, it is essential to build a strong foundation in core computational thinking skills first. This project will draw inspiration from all Areas of Learning Experience (AoLEs), particularly Science and Technology. By improving and developing educators' digital skills, this will better enable educators in Wales to apply these skills across the curriculum and all subject areas. The focus is on building core competencies before cross-curricular application. Although digital competency is designed to be embedded across the entire curriculum, concentrating on core skills first will better equip teachers to apply their technical knowledge across all of the learning areas. Every statement of learning from every AoLE will be analysed, to influence activities and lesson examples that will be contained in the guide.

Table 1: Extract from Statement of Learning from Science and Technology

Computation is the foundation for our digital World	
Progression Step Two	**Progression Step Three**
I can safely use a range of tools, materials and equipment to construct for a variety of reasons. I can use computational thinking techniques, through unplugged or offline activities. I can create simple *algorithms* and am beginning to explain errors.	I can use conditional statements to add control and decision-making to *algorithms*. I can identify repeating patterns and use loops to make my *algorithms* more concise. I can explain and debug *algorithms*. I can use sensors and actuators in systems that gather and process data about the systems' environment.

I can follow algorithms to determine their purpose and predict outcomes. I am beginning to explain the importance of accurate and reliable data to ensure a desired outcome. I can follow instructions to build and control a *physical device*.	I can identify positive and negative design elements that affect user interactions. I can explain how digital devices can be interconnected locally and globally. I can explain the importance of securing the technology I use and protecting the integrity of my data. I can explain how my data is used by services, which can help me make more informed decisions when using technology. I can explain how data is stored and processed. I can effectively store and manipulate data to produce and give a visual form to useful information.

This extract from the Science and Technology AoLE demonstrates some of the skill statements set out in the DCT strand.

Academic Justification:

Recent research supports this focused approach:

> *“The data and computational thinking element highlight the importance of computational thinking, which combines scientific inquiry, problem-solving, and thinking skills. The key elements are problem-solving and modelling, as well as data and*

> *information literacy. These enhance the learner's ability to create and refine algorithms and flowcharts to solve problems. Furthermore, this supports the construction, refinement, and interrogation of datasets within tables, charts, spreadsheets, and databases to test or support an investigation." (Carroll, F. et al. 2023) ("The Journey to Making 'Digital Technology' Education a Community Learning Venture").*

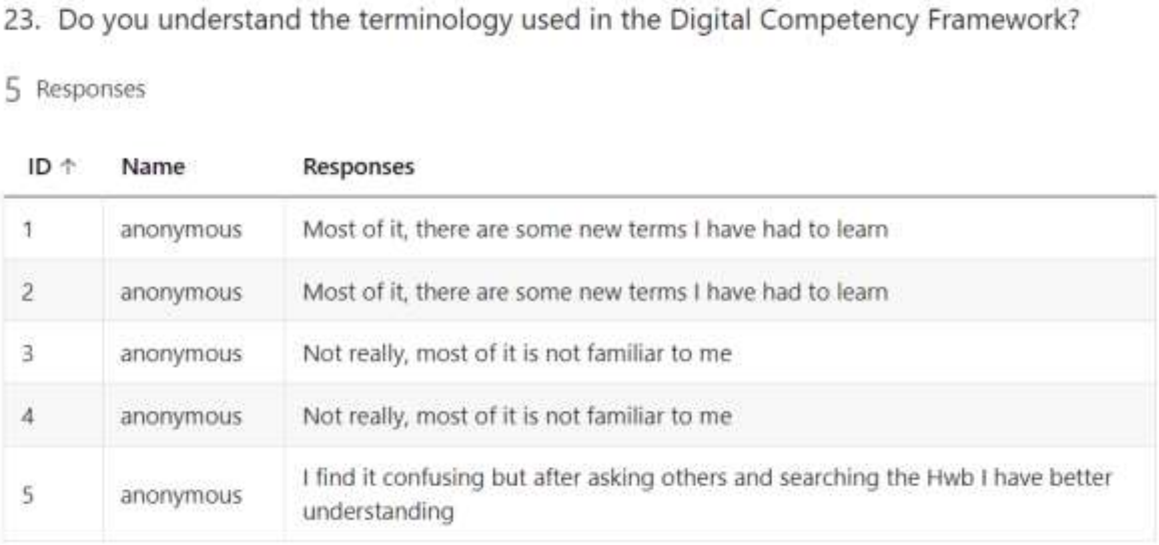
This structured approach will enable teachers to develop core competencies that will provide a foundation for cross-curricular application, so that they can better facilitate effective knowledge transfer to learners.

Challenges Faced by Educators in Implementing the DCF

Research identified several significant challenges faced by primary school educators in implementing the DCF. These challenges can be categorised into three main areas: technical and conceptual barriers, practical implementation constraints, and professional development limitations.

1. Technical and Conceptual Challenges

Primary research conducted through a survey among primary school educators revealed that educators lack confidence in their own understanding of computational thinking concepts and terms, such as abstraction and decomposition. The DCF is presented as statements with no examples or explanations, making it difficult to teach these computational thinking concepts effectively, with no clear understanding of the DCFs expectations and practical application.

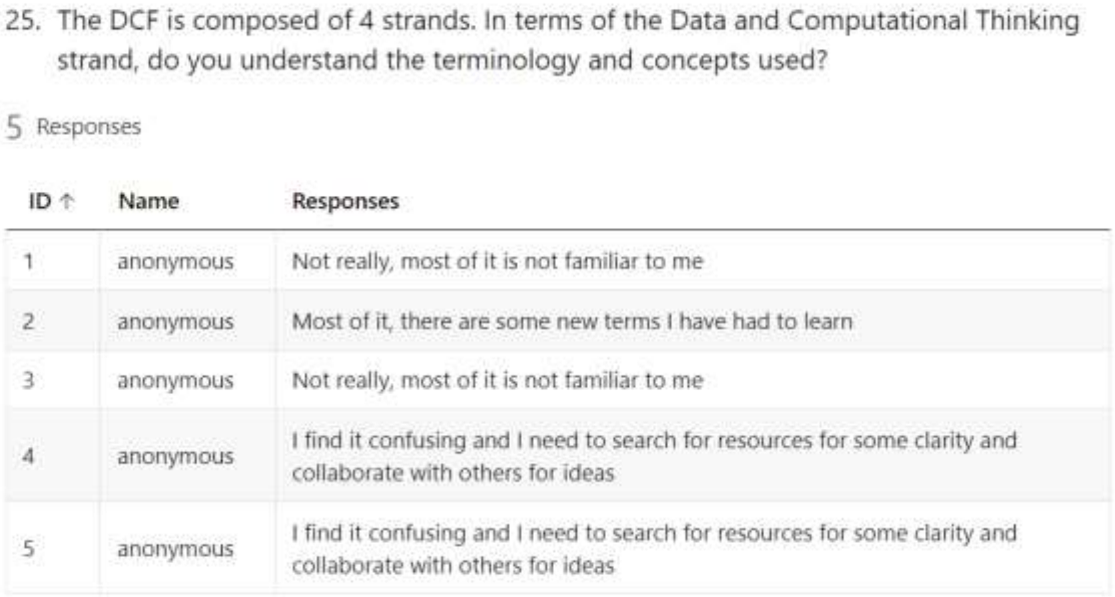


23. Do you understand the terminology used in the Digital Competency Framework?

5 Responses

ID ↑	Name	Responses
1	anonymous	Most of it, there are some new terms I have had to learn
2	anonymous	Most of it, there are some new terms I have had to learn
3	anonymous	Not really, most of it is not familiar to me
4	anonymous	Not really, most of it is not familiar to me
5	anonymous	I find it confusing but after asking others and searching the Hwb I have better understanding

Figure 1: Survey responses to DCF terminology

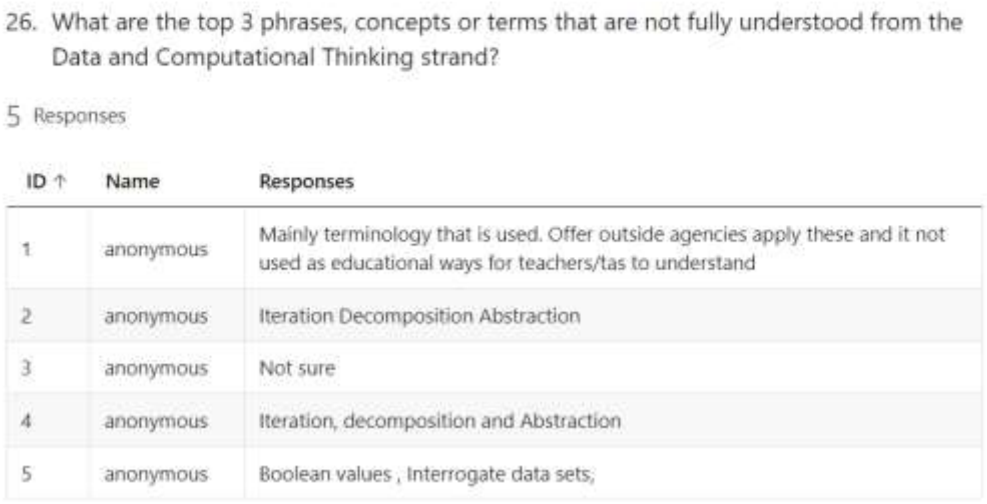


25. The DCF is composed of 4 strands. In terms of the Data and Computational Thinking strand, do you understand the terminology and concepts used?

5 Responses

ID ↑	Name	Responses
1	anonymous	Not really, most of it is not familiar to me
2	anonymous	Most of it, there are some new terms I have had to learn
3	anonymous	Not really, most of it is not familiar to me
4	anonymous	I find it confusing and I need to search for resources for some clarity and collaborate with others for ideas
5	anonymous	I find it confusing and I need to search for resources for some clarity and collaborate with others for ideas

Figure 2: Survey responses to understanding of DCT concepts

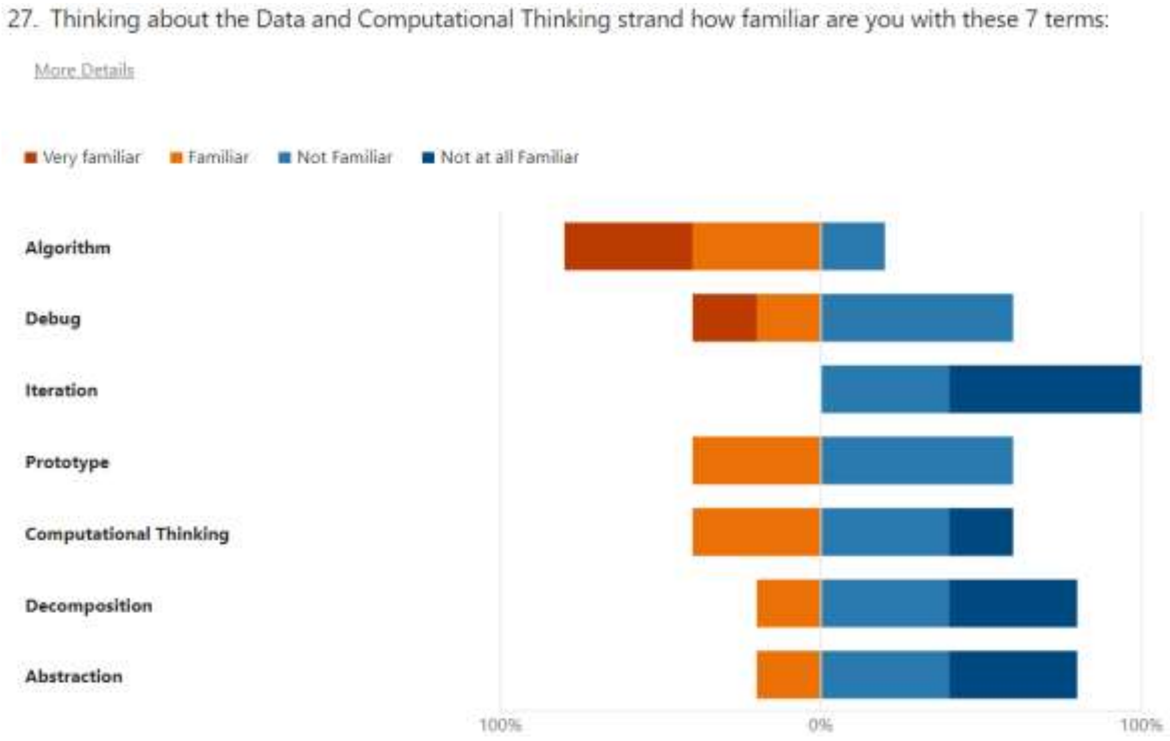


26. What are the top 3 phrases, concepts or terms that are not fully understood from the Data and Computational Thinking strand?

5 Responses

ID ↑	Name	Responses
1	anonymous	Mainly terminology that is used. Offer outside agencies apply these and it not used as educational ways for teachers/tas to understand
2	anonymous	Iteration Decomposition Abstraction
3	anonymous	Not sure
4	anonymous	Iteration, decomposition and Abstraction
5	anonymous	Boolean values , Interrogate data sets,

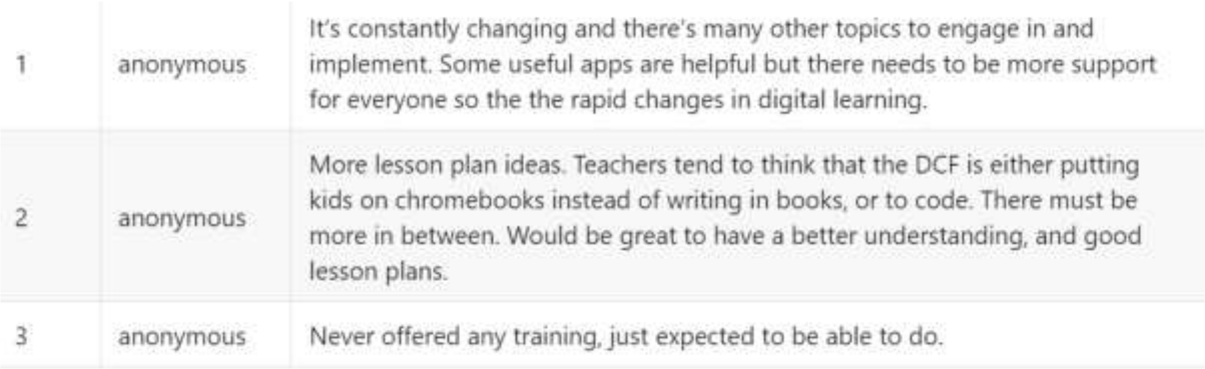
Figure 3: Survey responses top 3 phrases or terms not understood



27. Thinking about the Data and Computational Thinking strand how familiar are you with these 7 terms:

Figure 4: Survey responses to DCT terminology

Survey respondent [1] emphasised the challenges in keeping pace with rapid technological changes, and continuous updates in digital tools and methodologies create ongoing adaptation challenges. Survey respondent [2] highlighted the difficulty in translating abstract concepts into practical classroom activities, with more substance than just using a chrome book. This shows that the integration of meaningful computational thinking activities beyond basic device usage remains challenging.



1	anonymous	It's constantly changing and there's many other topics to engage in and implement. Some useful apps are helpful but there needs to be more support for everyone so the the rapid changes in digital learning.
2	anonymous	More lesson plan ideas. Teachers tend to think that the DCF is either putting kids on chromebooks instead of writing in books, or to code. There must be more in between. Would be great to have a better understanding, and good lesson plans.
3	anonymous	Never offered any training, just expected to be able to do.

Figure 5: Survey responses to challenges faced implementing DCF

2. Practical Implementation Challenges

Survey results showed a variation in access to hardware. While all respondents use basic devices such as tablets, not all respondents have access to devices that are promoted by the Welsh Government and on the Hwb website such as Micro:bit. This demonstrates disparity in access to equipment, and even if new equipment is introduced and given to schools, training or instruction will be required.

Further to this, educators often face time constraints in developing their digital skills to master the available hardware and software, as well as limited time to plan lessons that fully integrate digital competency across all subjects.

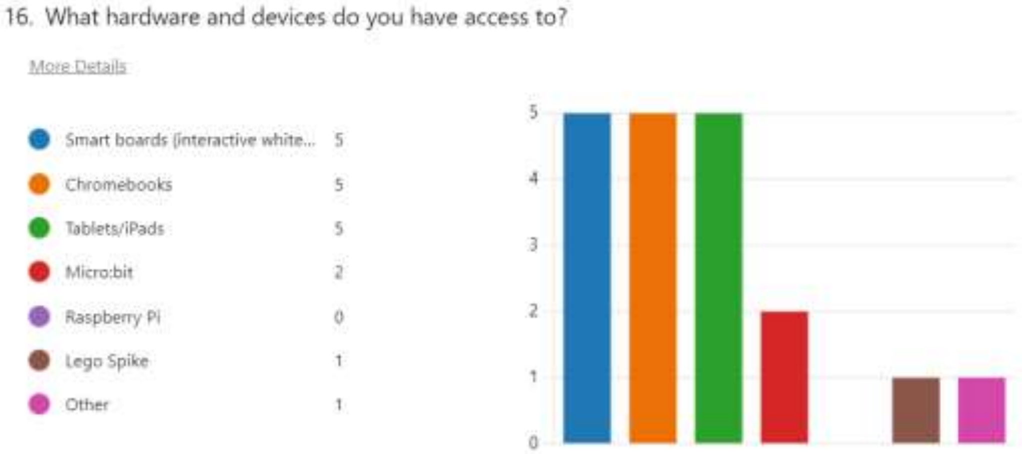
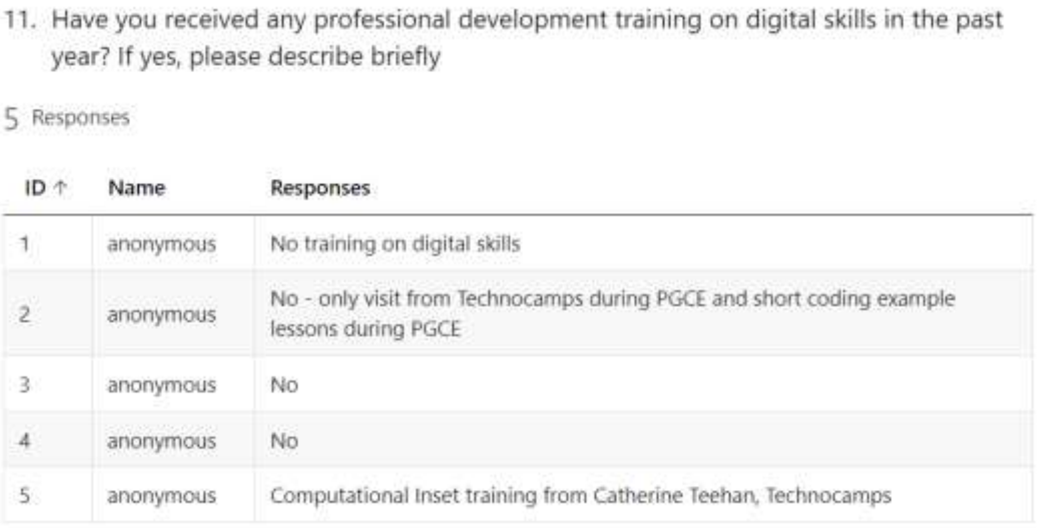


Figure 6: Survey results for hardware accessibility

3. Professional Development Challenges

Survey responses clearly demonstrate that there is a lack of adequate training and professional development opportunities tailored to the DCF. The absence of uniform professional development frameworks may lead to inconsistent digital skills among teaching staff that can hinder effective implementation. With limited access to development opportunities, there is not sufficient technical training being provided.

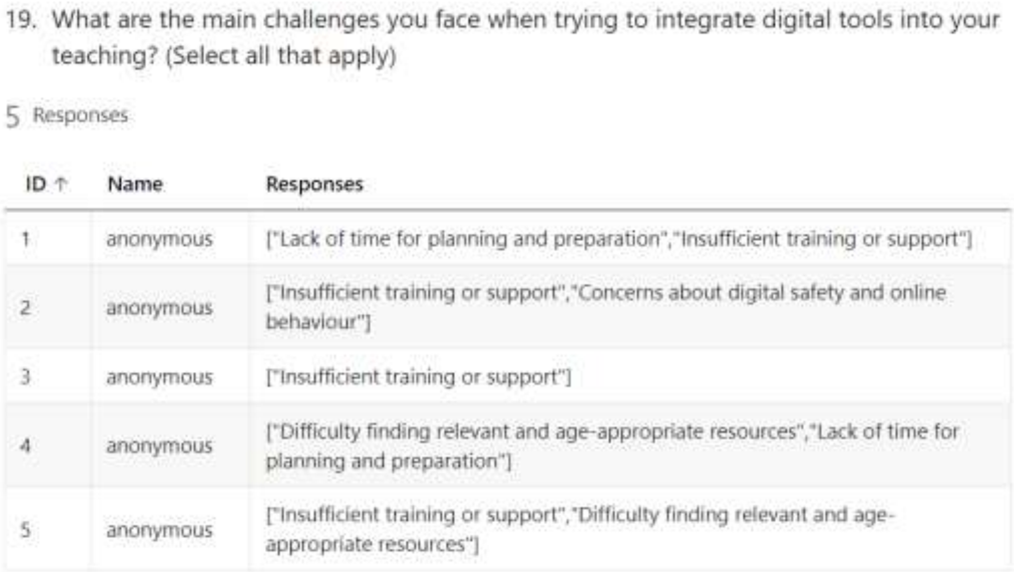


11. Have you received any professional development training on digital skills in the past year? If yes, please describe briefly

5 Responses

ID ↑	Name	Responses
1	anonymous	No training on digital skills
2	anonymous	No - only visit from Technocamps during PGCE and short coding example lessons during PGCE
3	anonymous	No
4	anonymous	No
5	anonymous	Computational Inset training from Catherine Teehan, Technocamps

Figure 7: Survey responses regarding recent professional development



19. What are the main challenges you face when trying to integrate digital tools into your teaching? (Select all that apply)

5 Responses

ID ↑	Name	Responses
1	anonymous	["Lack of time for planning and preparation","Insufficient training or support"]
2	anonymous	["Insufficient training or support","Concerns about digital safety and online behaviour"]
3	anonymous	["Insufficient training or support"]
4	anonymous	["Difficulty finding relevant and age-appropriate resources","Lack of time for planning and preparation"]
5	anonymous	["Insufficient training or support","Difficulty finding relevant and age-appropriate resources"]

Figure 8: Survey responses regarding challenges integrating digital tools

Impact on Implementation

These challenges collectively hinder the effective implementation of the DCF. Inconsistent delivery of the DCF across schools can lead to disparities in student learning outcomes, potentially leaving gaps in their digital literacy and computational thinking skills.

Support and Resources Available for DCF Implementation

This section analyses the primary digital resources available to Welsh educators to implement the DCF and evaluates their effectiveness while identifying areas for improvement.

- **Hwb Digital Learning Platform**

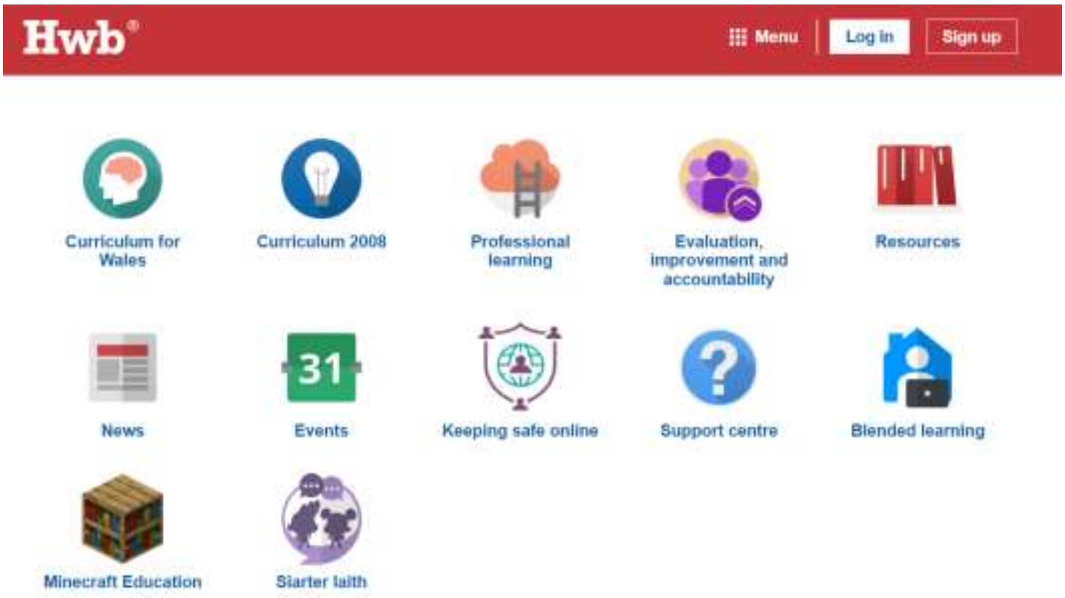


Figure 9: Landing Page for Hwb

Hwb is the Welsh Government's national digital learning platform. It provides schools, teachers, students, and parents, with resources to support the application of the DCF.

Key Features and Analysis

Hwb has a repository that contains an extensive resource section for lesson plan ideas and activities. Hwb sources these from various publishers, including BBC Bitesize, Code Club, Barefoot Computing, Technocamps and Twinkl.

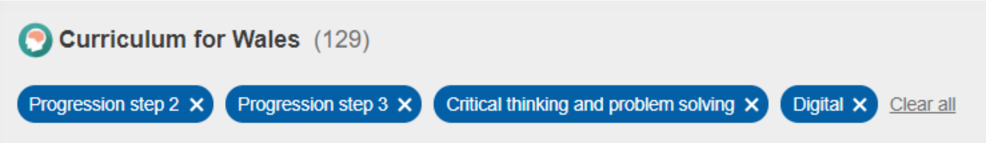
Strengths

The resources section has metrics tracking, including how many views per resource, which can indicate the popularity of a resource and possibly highlighting an area where teachers need guidance. There is significant user engagement with the highest viewed resource having 62,989 views.

There is a section to leave feedback, titled ‘reviews,’ although I had not seen a single review left indicating that no one had interacted with this feature on the resources I viewed, suggesting that this feature needs to be made more accessible.

The resources are available in multiple formats such as editable documents, PDFs, PowerPoints, and videos. There were many Technocamps resources available on Hwb. The resources can be sorted in a variety of orders, based on the date, highest rated or most viewed. There are advanced filtering capabilities for AoLE, progression step, audience targeting, resource type, and skills and themes.

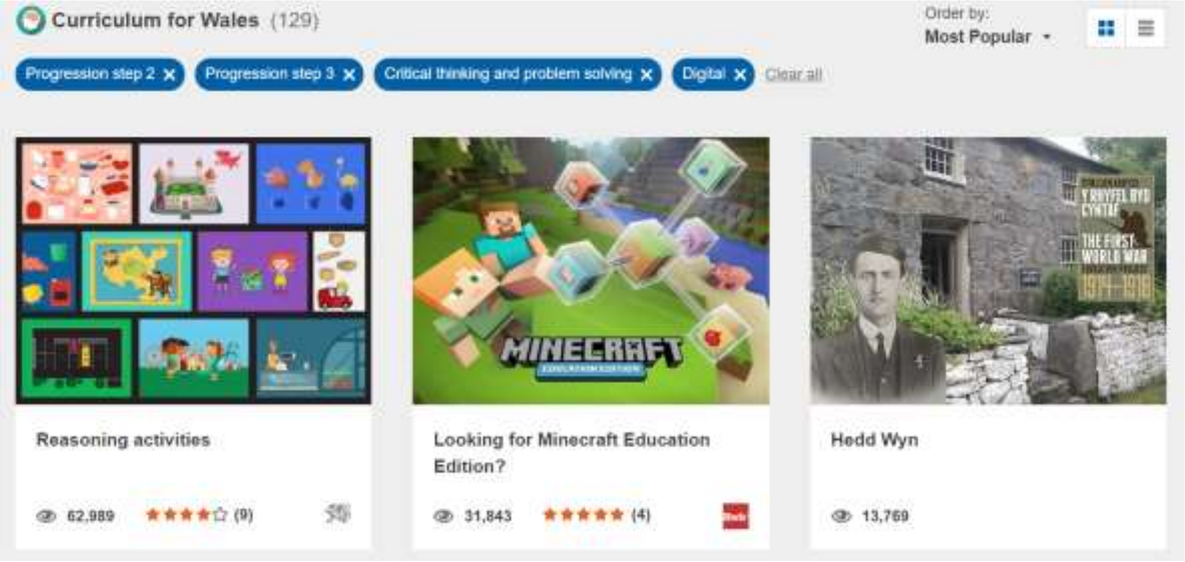
When I filtered the resources to show progression step 2 and 3, for critical thinking and problem-solving, and digital filter, 129 resources were displayed.



Curriculum for Wales (129)

Progression step 2 ✕ Progression step 3 ✕ Critical thinking and problem solving ✕ Digital ✕ Clear all

Figure 10: Search results displayed for Critical Thinking and Problem-Solving on Hwb



Curriculum for Wales (129)

Order by: Most Popular

Progression step 2 ✕ Progression step 3 ✕ Critical thinking and problem solving ✕ Digital ✕ Clear all

Reasoning activities

62,989 ★★★★☆ (9)

Looking for Minecraft Education Edition?

31,843 ★★★★★ (4)

Hedd Wyn

13,769

Figure 11: Results sorted by most popular resources on Hwb

Limitations:

There is limited user interaction with review features, and there is a basic feedback mechanism present, through star ratings only. This lack of qualitative feedback options could be improved with a text area to leave more substantial feedback.

- ### Technocamps Initiative

Technocamps is an outreach organisation funded partly by the Welsh Government. Among the work that Technocamps does, Technocamps facilitate and deliver coding and computational workshops in schools across Wales. Technocamps has a comprehensive website, offers a variety of workshops, as well as extensive CPD courses for teachers.

School Engagement Workshops

There has been significant growth in school engagement since the implementation of the DCF:

Key Statistics:

School engagement growth:

- 2021: 134 schools
- 2023: 596 schools
- 328% increase over two years

Figures from Technocamps annual reports show that in 2021 there were 134 schools engaged, in 2023 this number rose to 596 schools – an increase of 328, more than triple in 2 years (Technocamps 2024a). This could suggest an increase in demand since the implementation of the DCF, which could lead to the assumption that schools are outsourcing lessons to fill the digital skills gap that their staff have. Technocamps also provides training and professional development for teachers.

Technocamps Website is user-friendly and easy to navigate, with repeating characters and colour scheme to create familiarity.

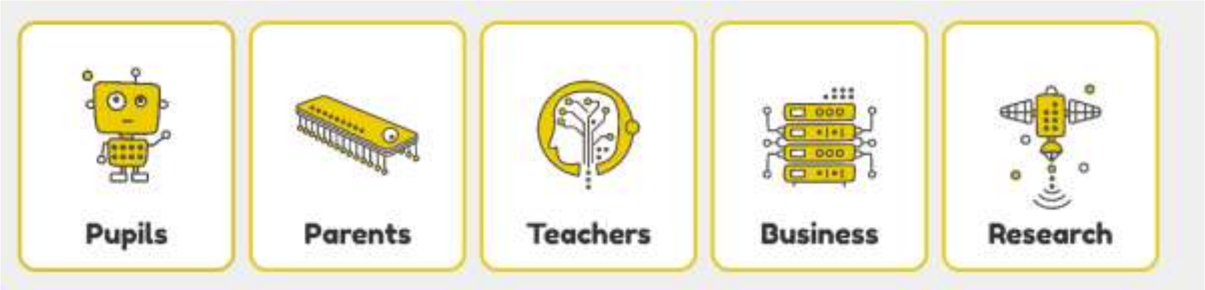


Figure 12: Technocamps banner with navigation links

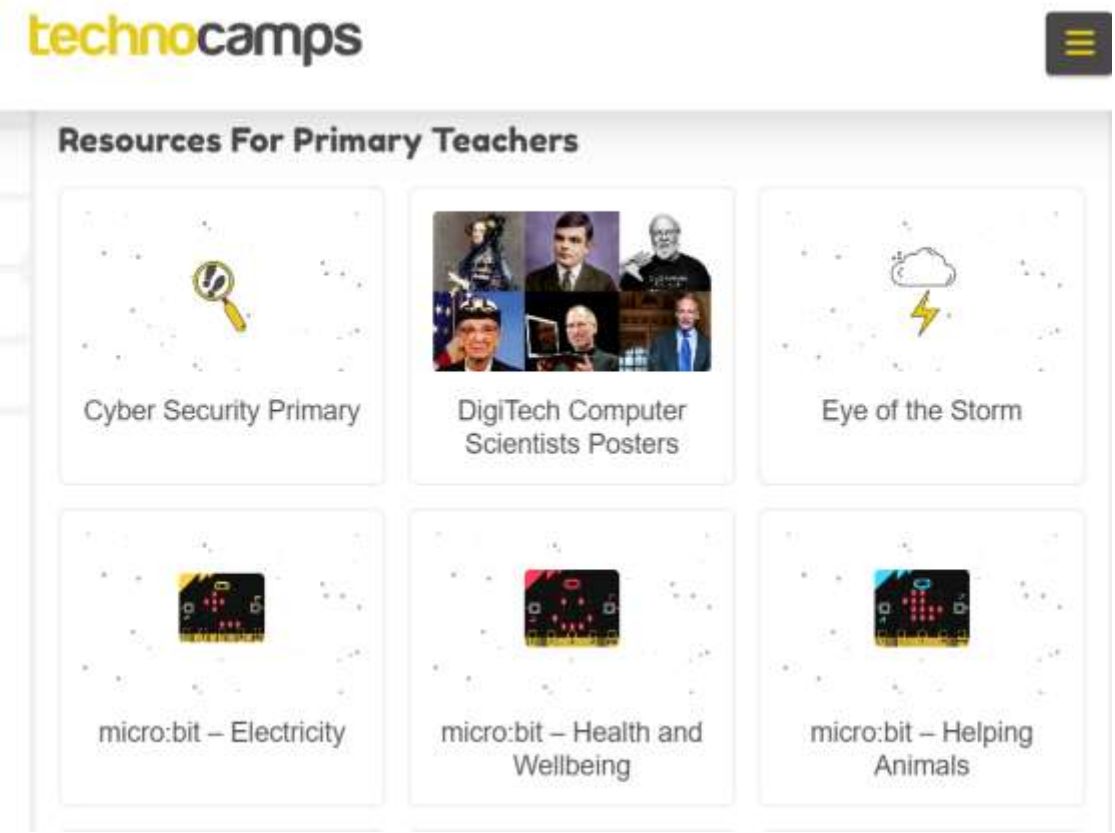

technocamps
Resources For Primary Teachers
Cyber Security Primary
DigiTech Computer Scientists Posters
Eye of the Storm
micro:bit – Electricity
micro:bit – Health and Wellbeing
micro:bit – Helping Animals

Figure 13: Technocamps resources page for primary teachers

An effective feature was clicking on a question and being redirected to a YouTube video, thus incorporating different interactive formats for user engagement.

Figure 14: Question that links to YouTube video

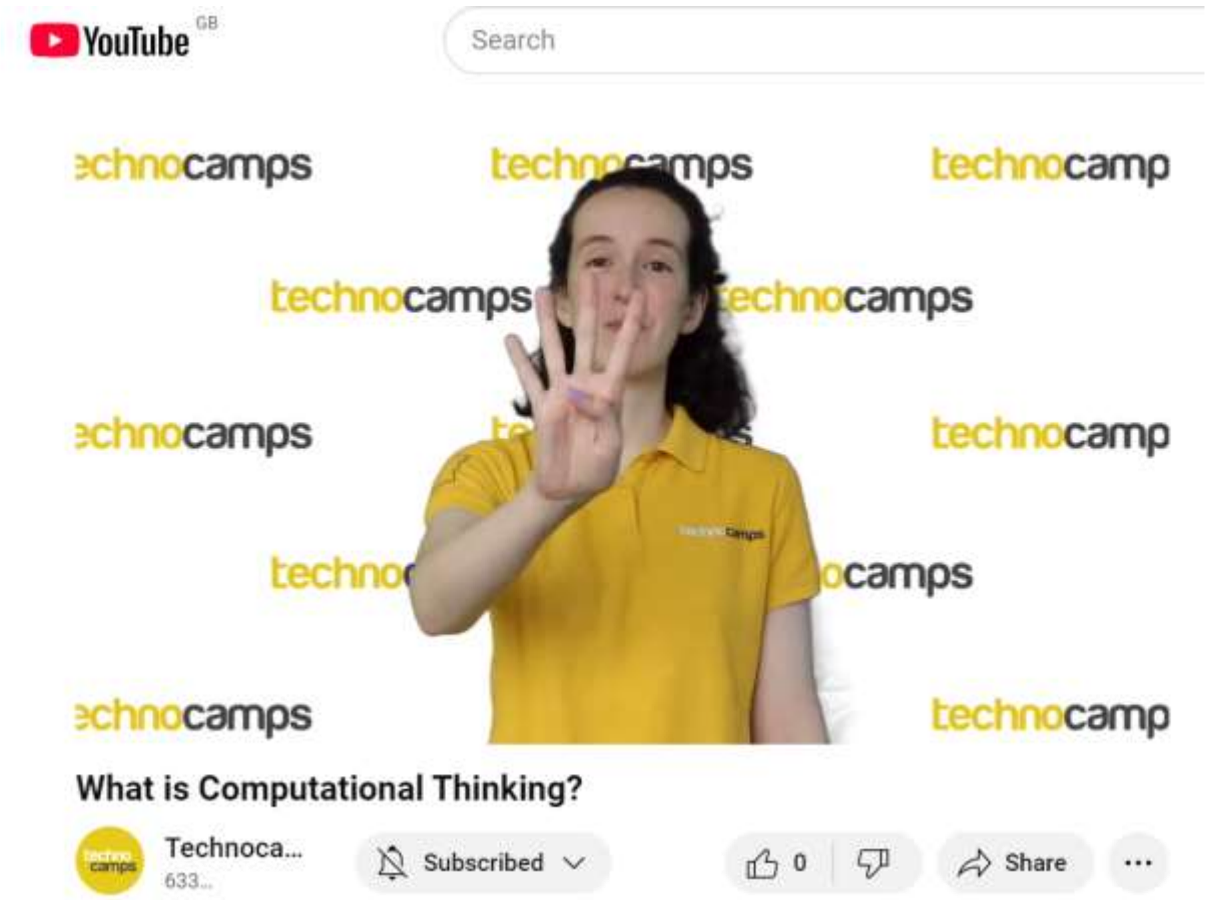


Figure 15: Screenshot of redirected YouTube video

Notable Features:

- User-friendly interface with consistent branding.
- Multimedia approach to content delivery.
- Integration with YouTube for video tutorials.
- Separate resource sections for teachers and students.
- Professional development opportunities.

CPD - Technoteach qualification

Technocamps offers a free accredited CPD opportunity for teachers across Wales, supported by the Welsh Government and the National Science Academy, designed to help educators develop the skills and confidence to teach Computer Science and digital learning in their classrooms, as well as support the introduction of the DCF (Technocamps 2024b).

- **Barefoot Computing**

Barefoot Computing is a platform developed by teachers, offering specialised computing resources.

Strengths

- Free access to resources, although registering an account is required.
- Clean, well-structured layout and presentation with age-appropriate content organisation. Easy to navigate and access, a wealth of resources in an organised manner.
- Barefoot uses tabs to navigate different ages, for seamless interactivity.

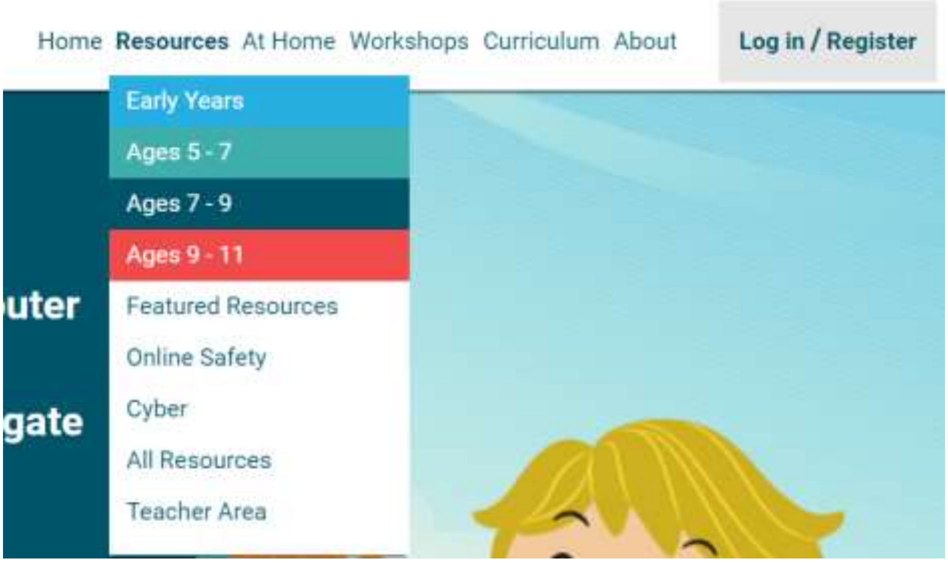
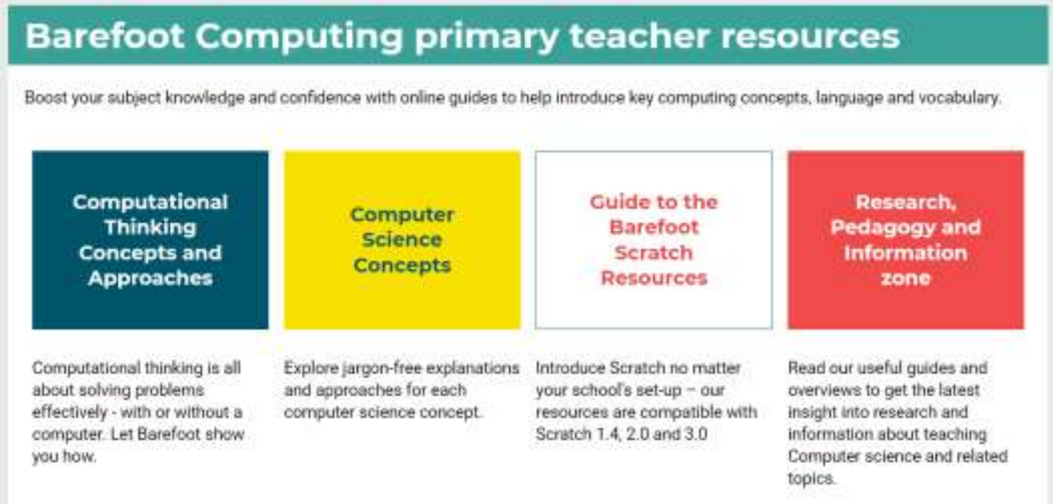


Figure 16: Navbar on Barefoot Computing website



Barefoot Computing primary teacher resources

Boost your subject knowledge and confidence with online guides to help introduce key computing concepts, language and vocabulary.

Computational Thinking Concepts and Approaches

Computational thinking is all about solving problems effectively - with or without a computer. Let Barefoot show you how.

Computer Science Concepts

Explore jargon-free explanations and approaches for each computer science concept.

Guide to the Barefoot Scratch Resources

Introduce Scratch no matter your school's set-up – our resources are compatible with Scratch 1.4, 2.0 and 3.0

Research, Pedagogy and Information zone

Read our useful guides and overviews to get the latest insight into research and information about teaching Computer science and related topics.

Figure 17: Barefoot Computing resources for primary teachers

- **Twinkl Educational Resources**

Twinkl provides lesson plans and activities aligned with the DCF, although many of these resources tend to repeat the DCFs content without offering clear explanations or practical examples for teachers to follow. The layout on Twinkl's website is cluttered and messy, with an excess of downloadable PDFs that can make organising work more difficult, especially if a user simply wants to view the resource. This can be overwhelming for users who are looking for quick and actionable content. User registration is required, with limited free access, a free trial for 30 days is offered, and then requires payment.

Case Studies from Estyn Reports: Successful DCF Implementation

Estyn is an independent body responsible for inspecting and reporting on the quality and standards of education provided in Wales. In addition to identifying areas where improvement is needed, Estyn also highlights area of good practice. Estyn reports provide valuable insights into how schools are effectively implementing the DCF.

St Patrick's RC Primary School

St Patrick's RC Primary School has actively involved parents feedback in shaping the curriculum, by seeking their feedback via a questionnaire (St Patricks RC 2024). This open-source approach makes resources accessible to everyone, fostering community engagement and parental involvement in educational planning. Involving parents fosters a collaborative learning environment and ensures broader support for the digital curriculum. Furthermore, parents have the opportunity to develop their digital skills alongside their children.

This case study has influenced that the design of the project should be a fully open source and available for the community, therefore there will not to be a database requirement for user registration.

Pencoed Primary School

Pencoed Primary School strategically utilises technology to enhance learning (Pencoed 2024). By carefully selecting and integrating diverse tools (such as Google

GSuite, Adobe suite, and green screen technology), the school ensures that technology is used to support, rather than replace, traditional teaching methods. Through diverse tool utilisation, Pencoed allows teachers to tailor instructions to specific needs while integrating technology strategically, as opposed to using technology as a tick box exercise.

Cornist Park C.P. School

Cornist Park C.P. School recognises the significance of teacher training and development. By conducting annual skills audits and providing a comprehensive CPD cycle, the school ensures that teachers are equipped with targeted training aligned to the DCF (Estyn 2016). The school provides staff with additional planning time and individual support from the Staff Digital Lead to integrate digital tools creatively into teaching.

It is worth noting that Cornist Park is a digital Pioneer School. Digital Pioneer Schools emphasise collaboration for successful DCF implementation across schools, which may have contributed to Cornist Park's successful implementation of the DCF.

Conclusion

The DCF will "continue to evolve, reflecting changing technologies" (Welsh Government 2018b) and to ensure educators' can effectively integrate the DCF into their teaching, it is essential to address gaps in digital skills, particularly for those whose initial training may have lacked a strong technological foundation. The successful implementation of the DCF relies on the availability of accessible, user-friendly resources and comprehensive professional development opportunities for educators. A notable gap exists in providing clear explanations of key terminology and in offering practical guidance for classroom implementation. Much of the DCF is comprised of statements with no lesson examples, activities or connection to real-world.

To address this need, this dissertation project aims to create an open-access, interactive web-based resource designed to support primary educators in implementing the DCFs Data and Computational Thinking strand into their lesson planning. The guide will be designed as a fully open-access resource, eliminating the need for registration or logins. This approach ensures that the guide is accessible to a wide audience, including teaching staff, parents and the wider community. The design will be informed by insights from case studies, prioritising user-friendliness and clarity. The guide will eliminate technological jargon as much as possible and consist of language that educators are more familiar with. This project aims to create an interactive web guide designed for primary educators to serve as a comprehensive resource in the Data and Computational Thinking strand of the DCF.

As this is tailored for primary school educators, the requirements set in progression steps two and three, students in years 3 – 6 primary school, will be used. The guide shouldn't be a manual that limits educators what to do or what to use, but rather it should be a tool that empowers educators with the skills and knowledge that will allow educators to use their judgement and apply these skills where they see fit. The guide will not prescribe specific tools or methods but will provide a foundation of skills and knowledge that educators can apply as they see fit. To enhance engagement, the guide will incorporate videos and activities, demonstrating how to teach computational thinking effectively.

This project was nominated by a digital lead in a primary school who identified a need for clearer explanations of key terminology and more user-friendly resources. The digital lead stated that there are no clear examples to integrate in lessons and necessitated a need for breakdown of the DCF in a clear and understanding manner. This web guide will fill this gap by offering an easily navigable and open-access platform. By creating a skills-focused resource, this project aims to inspire and support educators, paving the way for meaningful implementation of computational thinking skills in primary education.

Approach

This section details the methodologies and technical choices used during the project. By combining the structured progression of the Waterfall phases model, with the adaptability of Agile principles I was able to leverage the strengths of both methodologies. Waterfall provided a clear, linear project structure, while Agile allowed iterative development and flexibility, enabling the project to respond effectively to changes during software development.

Waterfall vs Agile Methodology

The Waterfall methodology follows a linear progression through five distinct phases: Requirements, Design, Implementation, Verification and Maintenance (Atlassian 2024a). Each phase must be completed before moving to the next phase, making this approach ideal for projects with clearly defined deliverables and milestones agreed with clients from the outset. While I structured my project with distinct deliverables, working independently without a client introduced flexibility in the project, which deviates from the rigidity of a traditional Waterfall approach. Furthermore, this is the first time that I have conducted a project of this scale, and my lack of experience introduced an unpredictability not suited to a rigid Waterfall approach, further reinforcing the need for adaptability.

In contrast, Agile methodologies embrace an iterative process involving planning, executing, and evaluating (Atlassian 2024b). Rather than a concrete long-term deliverable, Agile project management uses a series of shorter work periods, called ‘sprints’, to deliver incremental progress. Agile is well-suited for projects like mine, where adaptability and continuous learning are crucial. The iterative nature of Agile allowed me to reach milestones quickly, manage shifting priorities while releasing and refining sections continuously.

Regular feedback from clients for evaluation and adaptations between sprints is a critical aspect of Agile. Since I was working independently without a client or team, this feedback loop inherent in Agile was not as structured. However, the principles of Agile were still incorporated to enable flexibility and continuous improvement, where I used Agile principles to adjust as I gained knowledge.

Hybrid Waterfall-Agile Approach

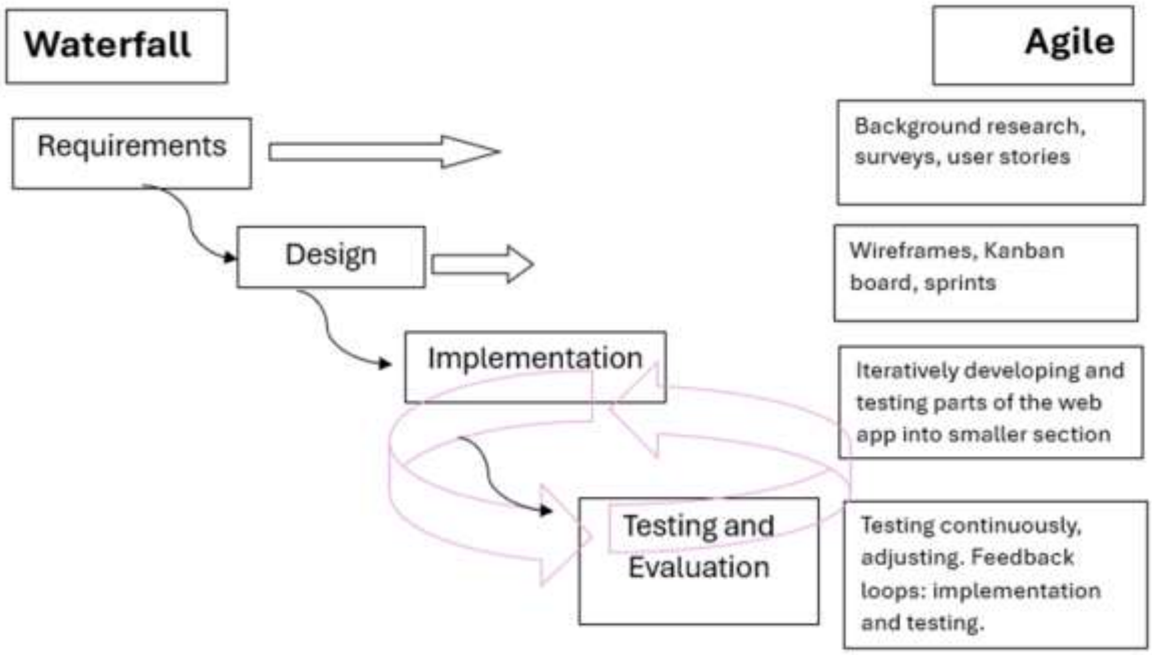

Waterfall
Agile
Requirements
Design
Implementation
Testing and Evaluation
Background research, surveys, user stories
Wireframes, Kanban board, sprints
Iteratively developing and testing parts of the web app into smaller section
Testing continuously, adjusting. Feedback loops: implementation and testing.

Figure 18: The Hybrid Approach undertaken throughout the project

For this project, I implemented a hybrid Waterfall-Agile methodology. This blended approach allowed me to benefit from the structured linear phased nature of Waterfall, while also benefiting from Agile's iterative and flexible nature being implemented through these phases. Waterfall's linear structure guided the early stages of the project, while Agile principles drove iteration and adaptability in later phases. This combination allowed me to build a solid foundation while remaining responsive to challenges and opportunities for improvement.

Initially, I followed a traditional Waterfall approach, where I progressed linearly through the **Requirements** and **Design** phases. During the **Requirements** phase, I conducted thorough background research and defined objectives. I used surveys and user stories, inspired by Agile practices, to understand the users' needs and gather functional and non-functional requirements. This phase helped me establish a clear direction for the project, ensuring that I could proceed to the next phase with a solid foundation, as well as identify any constraints.

After solidifying clear **Requirements** for the project, I progressed to the **Design** phase, where I created and refined wireframes. Although the process followed Waterfall's sequential flow, keeping in line with Agile principles, I regularly reviewed these designs, and would refine and make improvements, as I thought of better ways to enhance the guide’s content and information, ensuring user needs were met effectively.

The **Implementation** phase marked the start of the development process. I applied an Agile approach here, breaking down the web application’s functionality into smaller tasks or “sprints” that could be worked on incrementally. Using the Kanban board, I tracked my progress, moving tasks between the “To Do,” “In Progress,” “Done,” and “For Review” columns. This allowed for continuous feedback and improvements during the coding process, avoiding the pitfalls of Waterfall’s rigid, sequential nature.

Finally, during the **Testing and Evaluation** phase, I implemented a continuous evaluation process as advocated by Agile practices. Instead of waiting until the end to test the web application when fully complete, I conducted testing throughout development. This allowed me to identify bugs or issues early, and to refine the product incrementally, and ensure that the final app met the objectives defined in the initial **Requirements** phase.

Benefits of the Hybrid Approach

The hybrid approach enabled me to blend the clarity and structure of Waterfall with Agile’s adaptability. As I was working independently on this project, the Agile aspects were particularly beneficial, allowing me to continuously evaluate and improve the web application as I gained knowledge and experience during its development. The iterative approach seen within Agile methodologies allowed me to write working software with limited knowledge and build upon the functionality in iterations. I was able to manage and organise the project while maintaining flexibility to accommodate for unforeseen challenges and opportunities for improvement. Following an Agile approach enabled me to have better control and increased flexibility over my project. I was able to set clearer metrics for success in the form of tasks along my schedule, creating soft deadlines for myself as well as pushing myself with hard deadlines to avoid roadblocks and recognise what issues I was having. The iterative approach within the Agile methodology provided me with time to learn while I implemented throughout the process, whilst also providing additional time for research and to debug which was inevitable to happen. Following Agile allowed me to break down my workload into multiple phases throughout development, where I was able to shift the workload around safely with as much flexibility as I needed, with better controls of the issues that arose.

Technologies for Development

This section outlines the key technologies and tools selected to develop the interactive web guide. Since the project is small-scale, I selected technologies that were lightweight, simple, and effective, enabling the rapid creation of a functional protype to showcase the key features.

Web Application Approach

The specification of this project was that this application would be web-based. For the front-end of the web guide, I chose technologies that were straightforward, widely supported, and optimal for small-scale projects.

Front-End Development

HTML, CSS and JavaScript were chosen as front-end technologies because they are the basic standards for web development and are widely supported and adopted (MozDevNet 2024). HTML and CSS can be seamlessly integrated with other front-end technologies such as JavaScript and back-end technologies such as Flask. JavaScript was implemented to enhance the interactivity of the web guide, providing dynamic functionality such as collapsible sections, tooltips, and user-triggered actions. There were many JavaScript libraries I explored with, and ultimately, I incorporated Handsontable to create an interactive spreadsheet.

CSS Framework: Bootstrap

Bootstrap was used to ensure the web guide was visually appealing, responsive and consistent across devices and screen sizes. Its comprehensive documentation and active development community made it a reliable choice for this project, as it is continuously being updated and issues are fixed regularly. Bootstrap's minimalist design philosophy allowed for a clean and professional interface, emphasising user experience.

Back-End Development

Flask

For back-end development and routing, the Flask framework was selected. Flask is a popular, open-source, and lightweight Python framework known for its simplicity and flexibility. Its modular architecture provided the customisation capabilities required for this project while maintaining ease of use.

Version Control: GitHub

Version control is essential for tracking code changes and ensuring that the project was safely stored. GitHub was used for version control throughout the project's development. The link to the project repository is: https://github.com/ELIIMORRIS/DCF

Development Environment

VS Code

Visual Studio Code (VS Code) is a source code editor that supports numerous programming languages, including Python. It features syntax highlighting, auto-indentation and code snippets to enhance productivity, along with an interactive debugger for executing commands.

Technologies for User Research and Feedback

To collect user feedback during the research phase, Microsoft Forms was used to create and distribute surveys. This platform was chosen due to its simplicity, ease of use, and integration with my university email account, making it a practical tool for collecting data from participants.

Requirements Derivation

The requirements derivation phase played a critical role in shaping the subsequent design and implementation of the web application. This chapter details the methodology used to elicit requirements. By adopting a user-centred approach, the requirements were structured to address the practical needs and challenges faced by primary school educators in Wales. Requirements for the project were derived from a combination of background research, a survey and an analysis of existing educational resources available. In the context of this project, requirements refer to the specific needs and functionalities that the web guide must meet to serve its intended users (Wiegers and Beatty 2013).

Requirement Gathering Methodology

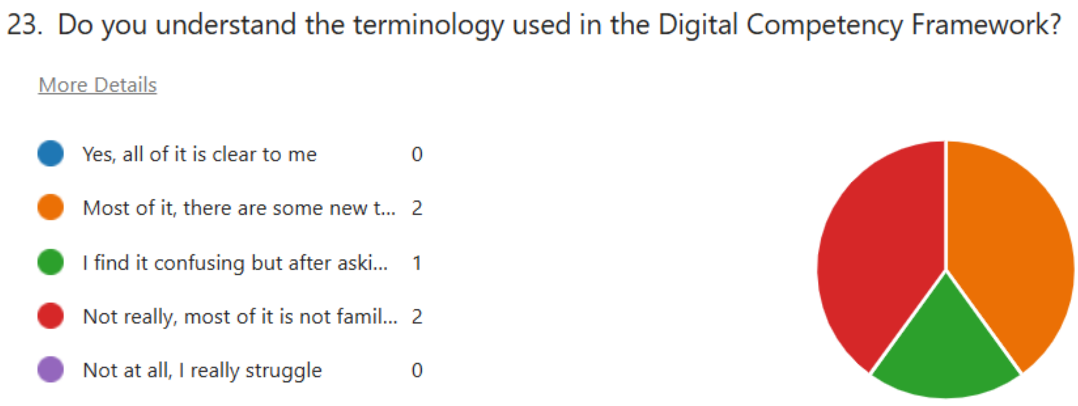
Survey Design and Implementation

To gather user requirements effectively, a survey was designed to collect data. Only primary school educators were invited to participate, and participants were recruited through professional networks, word of mouth and social media. The survey and its ethics approval are documented in [Appendix 1]. The survey used both qualitative and quantitative elements, to provide both measurable data and detailed insights into teachers' digital skills, confidence, and familiarity with the DCF. Most of the questions were quantitative and used response scales to measure the participant level, while the qualitative questions provided detailed, tailored responses from individuals with new insights, which allowed for a comprehensive understanding of user needs.

Analysis of Survey Results

The survey responses reveal valuable insights into the digital skills, challenges and professional development needs of primary school educators. Only one respondent mentioned recent training on computational skills (Technocamps), while others reported no recent training. This points to a gap in accessible professional development opportunities. All participants expressed a strong interest in improving their skills in creating engaging lessons across the curriculum and incorporating coding and computational thinking. Most respondents started that they are unfamiliar with

DCF terminology. Only 40% find "most of it" clear, while the remaining 60% find it confusing or unfamiliar.



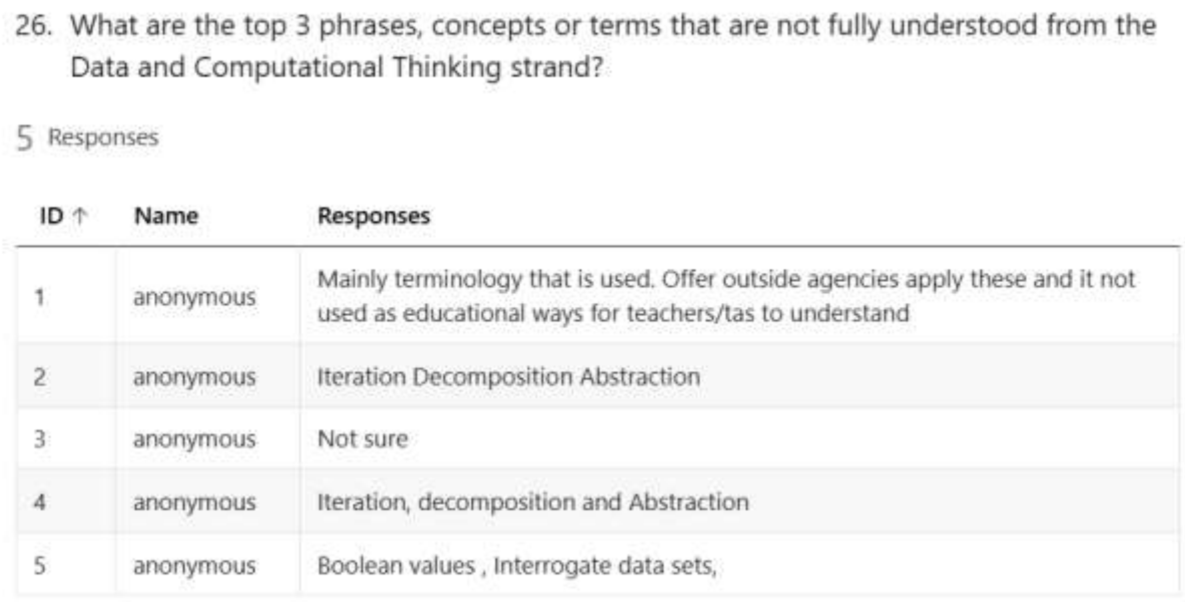
23. Do you understand the terminology used in the Digital Competency Framework?

More Details

🔵	Yes, all of it is clear to me	0
🟠	Most of it, there are some new t...	2
🟢	I find it confusing but after aski...	1
🔴	Not really, most of it is not famil...	2
🟣	Not at all, I really struggle	0

Figure 19: Survey results for DCF terminology

Participants particularly struggle with understanding key terms such as "Boolean values," "iteration," and "decomposition".

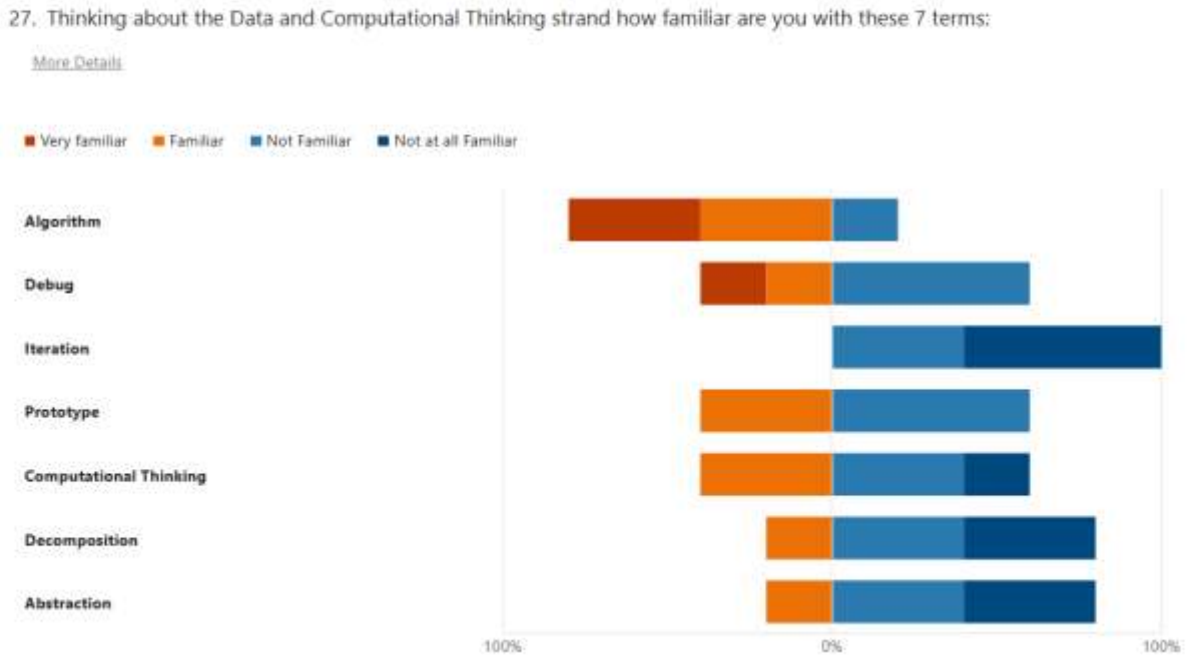


26. What are the top 3 phrases, concepts or terms that are not fully understood from the Data and Computational Thinking strand?

5 Responses

ID ↑	Name	Responses
1	anonymous	Mainly terminology that is used. Offer outside agencies apply these and it not used as educational ways for teachers/tas to understand
2	anonymous	Iteration Decomposition Abstraction
3	anonymous	Not sure
4	anonymous	Iteration, decomposition and Abstraction
5	anonymous	Boolean values , Interrogate data sets,

Figure 20: Survey results for DCT terminology



27. Thinking about the Data and Computational Thinking strand how familiar are you with these 7 terms:

Figure 21: Survey results for DCT familiarity

All respondents have access to smart boards, Chromebooks, and tablets/iPads, but only two have access to Micro:bits, and one has access to Lego Spike kits. No participants reported access to Raspberry Pi devices. Confidence levels vary across devices, with smart boards and Chromebooks being the most familiar tools, while devices like Micro:bit and Lego Spike are less utilised, suggesting that additional training is required.

Participants expressed strong interest in features for an interactive guide, including step-by-step tutorials and video demonstrations and lesson plans and activities aligned with the Welsh curriculum. None of the respondents reported being fully confident or proficient in applying the DCT strand's terminology or concepts. Key issues cited include limited understanding of terms like "abstraction," "iteration," and "data interrogation." All participants indicated that an online tool breaking down the DCF concepts would be extremely useful or useful.

Key Findings from the Survey

Participants are struggling with unfamiliar terminology and have not had recent digital training over the past year. An interactive guide that responds to these challenges is a favourable endeavour.

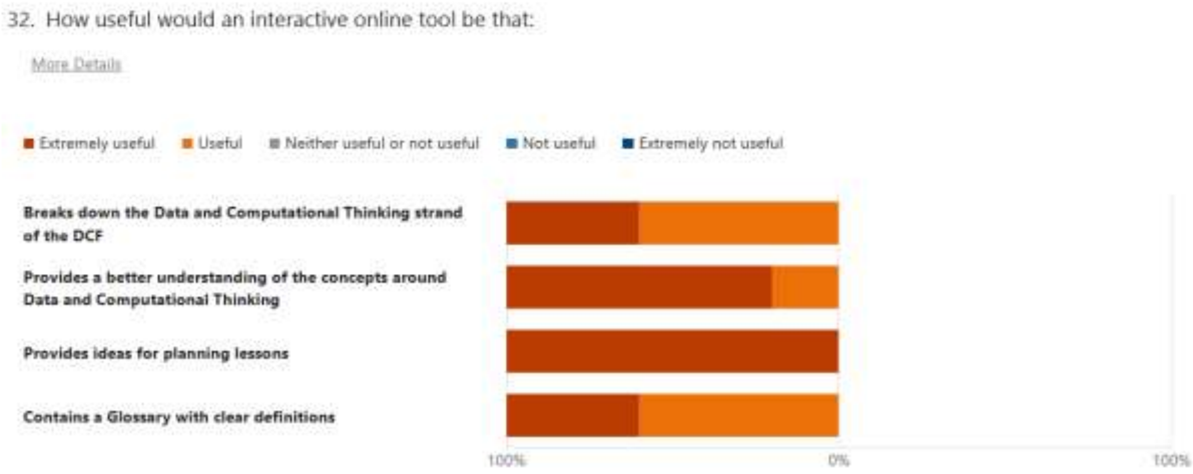


Figure 22: Survey results for usefulness of interactive online tool

Minimum Viable Product (MVP)

The MVP prioritises essential features that directly address the survey findings focusing on delivery value to educators through an intuitive and effective interface. The MVP emphasises easy to navigate simplified explanations of the DCT and DCF integration with accurate and up-to-date content.

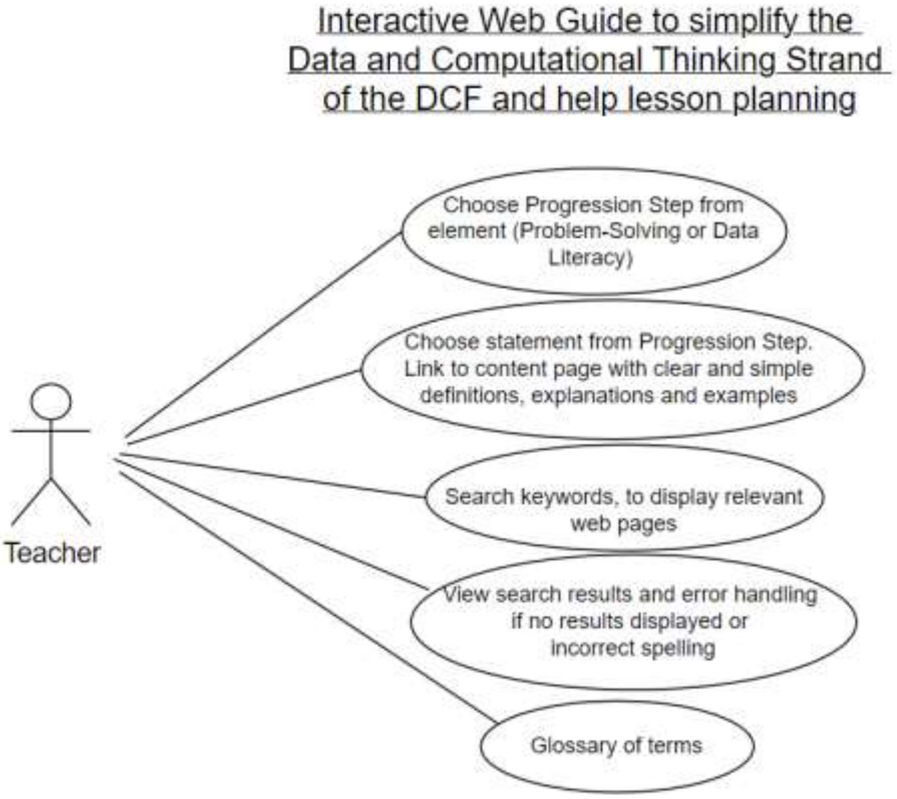

Interactive Web Guide to simplify the Data and Computational Thinking Strand of the DCF and help lesson planning
Choose Progression Step from element (Problem-Solving or Data Literacy)
Choose statement from Progression Step. Link to content page with clear and simple definitions, explanations and examples
Search keywords, to display relevant web pages
View search results and error handling if no results displayed or incorrect spelling
Glossary of terms
Teacher

Figure 23: MVP for web guide

User Story and Acceptance Criteria

To ensure the web guide meets the needs of primary school educators, user stories were developed as a tool to capture the requirements. A User Story is a description of a feature or functionality of a product from the perspective of an end user, and is usually written in simple, non-technical language, following a specific format:

“**As a** <type of user> **I want** <some goal> **so that** <some reason>”

Acceptance Criteria define the specific conditions that must be met for a user story to be considered complete. They provide a clear understanding of what is expected from the feature or functionality being developed. Acceptance Criteria are usually written as a list of statement, each describing a specific behaviour or outcome that must be achieved. This early phase provides a clear blueprint of what the project aims to achieve.

While User Stories provide a high-level overview of a feature by describing the user’s desire and how it benefits them, it is the Acceptance Criteria that breaks down the requirements that must be met for the feature to be considered complete. These requirements ensure that the deliverables meet the goals outlined in the User Story (AltexSoft 2023).

Core User Story

The development of the interactive web guide was driven by the following User Story:

“**As a** primary school educator in Wales, **I want** to access an interactive web guide that helps me improve my digital literacy, especially in Data and Computational Thinking, **so that** I can confidently incorporate the Digital Competency Framework strand of Data and Computational Thinking into my lesson planning and effectively teach these skills to my students.”

Acceptance Criteria

The following criteria define the necessary outcomes for the web guide to meet user needs:

- The web guide should be aligned with the specific expectations and learning objectives of the DCF data and computational thinking strand, as set out in the DCF.
- The web guide should provide clear and detailed explanations of data and computational thinking skills and concepts.
- The web guide should contain examples of activities for lesson plans to bring concepts to the classroom.
- The web guide should be easy to navigate, and users can easily find information for the topic and progression step that they choose.
- The web guide should have elements of interactivity to enhance user experience.
- The web guide should contain resources that are relevant to the curriculum with real World examples.

Functional and Non-Functional Requirements

After establishing the User Story, I could then gather and prioritise the projects requirements to ensure that the project would align with its intended purpose. Developing a complete web application requires meeting specific requirements across different aspects like features, user interface and security.

Functional Requirements

Lesson Planning Support

Educators can select a specific Progression Step, and the web guide provides a list of relevant learning activities and resources aligned with the selected Progression Step.

DCF Integration

The DCT is clearly represented within the web guide, and users can easily navigate between Progression Steps and their associated elements. Users can filter or search for activities and resources based on specific DCF elements – skills, knowledge areas, keywords - making it easy to find relevant content for different teaching goals.

Definitions for Key Terminology

Key terminology to be defined and recorded in one place in the form of a Glossary, for easy navigation.

Progress Tracking

Users can mark sections as complete when finished with the resource.

Welsh Language Implementation

The web application is available in both English and Welsh, with all content accurately translated.

Non-Functional Requirements

These will address how the system performs, making sure it is usable, accessible and maintainable:

Usability

The interface is intuitive and user-friendly, and easy to understand with clear instructions. The web guide provides clear visual cues and tooltips to assist users. There is consistent design and styling throughout the web application, with a navigation bar that allows users to easily find what they are looking for.

Error Handling

The web guide provides clear error messages and guidance if users encounter issues, ensuring users are redirected and they can resolve problems independently.

Security

All documents to be downloaded from the guide will be stored in static folder, making sure that anything downloaded from the web application is safe.

Exclusions and Considerations

While the integration of hardware-based learning is an essential aspect of developing digital competencies, this project limits its scope to conceptual understanding and software-based simulations. Survey findings have demonstrated there is a need to incorporate training on hardware such as Micro:bit, however this will not be included. For example, the **Micro:bit** was identified as a suitable tool due to its online simulation capabilities, which ensure equitable access for students in resource-limited schools. While it is an important aspect of developing digital skills through learning how to use hardware, the scope of this project will limit including this for now as the focus is on developing understanding of core computational thinking concepts. The Welsh Government has addressed this:

> *“To prepare for using the DCF, the main focus will be in relation to developing practitioners’/teachers’ digital confidence and competence, and ideas for integrating the DCF in their teaching, rather than in hardware or software.” (Welsh Government, 2018b).*

Conclusion

Based on the limitations identified in existing tools and insights gained from the survey, the project prioritises front-end development. This decision was driven by the need to create an intuitive and accessible user interface that would meet teachers' immediate needs, such as simplifying DCF concepts and producing content that is easy to understand.

Whilst advanced back-end features like user accounts and a community forum would enhance the platform, they are deferred to future iterations due to time constraints. The current focus is on delivering high-impact, essential functionalities that directly benefit educators.

The requirements gathered provide a solid foundation for development while allowing for future iterations to incorporate more advanced features based on user feedback and evolving educational needs.

Design and Development

This chapter documents the design phase of the web application. I focused on using a content-first approach and then implemented user-centred design (UCD) principles to ensure that the guide is both functional and user-friendly. Using wireframes, this design section plans out the content of the guide, and the layout of the website. Designing before implementation helps so that steps can be followed to build the guide to achieve the desired outcome. The design phase was essential for transforming conceptual ideas into reality. The design was informed by survey insights, and a clear understanding of user stories and acceptance criteria.

Design Methodology

The design methodology adopted approaches of **content-first** and **user-centered design (UCD)**. These methodologies informed decisions about content organisation, user interaction and visual layout.

Content-First Approach

Using a content-first methodology meant that the organisation and clarity of the guide's content took priority over the guide's visual design. This was essential for ensuring that the guide met the project objectives of creating a guide that supports teaching objectives outlined in the DCF and provides a structured, logical flow of information. Schools and practitioners must use the statements of what matters to guide the development of curriculum content**.** Structuring content to reflect progression steps and skills within DCT is crucial in the design of this project's web guide, and ensuring that resources, lesson ideas and glossary terms, align with DCF objectives. An initial objective of the project was to use resources that are already available, and to consolidate these in one guide.

Information Architecture

The structure of the web guide was designed to organise content logically and intuitively, enabling seamless navigation and comprehension. Key sections include:

Landing Page

The Landing Page introduces users to the web guide, providing an overview of the DCT strand and progression steps. Its design emphasises clarity and engagement. Each element of each progression step has a link to its own page for further information and resources, related to that element. The elements were taken exactly as they are shown on the DCF.

Content Pages

Each content page focuses on a specific skill set. The format of each page is a brief introduction explaining the concept, with example activities and ideas to illustrate concepts, and make abstract concepts tangible with relevant resources or guides. The content is aligned with the AoLE's, incorporating real-world examples from subjects like maths, computing, music and languages.

Problem-Solving and Modelling

Progression Step 2

- **break_down_a_problem.html**

Description given by DCF:	I can break down a problem to predict its outcome.
Ideas and concepts:	Introducing decomposition with real life examples such as identifying routine, everyday tasks.
Activity example:	"Let's break down the process of cleaning your bedroom." Ask learners to design their room and assign cleaning tasks.

- **debugging.html**

Description given by DCF:	I can detect and correct mistakes which cause instructions (a solution) to fail (debug).
Ideas and concepts:	Debugging, identify and correct errors in instructions.
Activity example:	Interactive debugging, to brush teeth in the correct order.

- **instructions_test_ideas.html**

Description given by DCF:	I can create and record verbal, written, and symbolic instructions to test ideas e.g. the order of waking up through a diagram or flowchart.
Ideas and concepts:	Communicate effectively, introduce symbols used in programming and flowchart, apply this to making algorithms.
Activity example:	Working in pairs have students create their own set of instructions for a simple task to give to each other. Create a flowchart for morning routine, using programming symbols.

- **instructions-change.html**

Description given by DCF:	I can change instructions to achieve a different outcome.
Ideas and concepts:	Introduce iteration , improving through repetition, order in maths, evaluating effectiveness of instructions.
Activity example:	Design a word ladder, where learners iteratively change one letter in a word to reach a target word. Give directions to a partner and a robot - find efficient or creative ways to solve navigation problems.

- **identify-repetitions.html**

Description given by DCF:	I can identify repetitions or loops in a sequence e.g. identify where to shorten a set of instructions by repeating steps, for instance, when learning a new song.
Ideas and concepts:	Recognise patterns, loops and repetitions in sequences. Identify where to shorten a set of instructions by looping.
Activity example:	Analyse the structure of songs, identifying repeated phrases or sections, simplify using loops.

Progression Step 3

- **refining-algorithms.htm**

Description given by DCF:	I can create and refine algorithms and flowcharts to solve problems, making use of features such as loops, Boolean values, and formulae.
Ideas and concepts:	Introduce pseudocode for initial algorithm creation, testing algorithm, refining the algorithm. Introduce basic programming concepts - loops, conditionals, Boolean values, formulae. Refining flowcharts.
Activity example:	Practice conditional flowchart design using simple problems like evaluating quiz scores to decide pass/fail.

- **importance-of-order.html**

Description given by DCF:	I can understand the importance of the order of statements within algorithms.
Ideas and concepts:	Algorithmic thinking, efficiency, flow of control, sequencing.
Activity example:	Introduce searching and sorting algorithms.

Data and Information Literacy

Progression Step 2

- **data-format.html**

Description given by DCF:	I can collect, enter, organise, and analyse data into different groups or formats e.g. tables, charts, databases, and spreadsheets.
Ideas and concepts:	Introduce different types of data, data collection, organising data, decided which is the best format.

Activity example:	Collect data with a class survey, organise the results and visualise the data collected.

- **evaluate-data.html**

Description given by DCF:	I can extract and evaluate information from tables and graphs to answer questions.
Ideas and concepts:	Understanding data from tables and graphs, and what the results mean. Answering questions from the results.
Activity example:	Showing tables and graphs, asking questions such as what is the largest/smallest value, what is the average, what is the most common value. Identify components such as title, axes and labels.

Progression Step 3

- **data-sets.html**

Description given by DCF:	I can construct, refine, and interrogate data sets within tables, charts, spreadsheets, and databases to test or support an investigation.
Ideas and concepts:	Working with and formatting data.
Activity example:	Create, edit and understand data sets using digital tools.

- **spreadsheet-formulae.html**

Description given by DCF:	I can use a range of spreadsheet formulae e.g. +, -, /, x, sum, average, max, min.
Ideas and concepts:	Introduce spreadsheet formulae and common functions. Introduce their real-world practical application.
Activity example:	Explore results in real time using an interactive spreadsheet with formulas already in use.

Additional Pages

- **Computational Thinking**

A dedicated section explaining computational thinking concepts, including links to relevant pages where these concepts are used in the application.

- **Resources Library**

A curated collection of external resources, including links to events and workshops, to support teachers implementing the DCF.

- **Glossary**

Clear definitions of key terms used throughout the application.

Chronological Development of the Design

The design and development process followed an iterative approach, evolving from initial concepts to functional wireframes and mock-ups.

Wireframes and Iterations

Wireframes are a tool to help visualise and plan website. They provide a clear idea of how the content will be organised and how users will interact with it. Wireframes provided a starting point for designing the web application with the focus on creating user-friendly, engaging and informative resource that support primary school teachers as they implement the DCF.

Stage 1: Initial Wireframe Concepts

The initial wireframes were created manually, by hand, I drew sketches to conceptualise the layout and navigation. Post-it notes and hand-drawn sketches allowed for rapid experimentation, enabling me to quickly visualise the structure and allowed me the ability to rearrange elements to optimise structure, enabling iterative improvements to layout and organisation. The focus was minimising cognitive load by avoiding clutter and walls of text while visualising how to link the competencies and progression steps.

These initial low-fidelity drafts captured the core concepts which were later digitised for refinement.

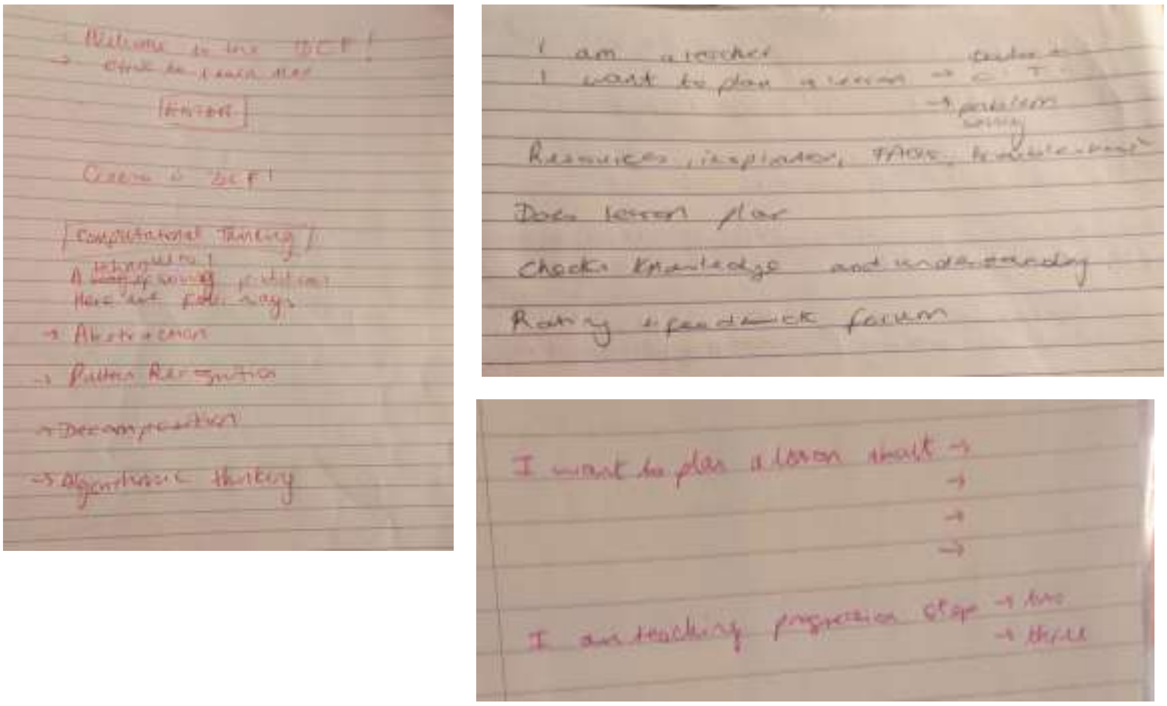


Figure 24: Initial wireframes by hand

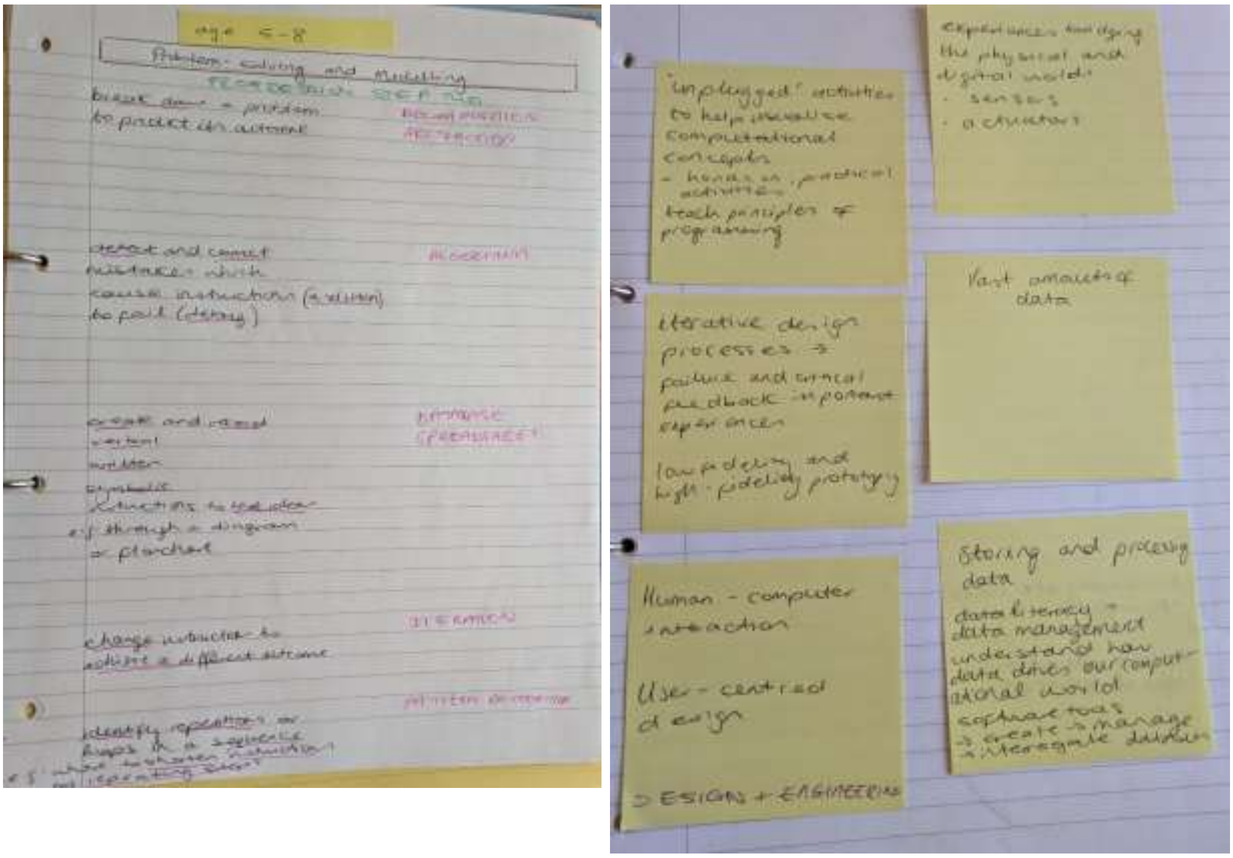
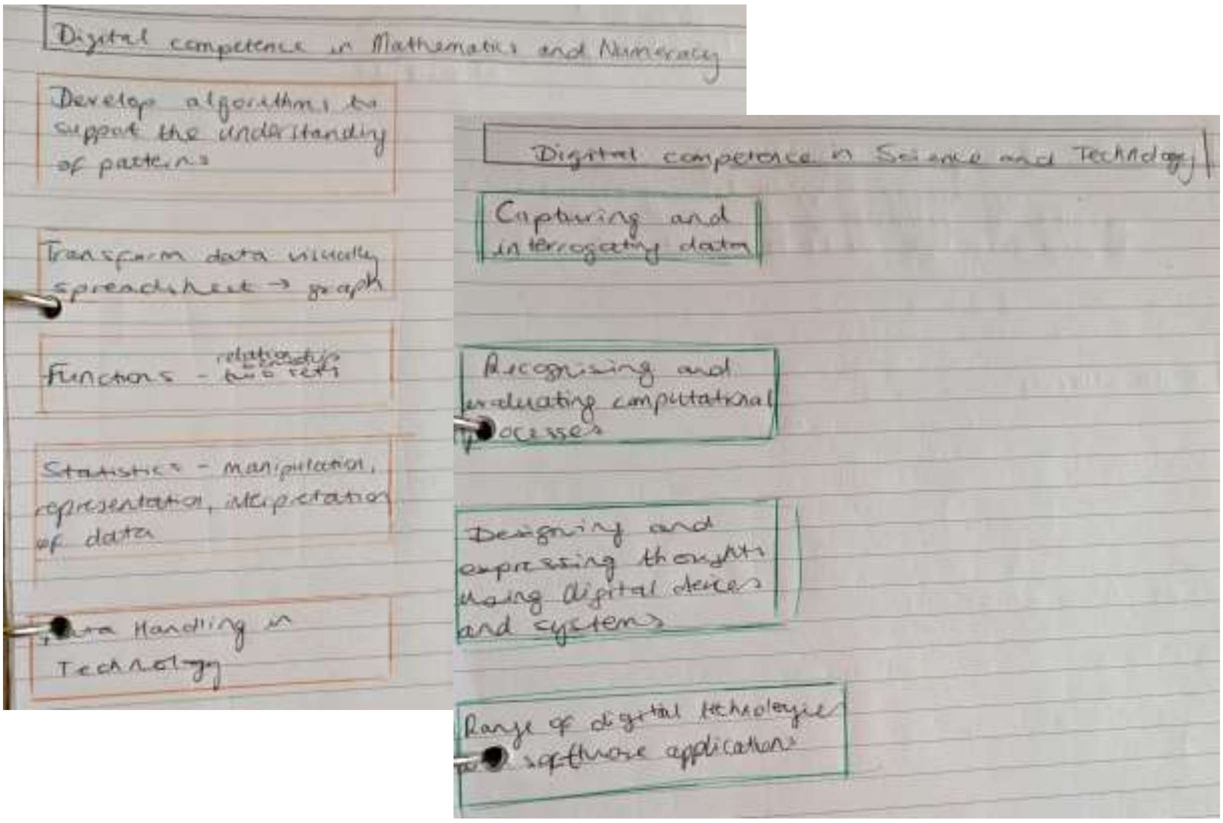

age 5-8
Problem-solving and Modelling
break down a problem to predict an outcome
detect and correct mistakes which cause instructions to fail (debug)
ALGORITHM
change instructions to achieve a different outcome
ITERATION
identify repetition or loops in a sequence
'unplugged' activities to help visualise computational concepts
- hands on, practical activities
teach principles of programming
iterative design processes →
failure and critical feedback important experiences
low-fidelity and high-fidelity prototyping
Human - computer interaction
User-centred design
DESIGN + ENGINEERING
experiences bridging the physical and digital worlds
· sensors
· actuators
Vast amounts of data
Storing and processing data
data literacy
data management
understand how data drives our computational world
software tools
→ create → manage
→ interrogate databases

Figure 25: Generating ideas for content while wireframing



Digital competence in Mathematics and Numeracy

Develop algorithms to support the understanding of patterns

Transform data visually spreadsheet → graph

Functions - relationship between two sets

Statistics - manipulation, representation, interpretation of data

Data Handling in Technology

Digital competence in Science and Technology

Capturing and interrogating data

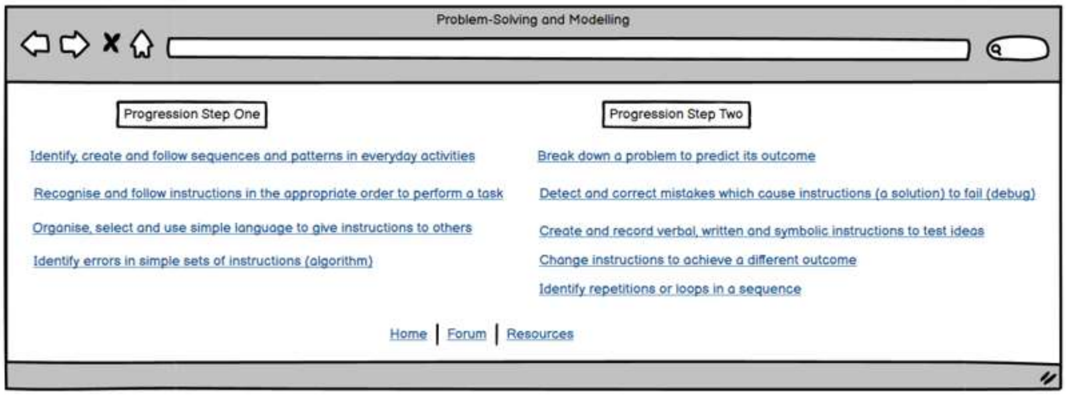
Recognising and evaluating computational processes

Designing and expressing thoughts using digital devices and systems

Range of digital technologies software applications

Stage 2: Low-Fidelity Wireframes

These initial hand-drawn designs were later transitioned to creating these wireframes digitally using tools such as Draw.io and Balsamiq, where I created individual pages of what the guide would look like. New iterations could be made quickly using features I liked from easily copying elements from previous designs to add to newer designs, which sped up the designing process of development. These early iterations provided a basic structure of the guide including a landing page with a central dashboard linking to key sections and a glossary page serving as a dedicated section for terminology. Experimenting with different layouts helped me to find the balance between clarity and functionality for the user.



Progression Step One

Identify, create and follow sequences and patterns in everyday activities

Recognise and follow instructions in the appropriate order to perform a task

Organise, select and use simple language to give instructions to others

Identify errors in simple sets of instructions (algorithm)

Progression Step Two

Break down a problem to predict its outcome

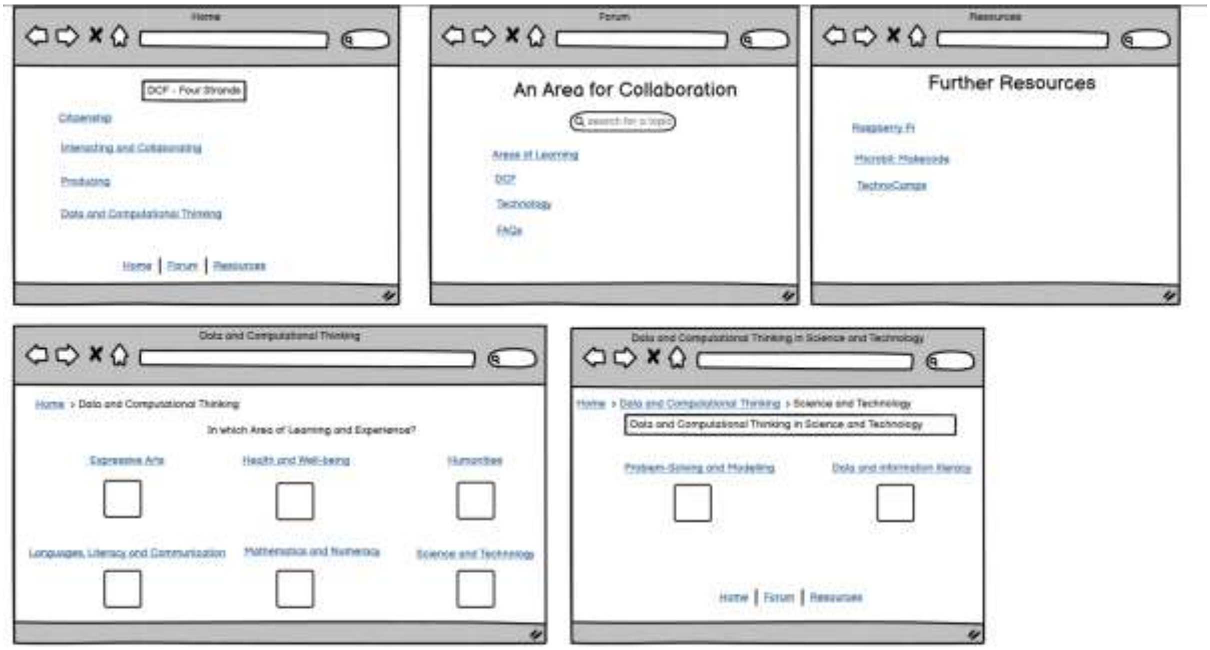
Detect and correct mistakes which cause instructions (a solution) to fail (debug)

Create and record verbal, written and symbolic instructions to test ideas

Change instructions to achieve a different outcome

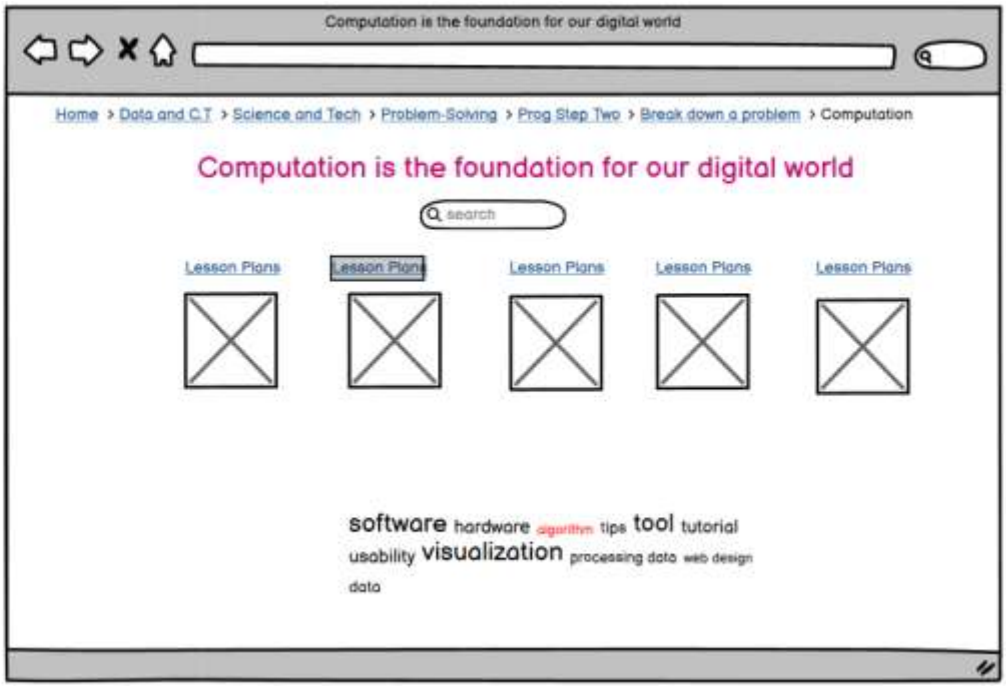
Identify repetitions or loops in a sequence

Figure 26: Initial mock-up of Problem-Solving and Modelling section



Home
DCF - Four Strands
Citizenship
Interacting and Collaborating
Producing
Data and Computational Thinking
Home | Forum | Resources
Forum
An Area for Collaboration
search for a topic
Areas of Learning
DCF
Technology
FAQs
Resources
Further Resources
Raspberry Pi
Microbit: Makecode
TechnoCamps
Data and Computational Thinking
Home > Data and Computational Thinking
In which Area of Learning and Experience?
Expressive Arts
Health and Well-being
Humanities
Languages, Literacy and Communication
Mathematics and Numeracy
Science and Technology
Data and Computational Thinking in Science and Technology
Home > Data and Computational Thinking > Science and Technology
Data and Computational Thinking in Science and Technology
Problem-Solving and Modelling
Data and information literacy
Home | Forum | Resources

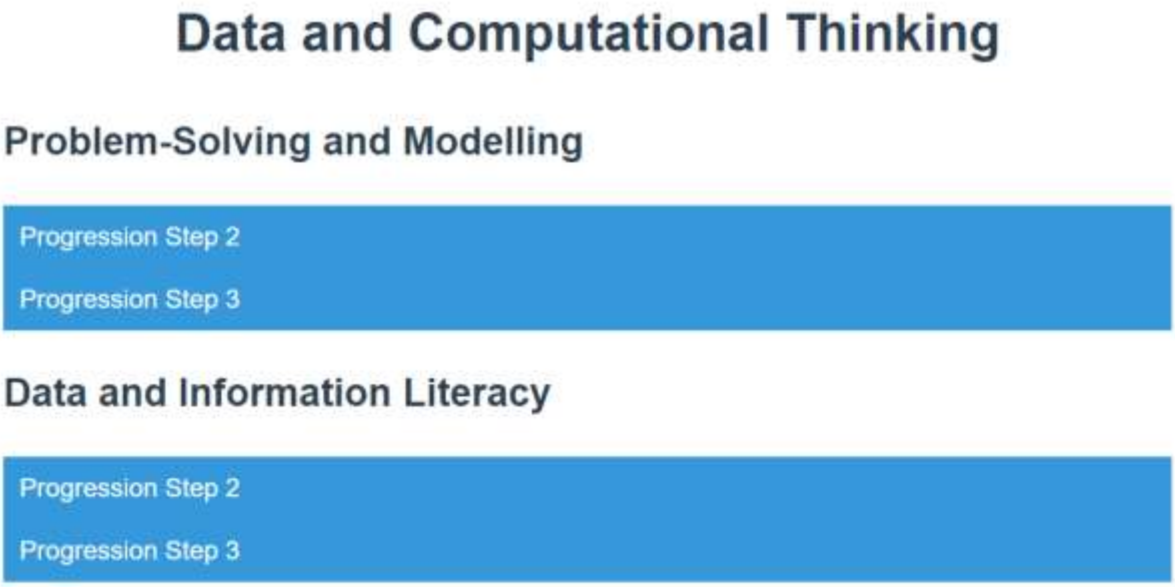
Figure 27: Early designs



Computation is the foundation for our digital world

Home > Data and C.T > Science and Tech > Problem-Solving > Prog Step Two > Break down a problem > Computation
search
Lesson Plans
Lesson Plans
Lesson Plans
Lesson Plans
Lesson Plans
software hardware algorithm tips tool tutorial
usability visualization processing data web design
data

Figure 28: Experimenting with designing word cloud as method of searching keywords



Data and Computational Thinking

Problem-Solving and Modelling

Progression Step 2

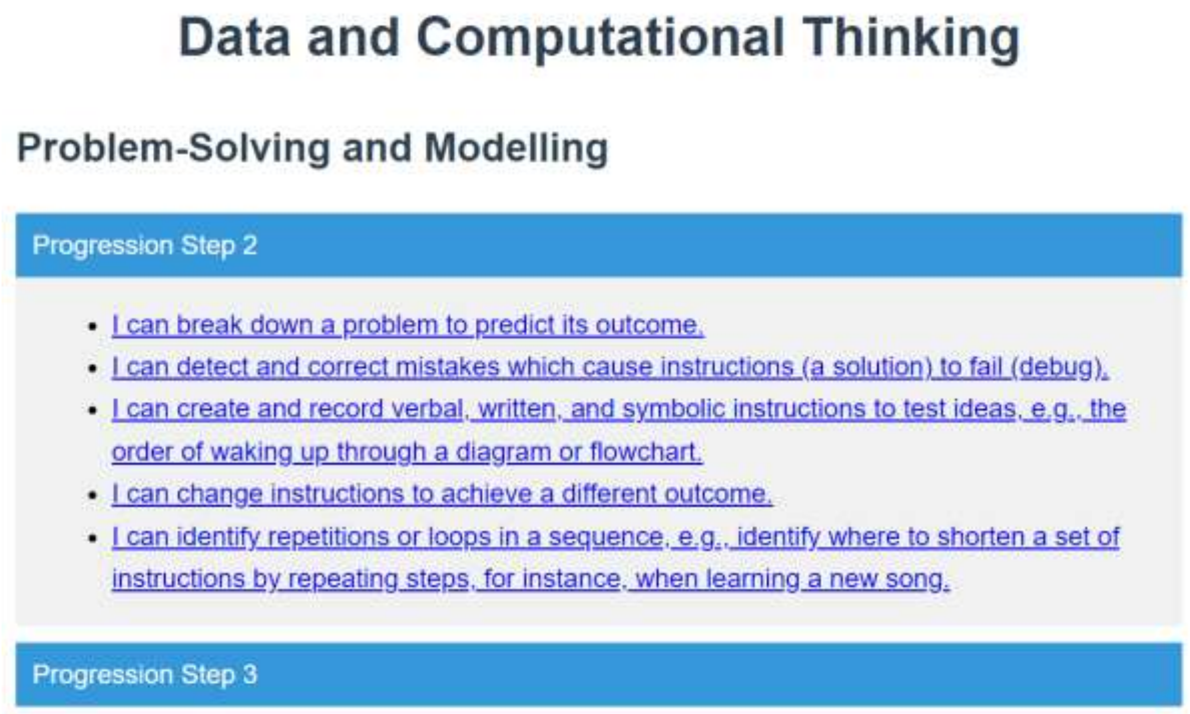
Progression Step 3

Data and Information Literacy

Progression Step 2

Progression Step 3

Figure 29: Initial Landing Page



Data and Computational Thinking

Problem-Solving and Modelling

Progression Step 2

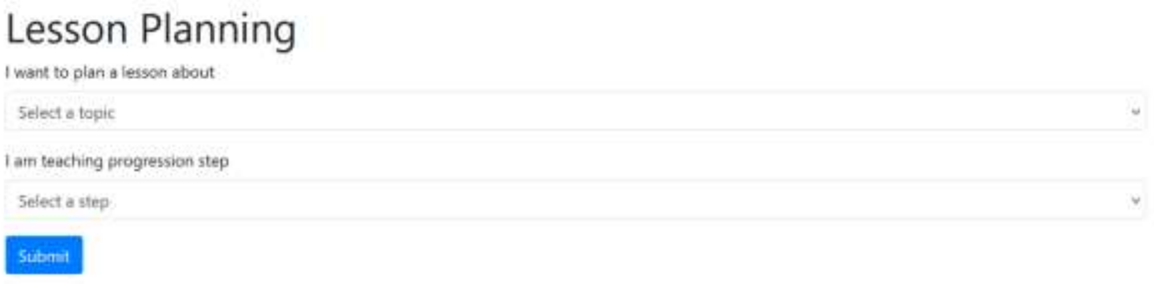
- I can break down a problem to predict its outcome.
- I can detect and correct mistakes which cause instructions (a solution) to fail (debug).
- I can create and record verbal, written, and symbolic instructions to test ideas, e.g., the order of waking up through a diagram or flowchart.
- I can change instructions to achieve a different outcome.
- I can identify repetitions or loops in a sequence, e.g., identify where to shorten a set of instructions by repeating steps, for instance, when learning a new song.

Progression Step 3

Figure 30: Initial Landing Page with linked elements

Stage 3: High-Fidelity Prototypes

Building upon the foundation of the low-fidelity versions, I created high-fidelity protypes that incorporated visual design elements, interactivity and responsive layouts. These protypes served as a more detailed representation of the final application, showcasing features like interactive dropdowns for progression steps.



Lesson Planning

I want to plan a lesson about

I am teaching progression step

Figure 31: Lesson Planning tool

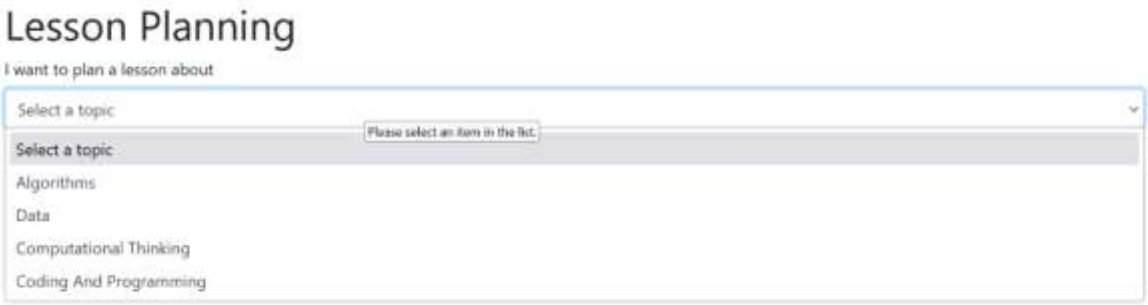


Figure 32: Lesson Planning tool topic dropdown

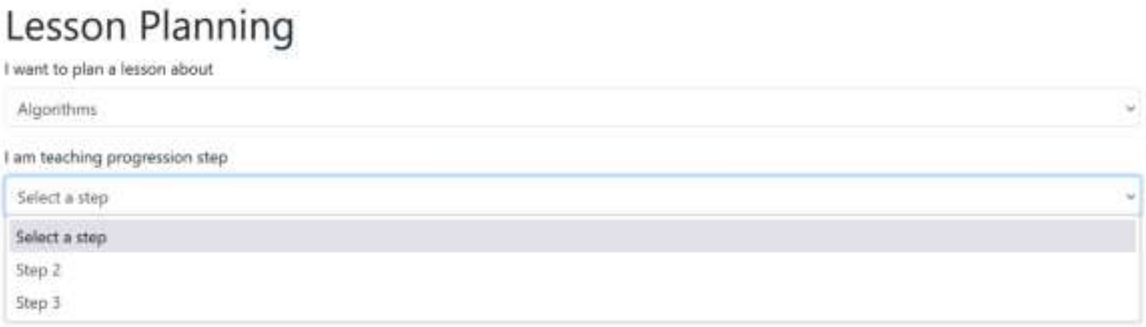
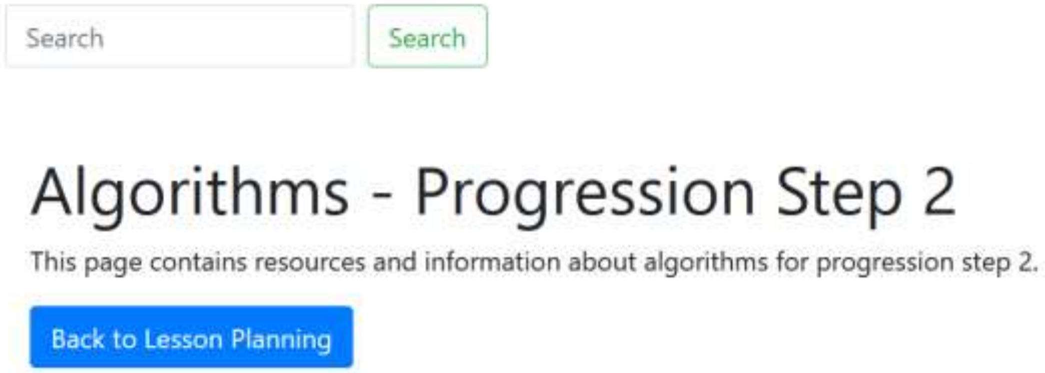
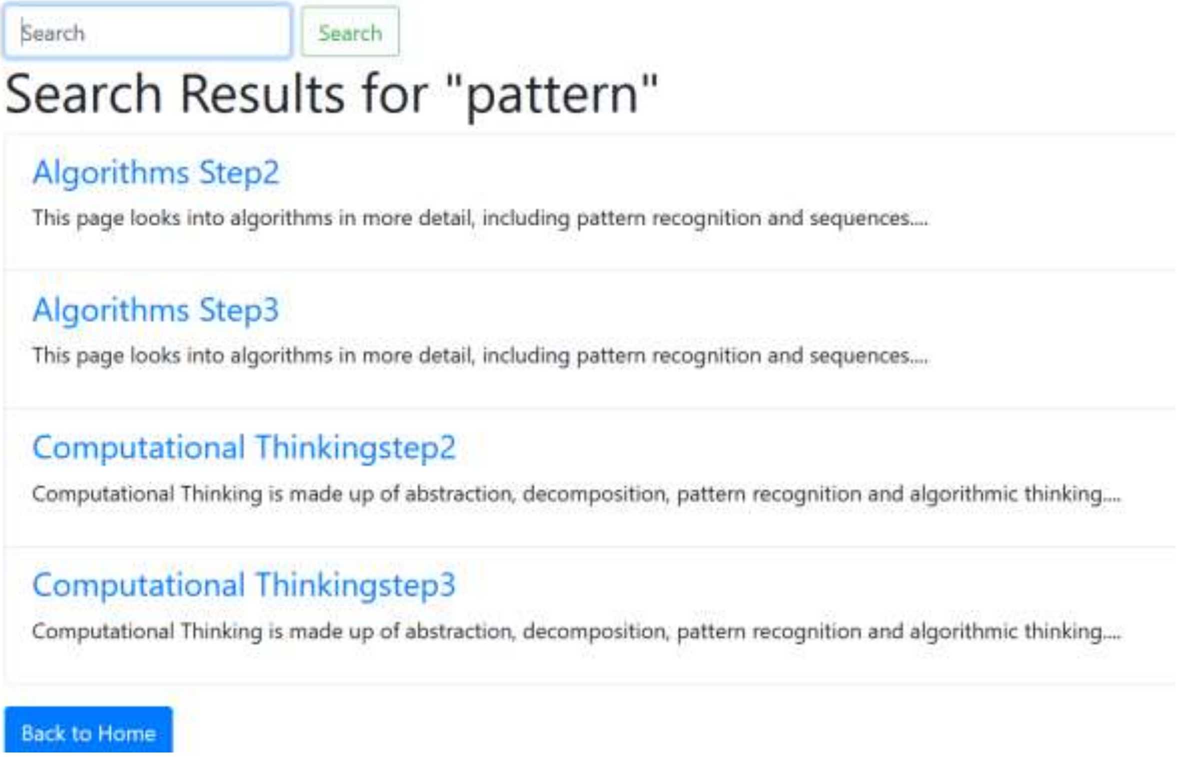


Figure 33: Lesson Planning progression step dropdown



This page contains resources and information about algorithms for progression step 2.

Figure 34: Page from dropdown selections



Search Results for "pattern"

Algorithms Step2

This page looks into algorithms in more detail, including pattern recognition and sequences....

Algorithms Step3

This page looks into algorithms in more detail, including pattern recognition and sequences....

Computational Thinkingstep2

Computational Thinking is made up of abstraction, decomposition, pattern recognition and algorithmic thinking....

Computational Thinkingstep3

Computational Thinking is made up of abstraction, decomposition, pattern recognition and algorithmic thinking....

Figure 35: Search results prototype

Iterative Design Enhancements and Design Evolution

The design constantly evolved through iterative refinement, addressing usability challenges and aligning with user-centered and content-first principles. The iterative process tackled issues like clutter and complex navigation, resulting in a clean, intuitive and accessible user interface. As the design progressed, dropdown menus were simplified, and navigation was enhanced with breadcrumbs for easier backtracking.

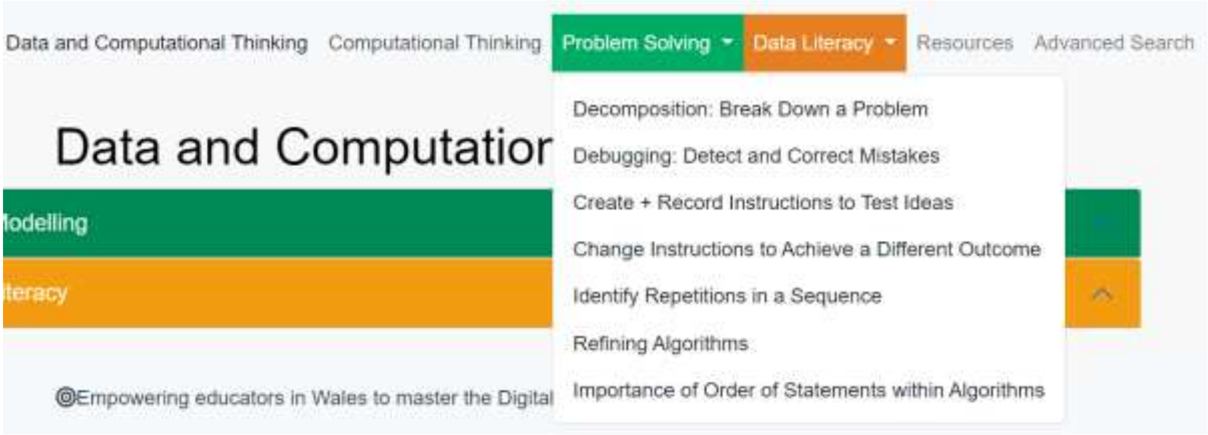

Data and Computational Thinking
Computational Thinking
Problem Solving
Data Literacy
Resources
Advanced Search
Data and Computatior
lodelling
iteracy
Empowering educators in Wales to master the Digital
Decomposition: Break Down a Problem
Debugging: Detect and Correct Mistakes
Create + Record Instructions to Test Ideas
Change Instructions to Achieve a Different Outcome
Identify Repetitions in a Sequence
Refining Algorithms
Importance of Order of Statements within Algorithms

Figure 36: Cluttered Problem-Solving drop-down menu from navbar

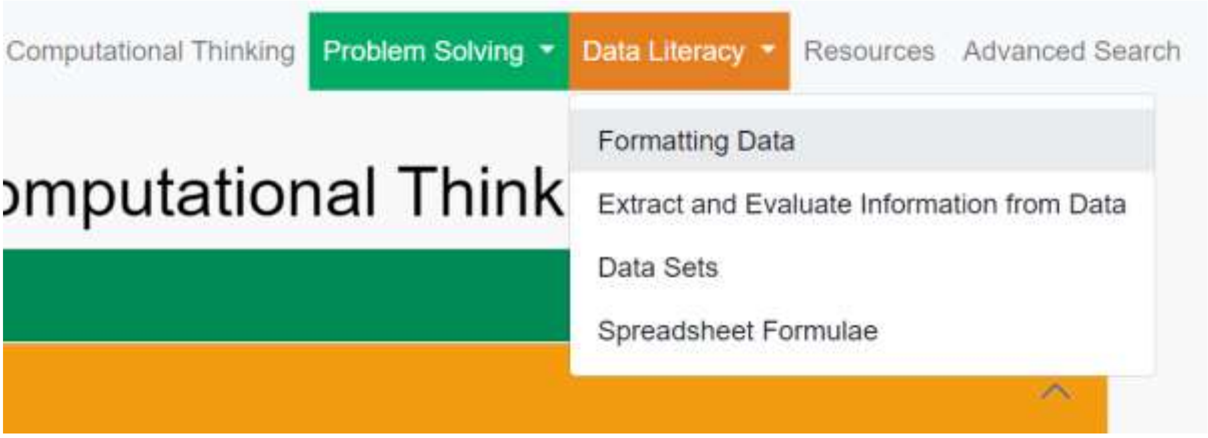


Figure 37: Cluttered Data Literacy drop down menu from navbar

During implementation I found the navbar drop downs too cluttered and re-arranged the design, as having the drop down to links may be a waste of memory as they are already linked. A clean style was to implement Bootstraps Breadcrumb feature which shows users how to quickly return to main page with all the navigation links.

High Level Website Architecture

To ensure a seamless user experience, I created a sitemap outlining the structure of the web application. This involved mapping out the main sections of the guide and breaking them down into sub-sections, considering the logical flow of information and how users would navigate between different parts of the site.

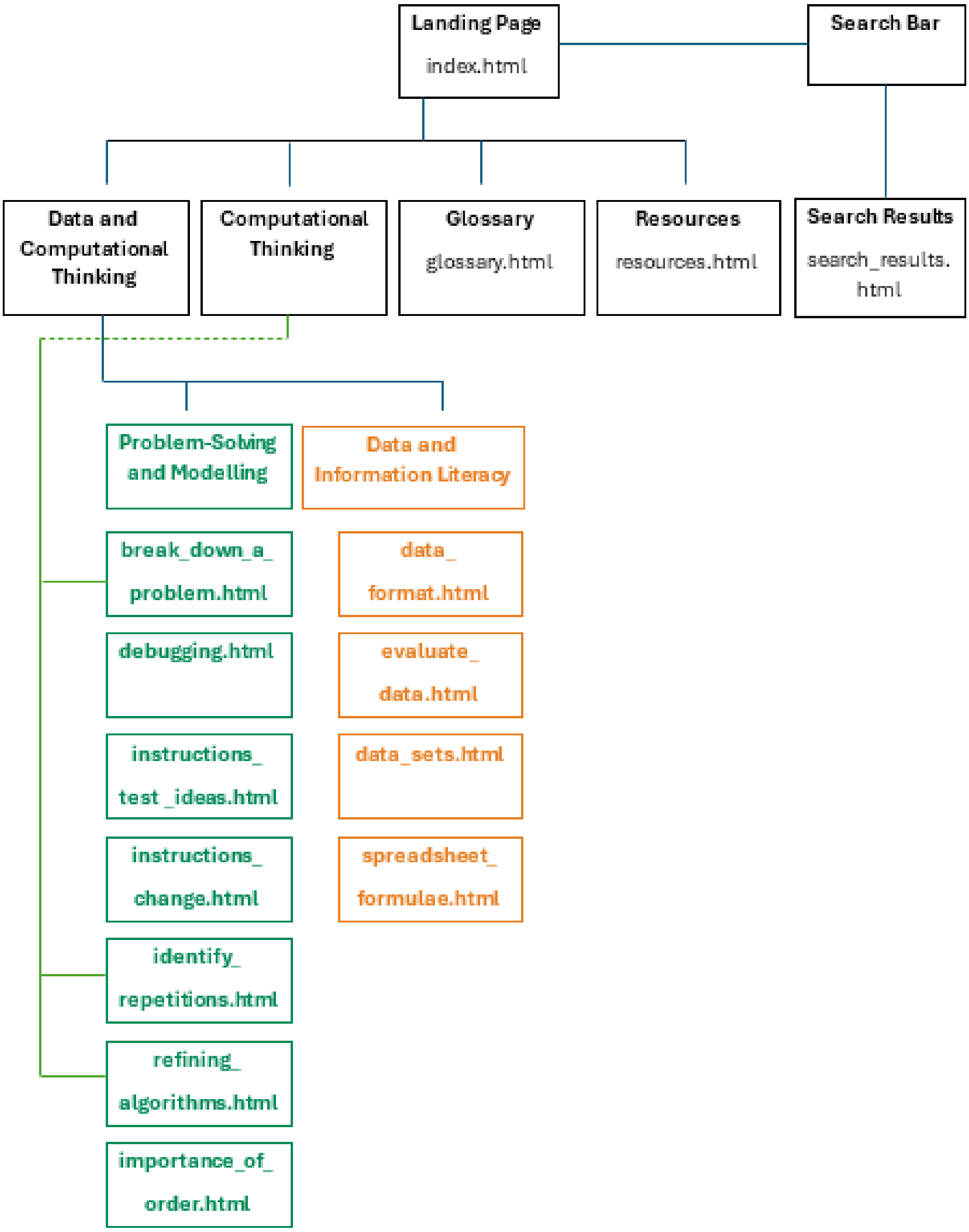


Figure 38: High Level Website Architecture diagram

Visual and User Interface Design

The visual design of the web guide followed **Nielsen's usability heuristics**, prioritising simplicity, consistency, and accessibility. Key visual design principles included a minimalist, clean and uncluttered layout, with ample white space to make the content easy to scan and navigate. A coordinated colour palette, where green was used for Problem-Solving and Modelling section and orange for Data Literacy, enhanced readability and distinguished sections. Fonts and layouts were carefully chosen to align with accessibility guidelines from the British Dyslexia Association. Font Awesome icons are used to break up text and provide intuitive visual cues, while alt text was included for all images and icons to accommodate users with visual impairments.

Conclusion

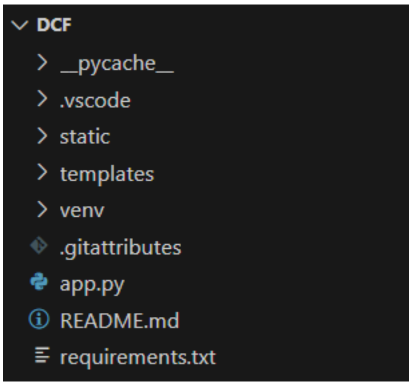
The design phase focused on balancing usability, accessibility and alignment with curriculum criteria. Wireframes and protypes provided a clear plan for development, facilitating a smooth transition to the implementation phase. Through iterative refinement, the application evolved into a practical tool for understanding Data and Computational Thinking concepts. The next chapter focuses on bringing these designs to life.

Implementation

The implementation phase transformed the design specifications into a functional and interactive web application. This section describes the project's structure, core features, and key technical decisions, complemented by relevant screenshots and code snippets.

Project Structure and Organisation

The project followed a modular structure for maintainability and scalability. The templates folder stored and organised HTML files for individual pages, leveraging Jinja2 templating for consistent design across pages via a base.html template. The static folder contained CSS and JavaScript for styling, and media files for interactivity. App.py served as the main application file, handling routing requests, dynamic content generation and rendering templates. This structure allowed for efficient development and clear separation of content, logic, and presentation.



```
DCF
  __pycache__
  .vscode
  static
  templates
  venv
  .gitattributes
  app.py
  README.md
  requirements.txt
```

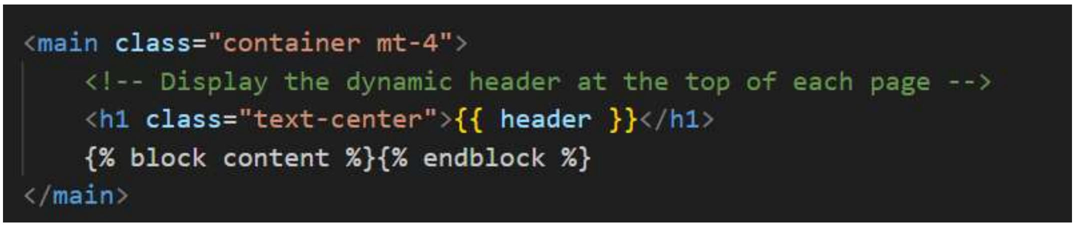
Figure 39: Project Structure

Flask Application (app.py)

The app.py file serves as the backend for the web application, managing data and dynamically rendering templates.

Dynamic Navigation

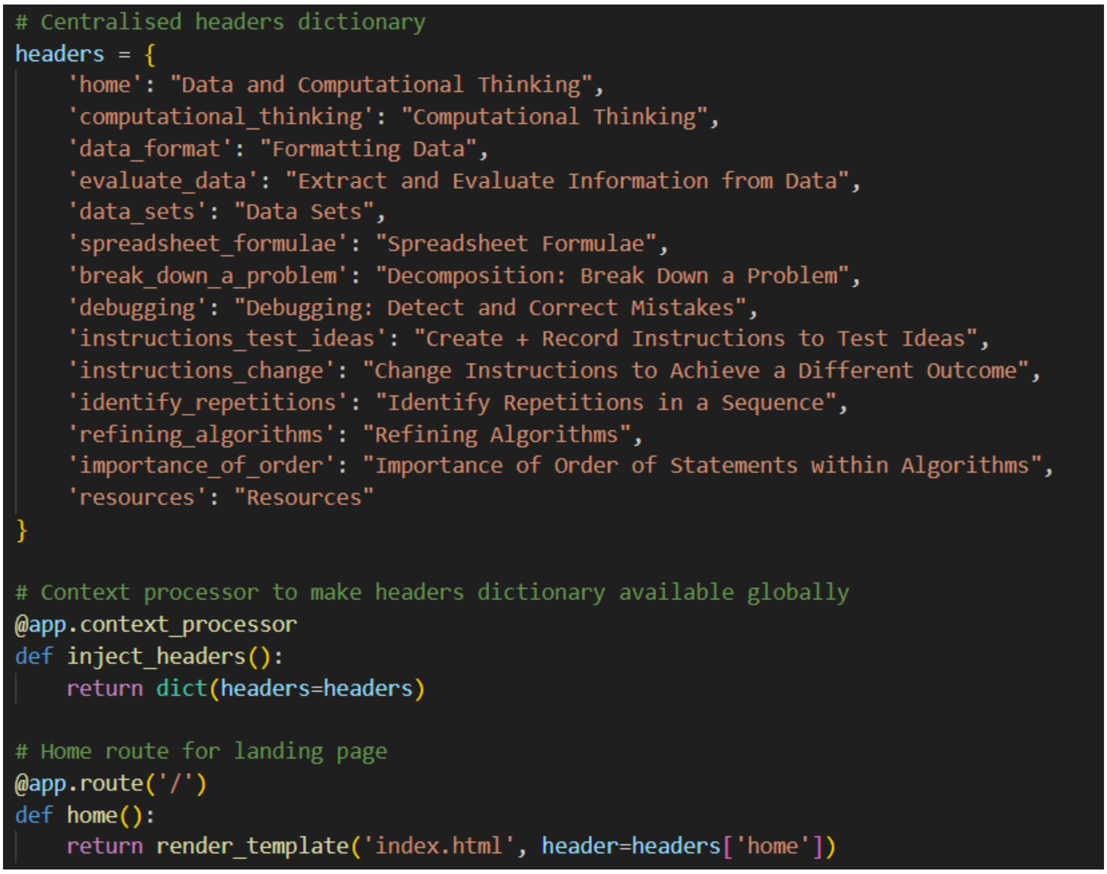
A dictionary-based headers system dynamically rendered navigation menus, reducing redundancy and enhancing scalability. Direct links such as ‘Home’ and ‘Computational Thinking’ remained hardcoded for simplicity.



```
<main class="container mt-4">
    <!-- Display the dynamic header at the top of each page -->
    <h1 class="text-center">{{ header }}</h1>
    {% block content %}{% endblock %}
</main>
```

Figure 40: Dynamic header code

Using the dictionary is the most helpful for the dropdown items, where there are several links with unique headers, to avoid hardcoding in multiple places.



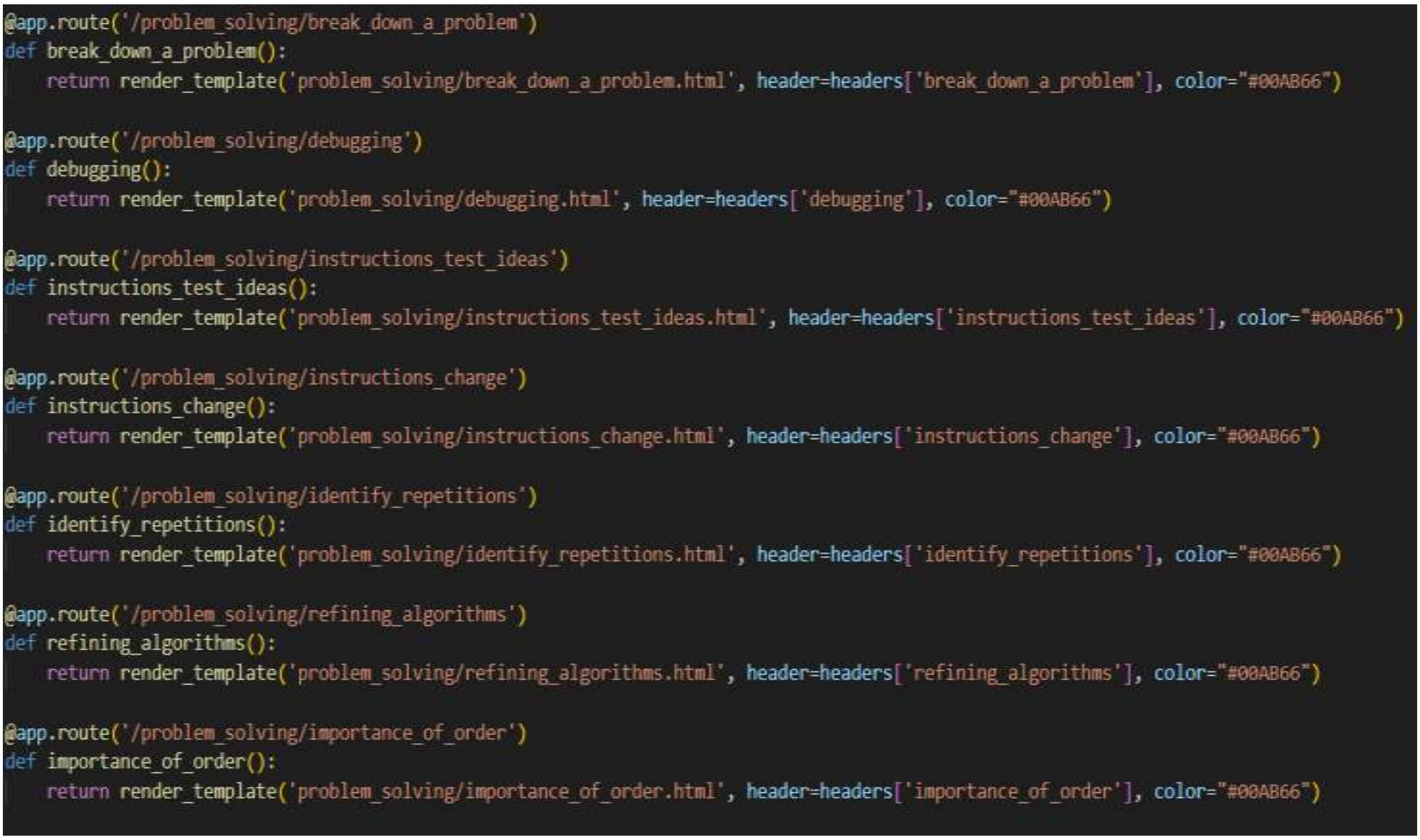
```
# Centralised headers dictionary
headers = {
    'home': "Data and Computational Thinking",
    'computational_thinking': "Computational Thinking",
    'data_format': "Formatting Data",
    'evaluate_data': "Extract and Evaluate Information from Data",
    'data_sets': "Data Sets",
    'spreadsheet_formulae': "Spreadsheet Formulae",
    'break_down_a_problem': "Decomposition: Break Down a Problem",
    'debugging': "Debugging: Detect and Correct Mistakes",
    'instructions_test_ideas': "Create + Record Instructions to Test Ideas",
    'instructions_change': "Change Instructions to Achieve a Different Outcome",
    'identify_repetitions': "Identify Repetitions in a Sequence",
    'refining_algorithms': "Refining Algorithms",
    'importance_of_order': "Importance of Order of Statements within Algorithms",
    'resources': "Resources"
}

# Context processor to make headers dictionary available globally
@app.context_processor
def inject_headers():
    return dict(headers=headers)

# Home route for landing page
@app.route('/')
def home():
    return render_template('index.html', header=headers['home'])
```

Figure 41: Illustrates the headers dictionary, dynamically displaying the correct headers on respective pages

Routes

Each route in app.py corresponds to a specific page of the website. For content pages, these were stored in their respective sub-folder of the template folder i.e. content for Problem-Solving and Modelling were stored in that sub-folder.

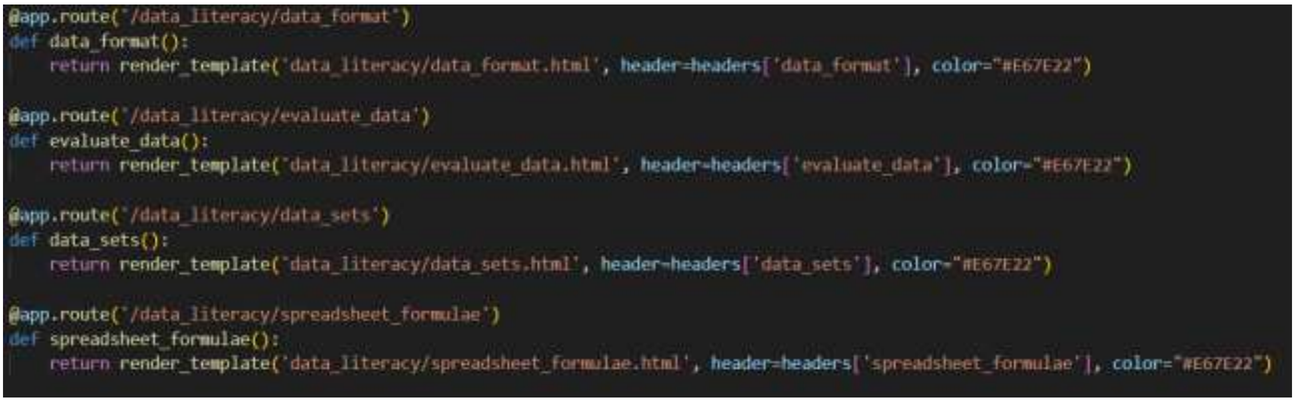
Problem-Solving routes dynamically render templates with a specific header, colour theme and content:



```
@app.route('/problem_solving/break_down_a_problem')
def break_down_a_problem():
    return render_template('problem_solving/break_down_a_problem.html', header=headers['break_down_a_problem'], color="#00AB66")

@app.route('/problem_solving/debugging')
def debugging():
    return render_template('problem_solving/debugging.html', header=headers['debugging'], color="#00AB66")

@app.route('/problem_solving/instructions_test_ideas')
def instructions_test_ideas():
    return render_template('problem_solving/instructions_test_ideas.html', header=headers['instructions_test_ideas'], color="#00AB66")

@app.route('/problem_solving/instructions_change')
def instructions_change():
    return render_template('problem_solving/instructions_change.html', header=headers['instructions_change'], color="#00AB66")

@app.route('/problem_solving/identify_repetitions')
def identify_repetitions():
    return render_template('problem_solving/identify_repetitions.html', header=headers['identify_repetitions'], color="#00AB66")

@app.route('/problem_solving/refining_algorithms')
def refining_algorithms():
    return render_template('problem_solving/refining_algorithms.html', header=headers['refining_algorithms'], color="#00AB66")

@app.route('/problem_solving/importance_of_order')
def importance_of_order():
    return render_template('problem_solving/importance_of_order.html', header=headers['importance_of_order'], color="#00AB66")
```

Figure 42: Routes for Problem-Solving content pages

Data Literacy routes dynamically render pages with a specific header, colour theme and content:

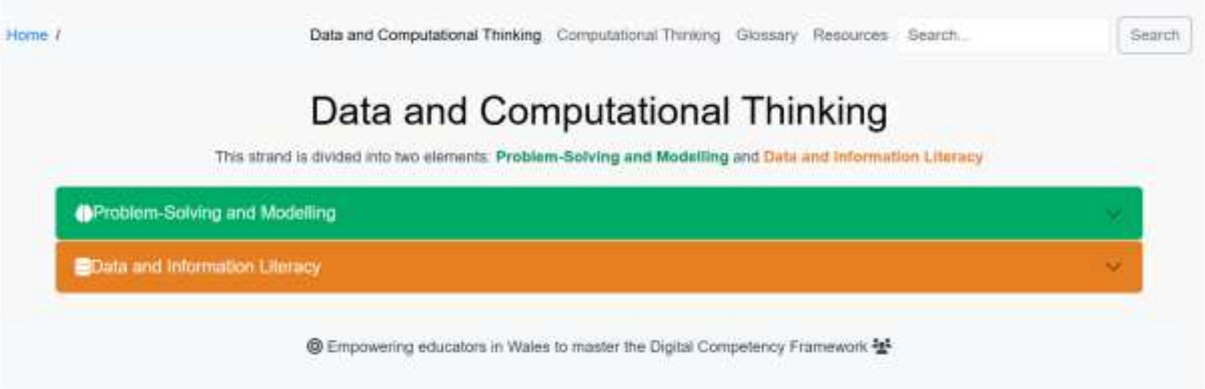


```
@app.route('/data_literacy/data_format')
def data_format():
    return render_template('data_literacy/data_format.html', header=headers['data_format'], color="#E67E22")

@app.route('/data_literacy/evaluate_data')
def evaluate_data():
    return render_template('data_literacy/evaluate_data.html', header=headers['evaluate_data'], color="#E67E22")

@app.route('/data_literacy/data_sets')
def data_sets():
    return render_template('data_literacy/data_sets.html', header=headers['data_sets'], color="#E67E22")

@app.route('/data_literacy/spreadsheet_formulae')
def spreadsheet_formulae():
    return render_template('data_literacy/spreadsheet_formulae.html', header=headers['spreadsheet_formulae'], color="#E67E22")
```

Figure 43: Routes for Data and Information Literacy content pages

Landing Page

When the project is executed, the landing page serves as the user's entry point to the guide. This page incorporates all the skills outlined in the two elements of the Data and Computational Thinking strand, and direct links to other sections of the guide.


Home /
Data and Computational Thinking
Computational Thinking
Glossary
Resources
Search...
Search
Problem-Solving and Modelling
Data and Information Literacy

Data and Computational Thinking

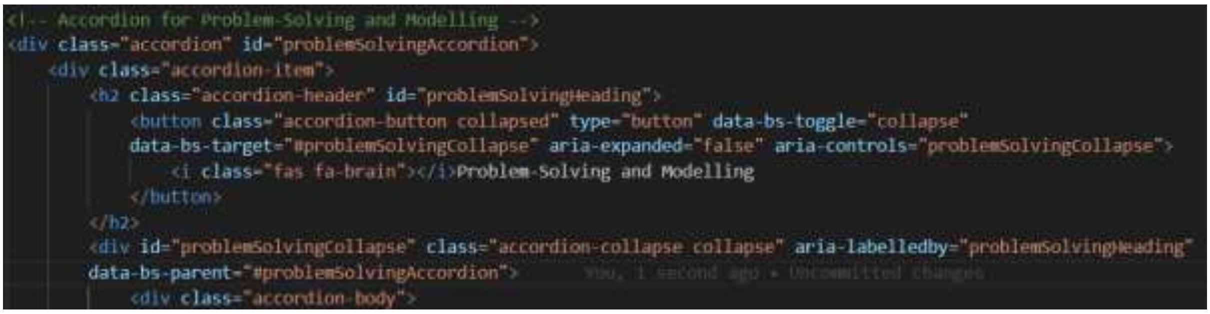
This strand is divided into two elements: Problem-Solving and Modelling and Data and Information Literacy

Empowering educators in Wales to master the Digital Competency Framework

Figure 44: Landing Page displayed in browser

Accordion Structure

To enhance content organisation and usability, the Bootstrap accordion component was used to create collapsible sections. Each major section can be expanded or collapsed, effectively reducing visual clutter and maintaining a clean layout. This design ensures that only the information selected by the user is displayed, providing a more focused and user-friendly experience.

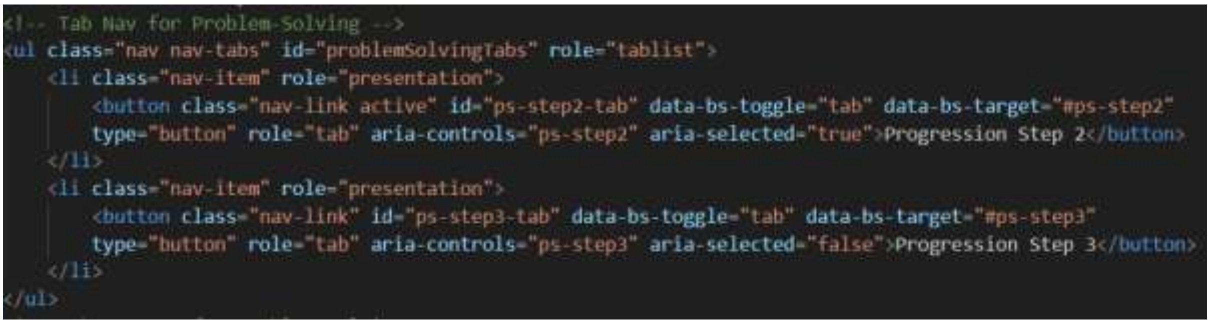


```html
<!-- Accordion for Problem-Solving and Modelling -->
<div class="accordion" id="problemSolvingAccordion">
    <div class="accordion-item">
        <h2 class="accordion-header" id="problemSolvingHeading">
            <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse"
            data-bs-target="#problemSolvingCollapse" aria-expanded="false" aria-controls="problemSolvingCollapse">
                <i class="fas fa-brain"></i>Problem-Solving and Modelling
            </button>
        </h2>
        <div id="problemSolvingCollapse" class="accordion-collapse collapse" aria-labelledby="problemSolvingHeading"
        data-bs-parent="#problemSolvingAccordion">      You, 1 second ago • Uncommitted changes
            <div class="accordion-body">
```

Figure 45: Accordion for Problem-Solving and Modelling code

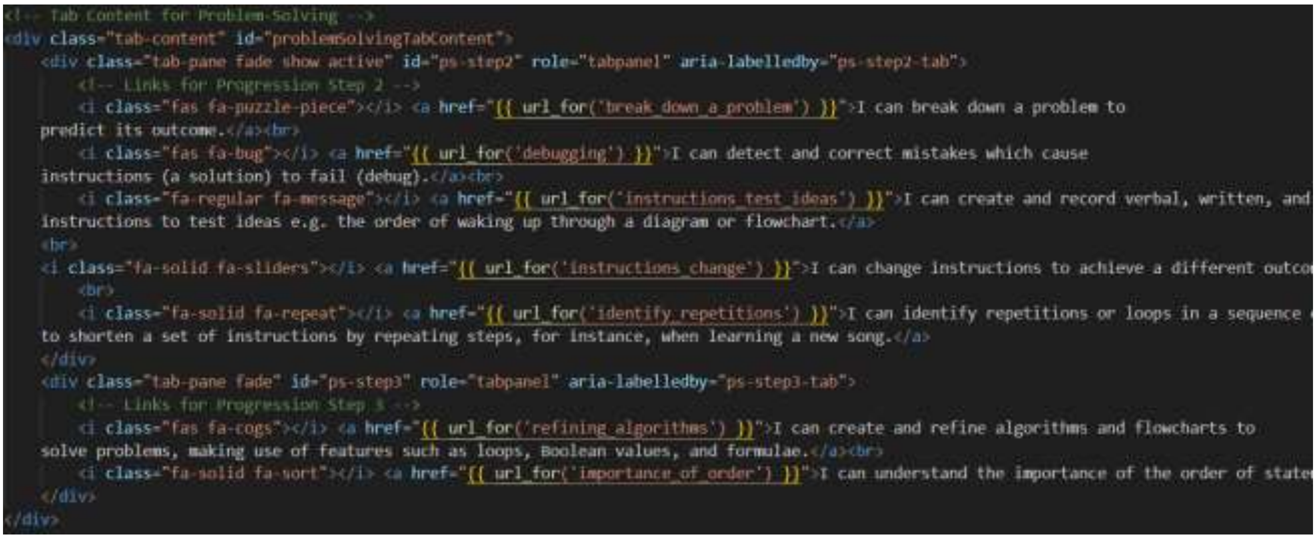
Tab Navigation and Content

Within each accordion section, Bootstrap tabs were implemented to facilitate navigation between progression steps. These tabs allow users to seamlessly switch between different progression steps. By separating the progression steps within the tabs, the interface offers a streamlined way for users to choose what information they want to be displayed.



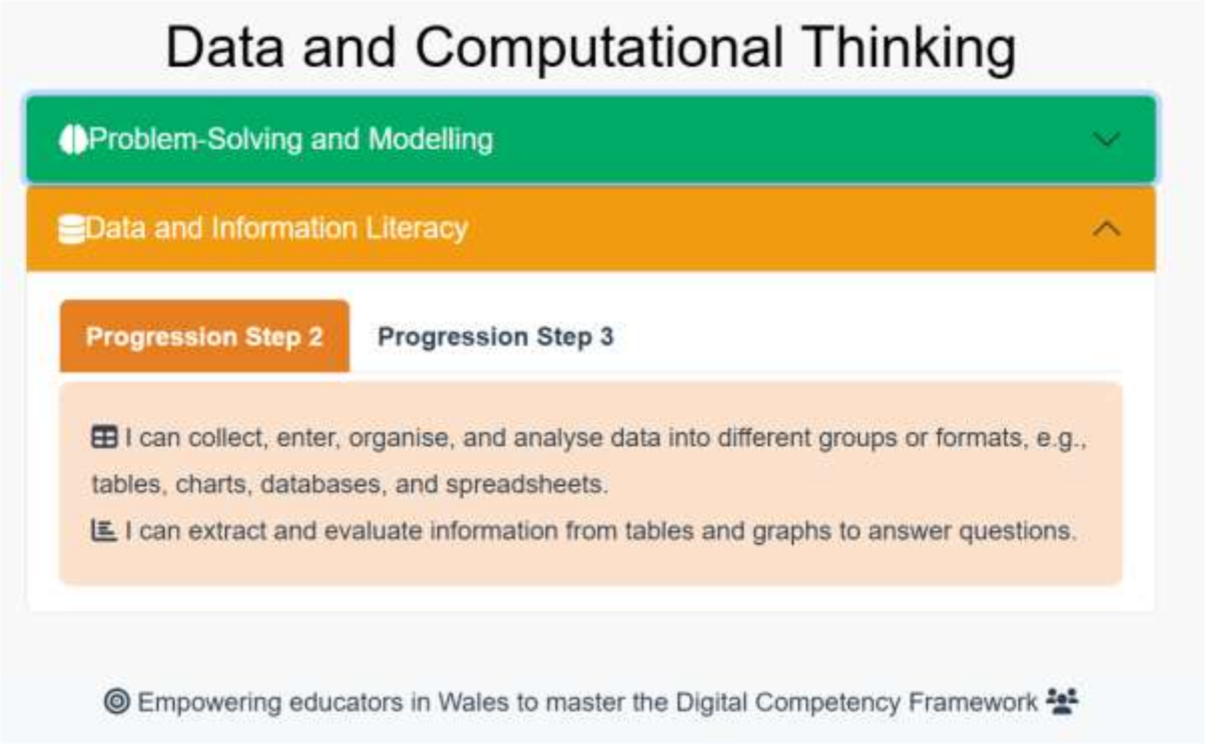
```html
<!-- Tab Nav for Problem-Solving -->
<ul class="nav nav-tabs" id="problemSolvingTabs" role="tablist">
    <li class="nav-item" role="presentation">
        <button class="nav-link active" id="ps-step2-tab" data-bs-toggle="tab" data-bs-target="#ps-step2"
        type="button" role="tab" aria-controls="ps-step2" aria-selected="true">Progression Step 2</button>
    </li>
    <li class="nav-item" role="presentation">
        <button class="nav-link" id="ps-step3-tab" data-bs-toggle="tab" data-bs-target="#ps-step3"
        type="button" role="tab" aria-controls="ps-step3" aria-selected="false">Progression Step 3</button>
    </li>
</ul>
```

Figure 46: Code for tab navigation



```html
<!-- Tab Content for Problem-Solving -->
<div class="tab-content" id="problemSolvingTabContent">
    <div class="tab-pane fade show active" id="ps-step2" role="tabpanel" aria-labelledby="ps-step2-tab">
        <!-- Links for Progression Step 2 -->
        <i class="fas fa-puzzle-piece"></i> <a href="{{ url_for('break_down_a_problem') }}">I can break down a problem to
    predict its outcome.</a><br>
        <i class="fas fa-bug"></i> <a href="{{ url_for('debugging') }}">I can detect and correct mistakes which cause
    instructions (a solution) to fail (debug).</a><br>
        <i class="fa-regular fa-message"></i> <a href="{{ url_for('instructions_test_ideas') }}">I can create and record verbal, written, and
    instructions to test ideas e.g. the order of waking up through a diagram or flowchart.</a>
    <br>
    <i class="fa-solid fa-sliders"></i> <a href="{{ url_for('instructions_change') }}">I can change instructions to achieve a different outco
        <br>
        <i class="fa-solid fa-repeat"></i> <a href="{{ url_for('identify_repetitions') }}">I can identify repetitions or loops in a sequence
    to shorten a set of instructions by repeating steps, for instance, when learning a new song.</a>
    </div>
    <div class="tab-pane fade" id="ps-step3" role="tabpanel" aria-labelledby="ps-step3-tab">
        <!-- Links for Progression Step 3 -->
        <i class="fas fa-cogs"></i> <a href="{{ url_for('refining_algorithms') }}">I can create and refine algorithms and flowcharts to
    solve problems, making use of features such as loops, Boolean values, and formulae.</a><br>
        <i class="fa-solid fa-sort"></i> <a href="{{ url_for('importance_of_order') }}">I can understand the importance of the order of state
    </div>
</div>
```

Figure 47: Code for tab content



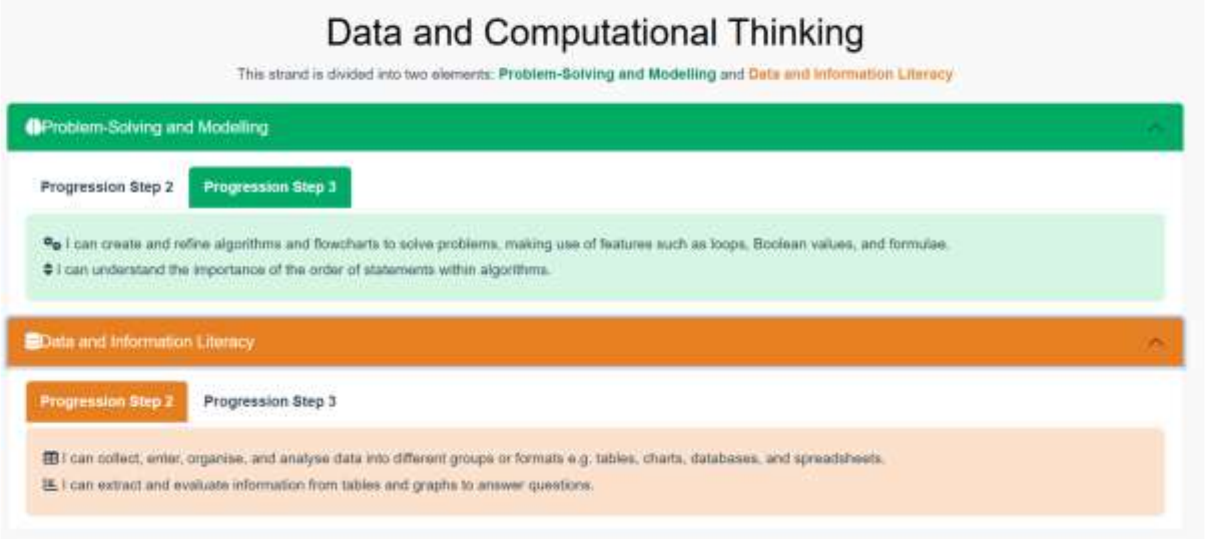
Data and Computational Thinking

I can collect, enter, organise, and analyse data into different groups or formats, e.g., tables, charts, databases, and spreadsheets.

I can extract and evaluate information from tables and graphs to answer questions.

Empowering educators in Wales to master the Digital Competency Framework

Figure 48: Tab navigation and content as displayed in browser



Data and Computational Thinking

This strand is divided into two elements: **Problem-Solving and Modelling** and **Data and Information Literacy**

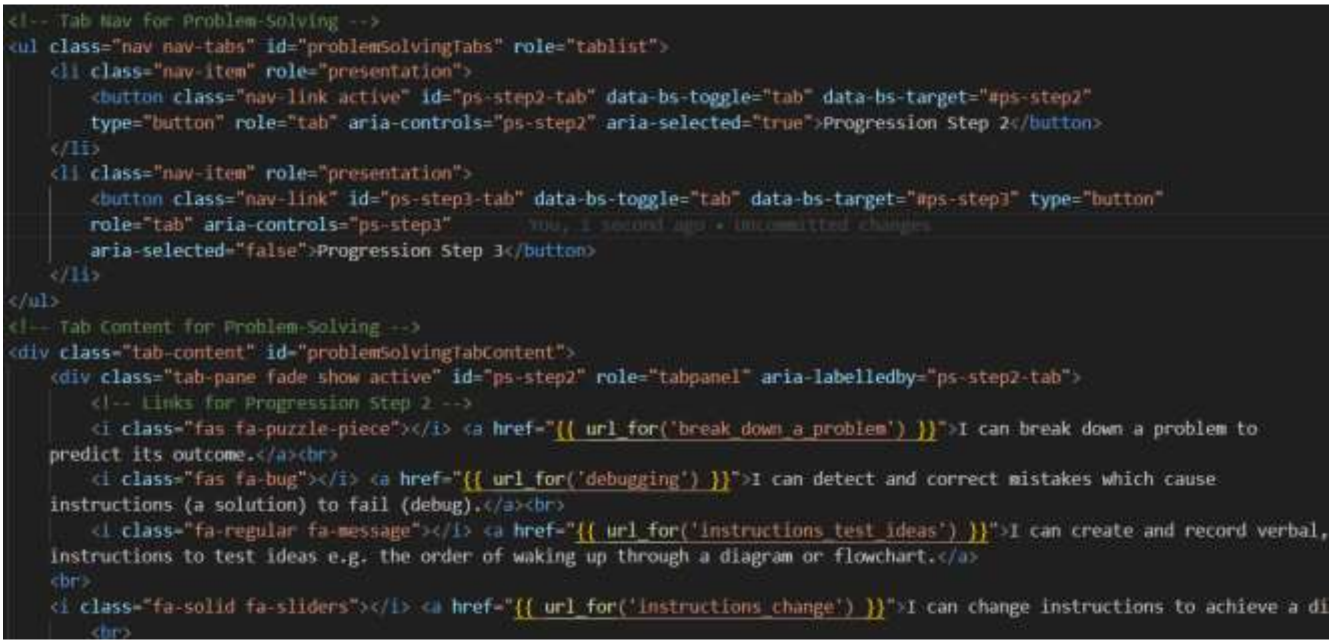
I can create and refine algorithms and flowcharts to solve problems, making use of features such as loops, Boolean values, and formulae.

I can understand the importance of the order of statements within algorithms.

I can collect, enter, organise, and analyse data into different groups or formats e.g. tables, charts, databases, and spreadsheets.

I can extract and evaluate information from tables and graphs to answer questions.

Figure 49: Tab navigation and content as displayed in browser



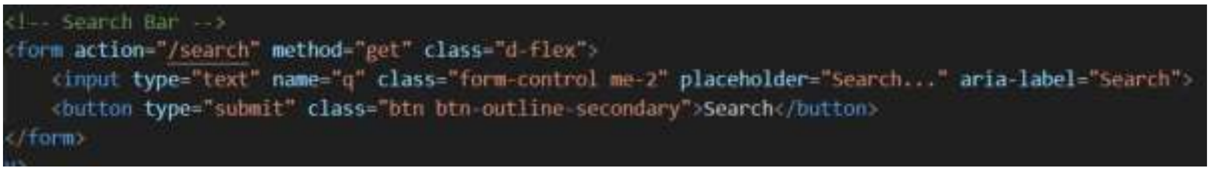
```
<!-- Tab Nav for Problem-Solving -->
<ul class="nav nav-tabs" id="problemSolvingTabs" role="tablist">
    <li class="nav-item" role="presentation">
        <button class="nav-link active" id="ps-step2-tab" data-bs-toggle="tab" data-bs-target="#ps-step2"
        type="button" role="tab" aria-controls="ps-step2" aria-selected="true">Progression Step 2</button>
    </li>
    <li class="nav-item" role="presentation">
        <button class="nav-link" id="ps-step3-tab" data-bs-toggle="tab" data-bs-target="#ps-step3" type="button"
        role="tab" aria-controls="ps-step3"        You, 1 second ago • Uncommitted changes
        aria-selected="false">Progression Step 3</button>
    </li>
</ul>
<!-- Tab Content for Problem-Solving -->
<div class="tab-content" id="problemSolvingTabContent">
    <div class="tab-pane fade show active" id="ps-step2" role="tabpanel" aria-labelledby="ps-step2-tab">
        <!-- Links for Progression Step 2 -->
        <i class="fas fa-puzzle-piece"></i> <a href="{{ url_for('break_down_a_problem') }}">I can break down a problem to
    predict its outcome.</a><br>
        <i class="fas fa-bug"></i> <a href="{{ url_for('debugging') }}">I can detect and correct mistakes which cause
    instructions (a solution) to fail (debug).</a><br>
        <i class="fa-regular fa-message"></i> <a href="{{ url_for('instructions_test_ideas') }}">I can create and record verbal,
    instructions to test ideas e.g. the order of waking up through a diagram or flowchart.</a>
    <br>
    <i class="fa-solid fa-sliders"></i> <a href="{{ url_for('instructions_change') }}">I can change instructions to achieve a di
        <br>
```

Figure 50: Code for tab navigation and content for Problem-Solving

Search Bar and Functionality

A lightweight search bar was implemented using Python dictionaries for content storage and retrieval. Python dictionaries provide a straightforward way to store and access data as key-value pairs and allowed for quick implementation which enabled rapid iteration and testing. This meant that the structure and organisation of the guide's content was crucial to enable effective search querying. While dictionaries are quick and easy to use for the prototype, they are limited in scalability. As I was only covering two progression steps, related to one strand of the DCF, this covers a small fraction of the entire curriculum, and scalability wasn't an issue for this project. This approach demonstrated the feasibility of search functionality with the potential for future database integration to enhance performance and scalability.

Search bar provides a form to search the website, sending the query to /search:



```
<!-- Search Bar -->
<form action="/search" method="get" class="d-flex">
    <input type="text" name="q" class="form-control me-2" placeholder="Search..." aria-label="Search">
    <button type="submit" class="btn btn-outline-secondary">Search</button>
</form>
```

Figure 51: Code for search bar

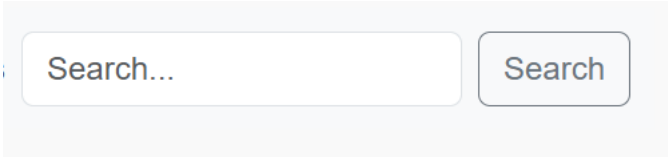
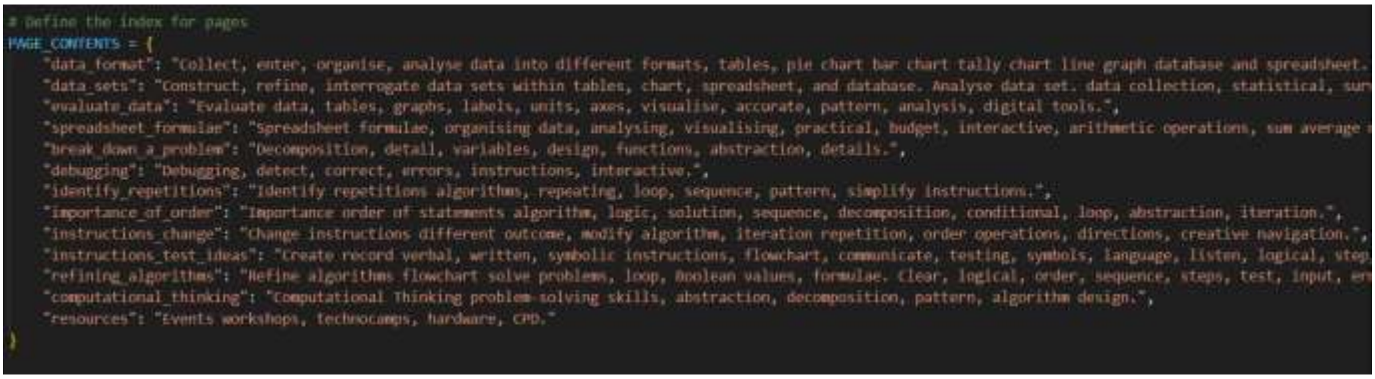


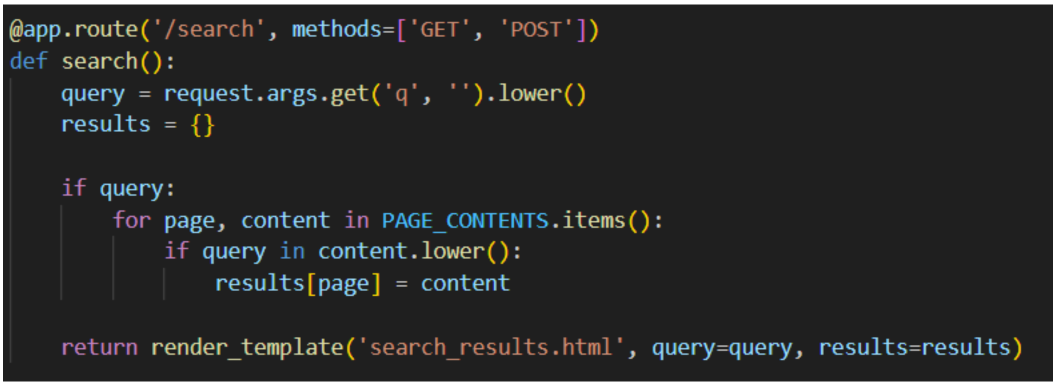
Figure 52: Search bar displayed in browser



```
# Define the index for pages
PAGE_CONTENTS = {
    "data_format": "Collect, enter, organise, analyse data into different formats, tables, pie chart bar chart tally chart line graph database and spreadsheet.
    "data_sets": "Construct, refine, interrogate data sets within tables, chart, spreadsheet, and database. Analyse data set. data collection, statistical, sur
    "evaluate_data": "Evaluate data, tables, graphs, labels, units, axes, visualise, accorate, pattern, analysis, digital tools.",
    "spreadsheet_formulae": "Spreadsheet formulae, organising data, analysing, visualising, practical, budget, interactive, arithmetic operations, sum average 
    "break_down_a_problem": "Decomposition, detail, variables, design, functions, abstraction, details.",
    "debugging": "Debugging, detect, correct, errors, instructions, interactive.",
    "identify_repetitions": "Identify repetitions algorithms, repeating, loop, sequence, pattern, simplify instructions.",
    "importance_of_order": "Importance order of statements algorithm, logic, solution, sequence, decomposition, conditional, loop, abstraction, iteration.",
    "instructions_change": "Change instructions different outcome, modify algorithm, iteration repetition, order operations, directions, creative navigation.",
    "instructions_test_ideas": "Create record verbal, written, symbolic instructions, flowchart, communicate, testing, symbols, language, listen, logical, step
    "refining_algorithms": "Refine algorithms flowchart solve problems, loop, Boolean values, formulae. Clear, logical, order, sequence, steps, test, input, en
    "computational_thinking": "Computational Thinking problem-solving skills, abstraction, decomposition, pattern, algorithm design.",
    "resources": "Events workshops, technocamps, hardware, CPD."
}
```

Figure 53: Code for defined index for content pages

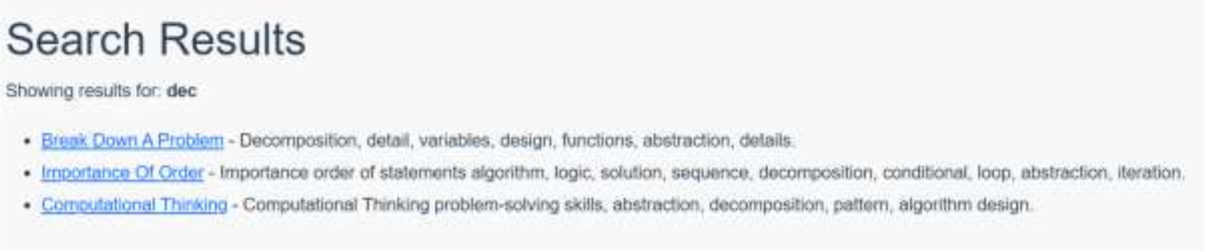
Search queries were matched against content stored in key-value pairs. Matching results were dynamically displayed on a results page.



```
@app.route('/search', methods=['GET', 'POST'])
def search():
    query = request.args.get('q', '').lower()
    results = {}

    if query:
        for page, content in PAGE_CONTENTS.items():
            if query in content.lower():
                results[page] = content

    return render_template('search_results.html', query=query, results=results)
```

Figure 54: Code for search functionality

Users can find results even when only part of a word matches the content:



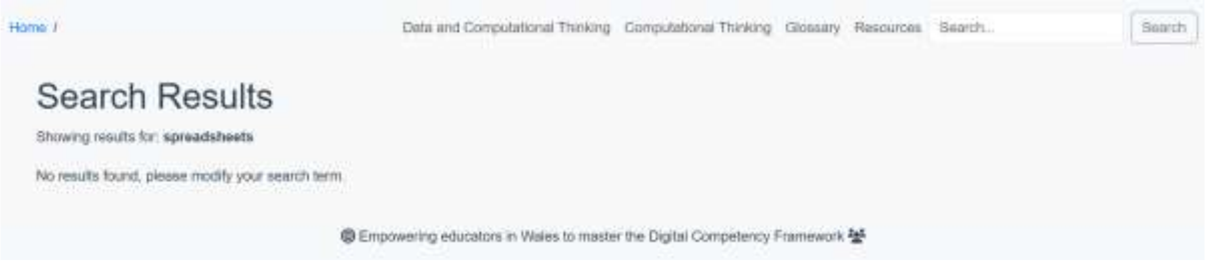
Search Results

Showing results for: **dec**

- Break Down A Problem - Decomposition, detail, variables, design, functions, abstraction, details.
- Importance Of Order - Importance order of statements algorithm, logic, solution, sequence, decomposition, conditional, loop, abstraction, iteration.
- Computational Thinking - Computational Thinking problem-solving skills, abstraction, decomposition, pattern, algorithm design.

Figure 55: Search Results displayed on browser using parts of a word match

If no results are found, users are prompted to refine their search. For example, there are no results for ‘spreadsheets’, however prompting a user to refine their search and there are results for ‘spreadsheet’. Results are dynamically displayed on the search results page, improving user experience.



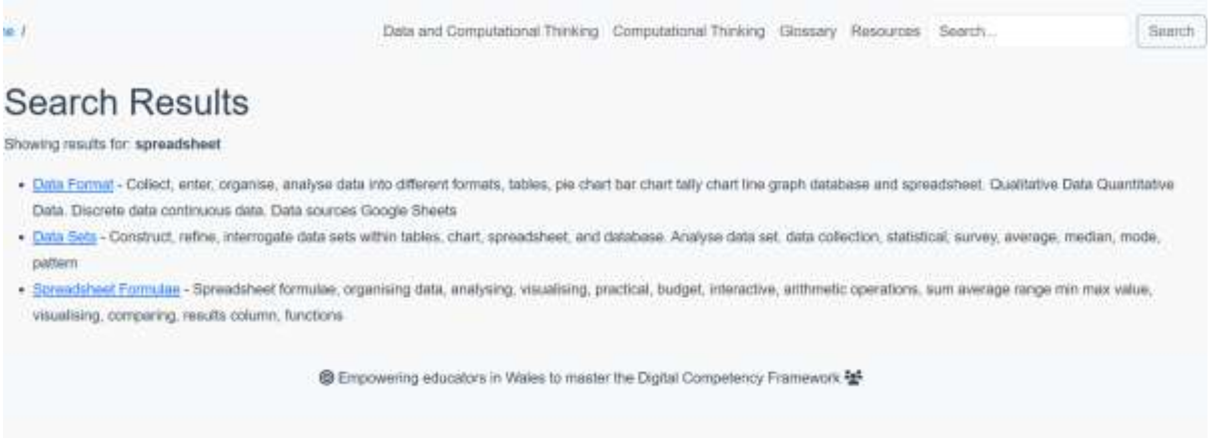
Search Results

Showing results for: **spreadsheets**

No results found, please modify your search term.

Empowering educators in Wales to master the Digital Competency Framework

Figure 56: No results displayed for Spreadsheets



Search Results

Showing results for: **spreadsheet**

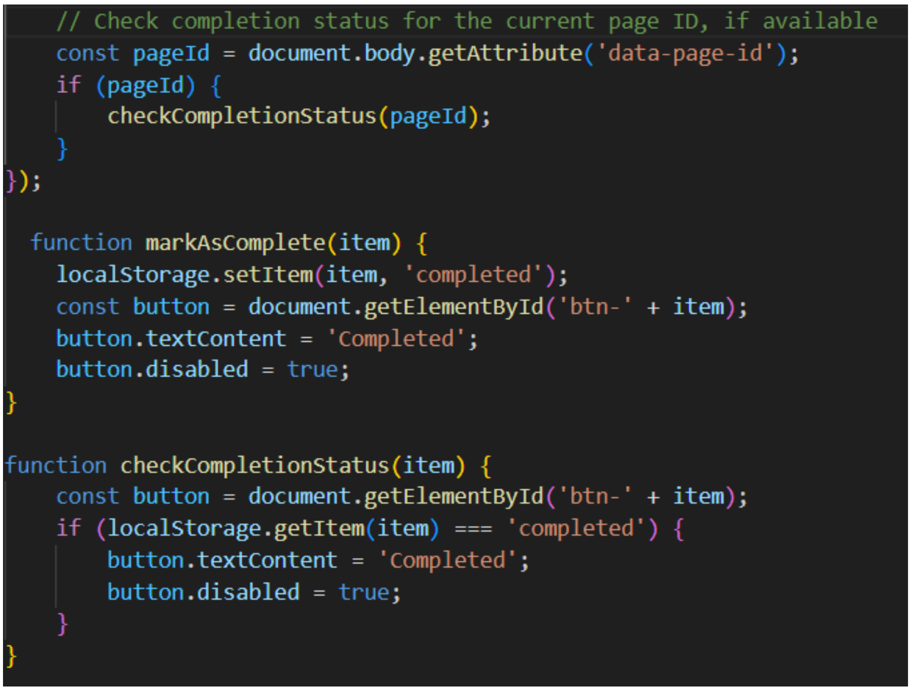
- Data Format - Collect, enter, organise, analyse data into different formats, tables, pie chart bar chart tally chart line graph database and spreadsheet. Qualitative Data Quantitative Data. Discrete data continuous data. Data sources Google Sheets
- Data Sets - Construct, refine, interrogate data sets within tables, chart, spreadsheet, and database. Analyse data set, data collection, statistical, survey, average, median, mode, pattern
- Spreadsheet Formulae - Spreadsheet formulae, organising data, analysing, visualising, practical, budget, interactive, arithmetic operations, sum average range min max value, visualising, comparing, results column, functions

Empowering educators in Wales to master the Digital Competency Framework

Figure 57: Results showing for Spreadsheet

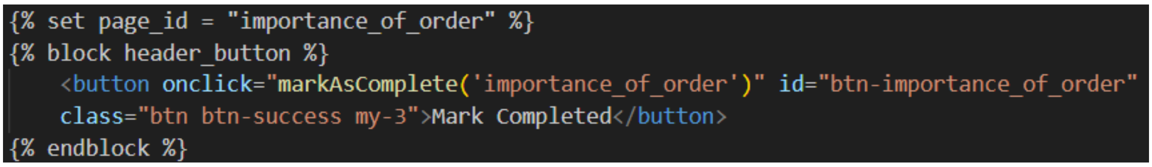
Tracking Progress “Mark as Completed”

Users can mark sections of the guide as completed using local storage. The “Mark Completed” button is provided for each Data and Computational Thinking content page, that stores page identifiers. When a user clicks the “Mark Completed” button, the corresponding page_id is stored in localStorage. On page load, the checkCompletionStatus function checks if the current page_id exists in local storage. If the section is marled as completed, the “Mark Completed” button is disabled to visually indicate “Completed” progress.



```
    // Check completion status for the current page ID, if available
    const pageId = document.body.getAttribute('data-page-id');
    if (pageId) {
        checkCompletionStatus(pageId);
    }
});

  function markAsComplete(item) {
    localStorage.setItem(item, 'completed');
    const button = document.getElementById('btn-' + item);
    button.textContent = 'Completed';
    button.disabled = true;
}

function checkCompletionStatus(item) {
    const button = document.getElementById('btn-' + item);
    if (localStorage.getItem(item) === 'completed') {
        button.textContent = 'Completed';
        button.disabled = true;
    }
}
```

Figure 58: Mark Completed function



```
{% set page_id = "importance_of_order" %}
{% block header_button %}
    <button onclick="markAsComplete('importance_of_order')" id="btn-importance_of_order"
    class="btn btn-success my-3">Mark Completed</button>
{% endblock %}
```

Figure 59: Template Integration

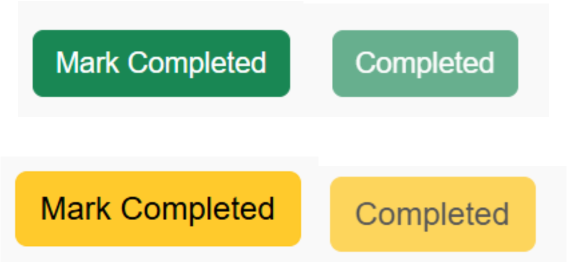
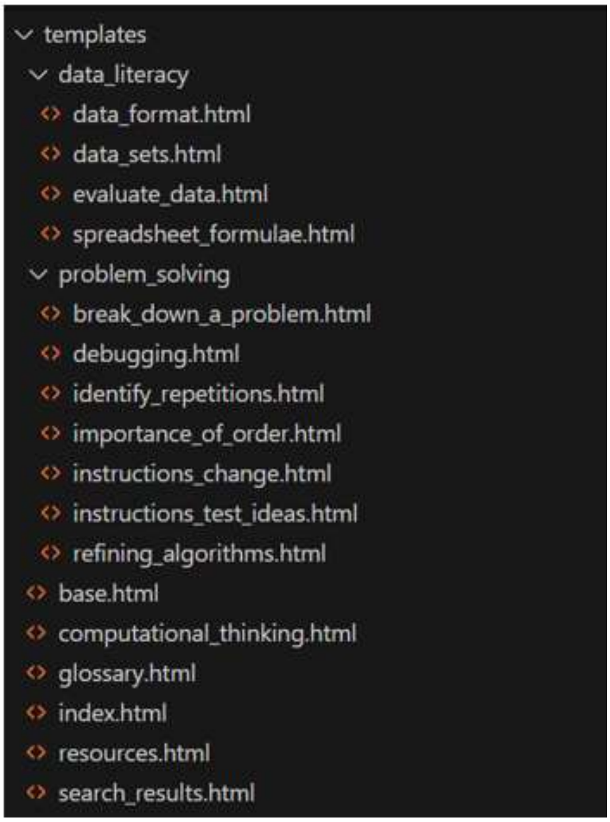


Figure 60: Mark Completed and Completed buttons displayed in browser

HTML Templates

The functions are all processed before a rendering of the changes is displayed within the template for the user to view. For all templates I ensured correct formatting was in place to enable clear readability and functionality.

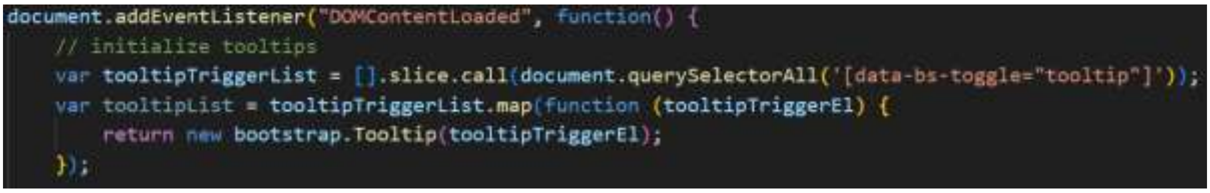


```
templates
  data_literacy
    data_format.html
    data_sets.html
    evaluate_data.html
    spreadsheet_formulae.html
  problem_solving
    break_down_a_problem.html
    debugging.html
    identify_repetitions.html
    importance_of_order.html
    instructions_change.html
    instructions_test_ideas.html
    refining_algorithms.html
  base.html
  computational_thinking.html
  glossary.html
  index.html
  resources.html
  search_results.html
```

Figure 61: Structure of templates folder

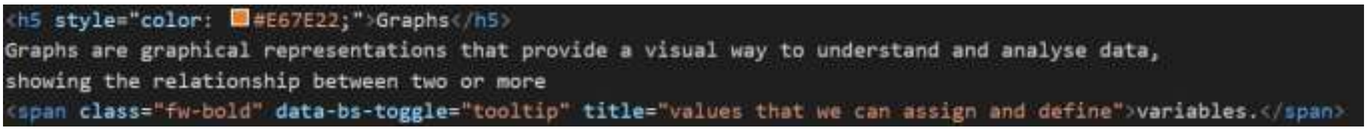
Tooltips

Tooltips provide users with contextual information as they navigate through the content. By hovering over specific terms, users can view additional explanations without leaving the current page, making the content more interactive and intuitive. Tooltips are dynamically initialised on page load using JavaScript. Bootstrap's tooltip component is used for consistent, responsive design.



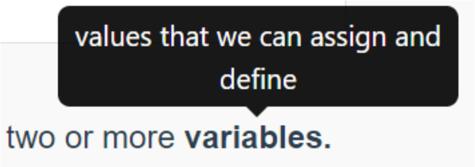
```
document.addEventListener("DOMContentLoaded", function() {
    // initialize tooltips
    var tooltipTriggerList = [].slice.call(document.querySelectorAll('[data-bs-toggle="tooltip"]'));
    var tooltipList = tooltipTriggerList.map(function (tooltipTriggerEl) {
        return new bootstrap.Tooltip(tooltipTriggerEl);
    });
```

Figure 62: Tooltip initialisation process in the code



```
<h5 style="color: 🟧#E67E22;">Graphs</h5>
Graphs are graphical representations that provide a visual way to understand and analyse data,
showing the relationship between two or more
<span class="fw-bold" data-bs-toggle="tooltip" title="values that we can assign and define">variables.</span>
```

Figure 63: Tooltip function code



values that we can assign and define

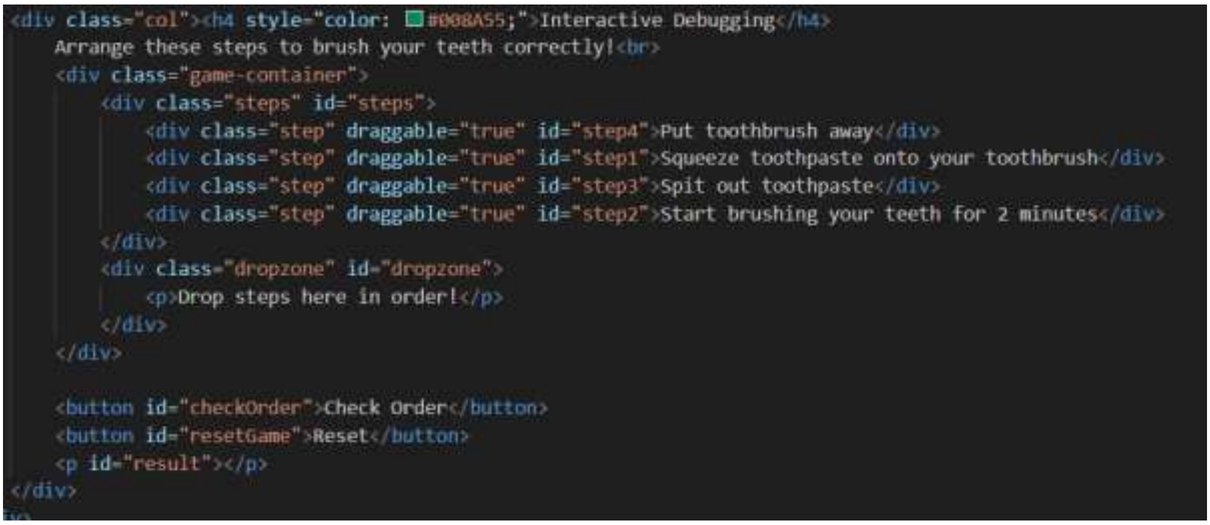
two or more **variables.**

Figure 64: Tooltip as displayed on the browser

Drag and Drop Functionality

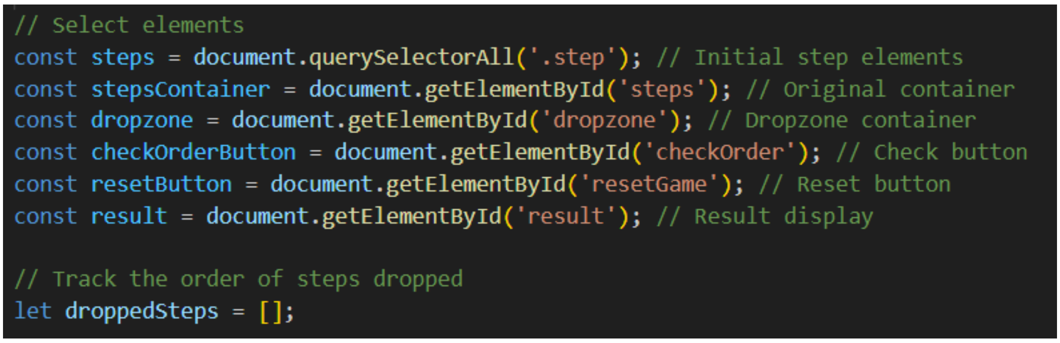
A drag-and-drop interface was implemented to teach sequential ordering and algorithm design. This feature enhanced user engagement by making abstract concepts more tangible. The functionality enabled users to reorder items in a drop zone, and items in the drop zone are draggable. Users can use the reset button to clear the drop zone, allowing start over again.

This section defines the game layout. It contains draggable steps and a container where users drop the steps in order.



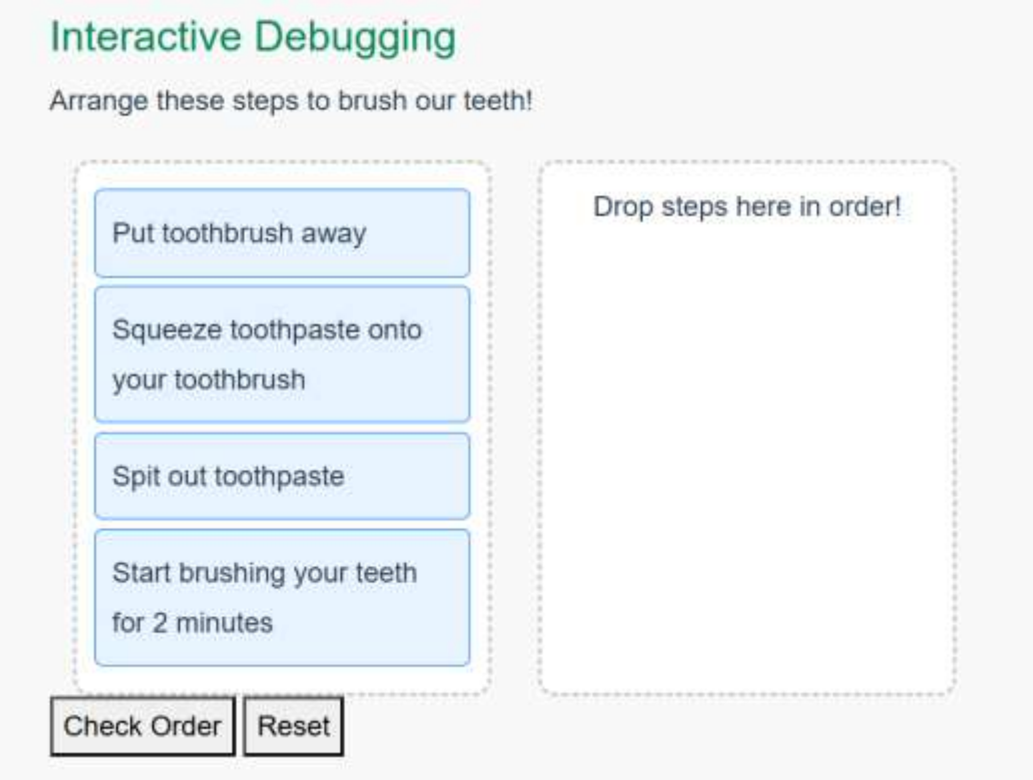
```
<div class="col"><h4 style="color: 🟩#008A55;">Interactive Debugging</h4>
    Arrange these steps to brush your teeth correctly!<br>
    <div class="game-container">
        <div class="steps" id="steps">
            <div class="step" draggable="true" id="step4">Put toothbrush away</div>
            <div class="step" draggable="true" id="step1">Squeeze toothpaste onto your toothbrush</div>
            <div class="step" draggable="true" id="step3">Spit out toothpaste</div>
            <div class="step" draggable="true" id="step2">Start brushing your teeth for 2 minutes</div>
        </div>
        <div class="dropzone" id="dropzone">
            <p>Drop steps here in order!</p>
        </div>
    </div>

    <button id="checkOrder">Check Order</button>
    <button id="resetGame">Reset</button>
    <p id="result"></p>
</div>
```

Figure 65: Code for game layout



```
// Select elements
const steps = document.querySelectorAll('.step'); // Initial step elements
const stepsContainer = document.getElementById('steps'); // Original container
const dropzone = document.getElementById('dropzone'); // Dropzone container
const checkOrderButton = document.getElementById('checkOrder'); // Check button
const resetButton = document.getElementById('resetGame'); // Reset button
const result = document.getElementById('result'); // Result display

// Track the order of steps dropped
let droppedSteps = [];
```

Figure 66: Defining elements to interact with



Interactive Debugging

Arrange these steps to brush our teeth!

Put toothbrush away

Squeeze toothpaste onto your toothbrush

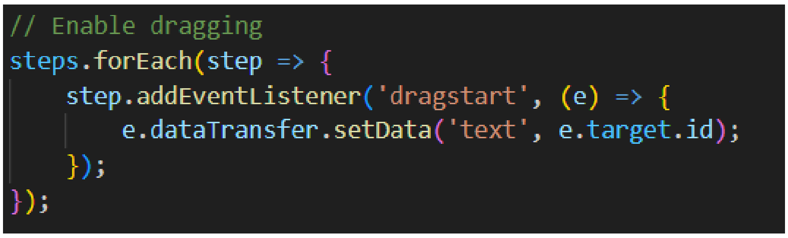
Spit out toothpaste

Start brushing your teeth for 2 minutes

Drop steps here in order!

Figure 67: Game displayed in browser

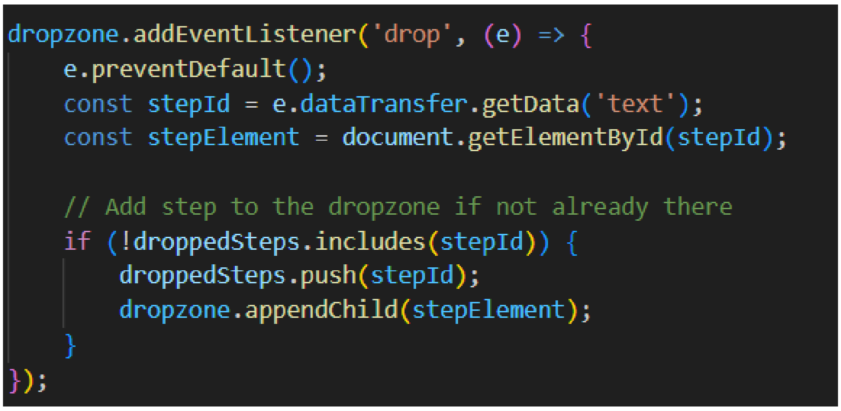
Drag Events



```
// Enable dragging
steps.forEach(step => {
    step.addEventListener('dragstart', (e) => {
        e.dataTransfer.setData('text', e.target.id);
    });
});
```

Figure 68: Attached the id of the dragged element, allowing identification when dropped

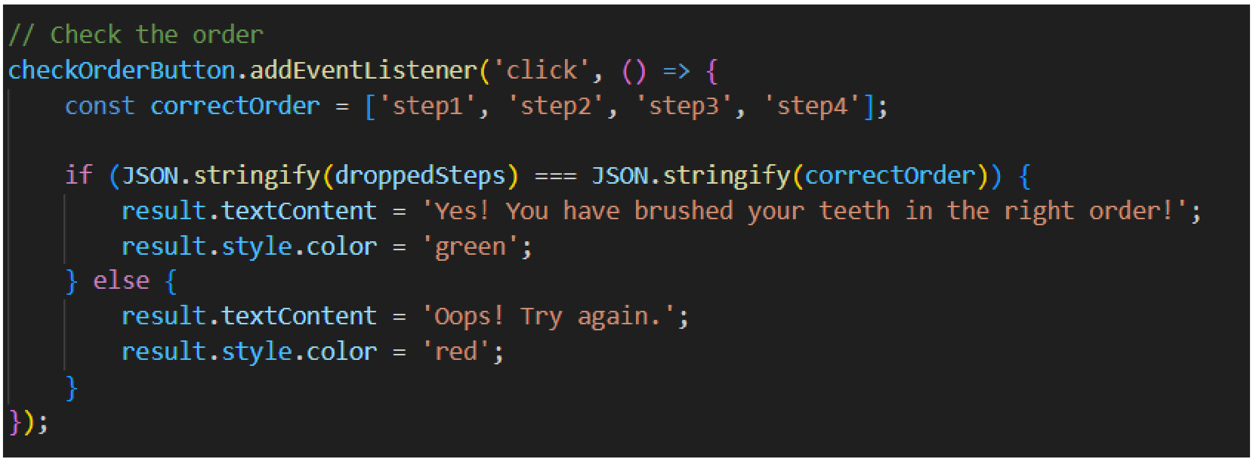
Drop zone Events



```
dropzone.addEventListener('drop', (e) => {
    e.preventDefault();
    const stepId = e.dataTransfer.getData('text');
    const stepElement = document.getElementById(stepId);

    // Add step to the dropzone if not already there
    if (!droppedSteps.includes(stepId)) {
        droppedSteps.push(stepId);
        dropzone.appendChild(stepElement);
    }
});
```

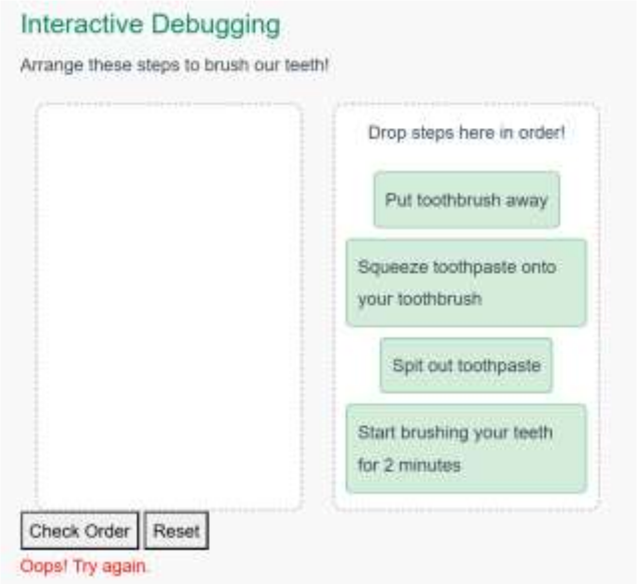
Figure 69: Tracks order of dropped steps in the droppedSteps array

Order Validation



```
// Check the order
checkOrderButton.addEventListener('click', () => {
    const correctOrder = ['step1', 'step2', 'step3', 'step4'];

    if (JSON.stringify(droppedSteps) === JSON.stringify(correctOrder)) {
        result.textContent = 'Yes! You have brushed your teeth in the right order!';
        result.style.color = 'green';
    } else {
        result.textContent = 'Oops! Try again.';
        result.style.color = 'red';
    }
});
```

Figure 70 checkOrderButton validates the dropped steps against the correct order, and then displays a success or failure message



Interactive Debugging

Arrange these steps to brush our teeth!

Drop steps here in order!

Put toothbrush away

Squeeze toothpaste onto your toothbrush

Spit out toothpaste

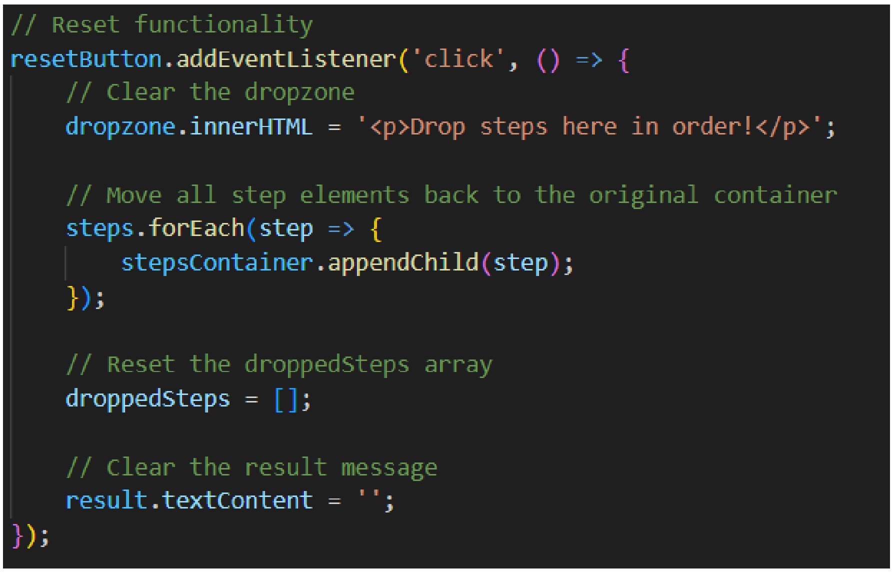
Start brushing your teeth for 2 minutes

Oops! Try again

Figure 71: Wrong order sequence as displayed in browser

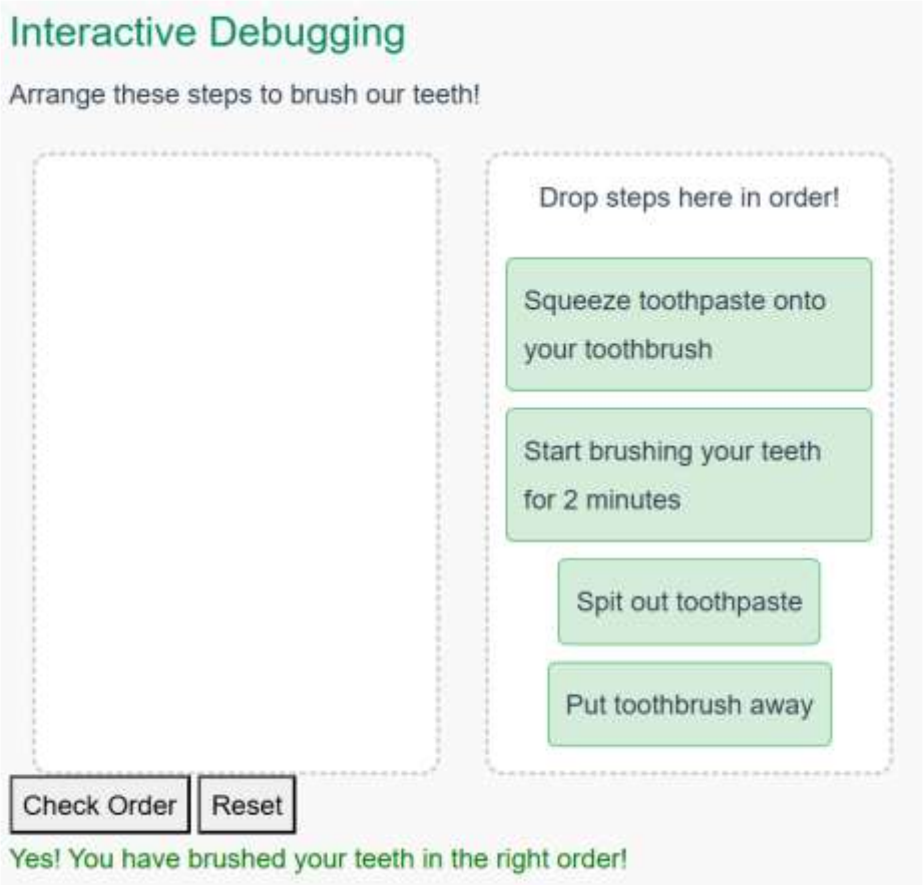
Reset Functionality

resetButton clears the drop zone, moves all steps back to their original container (stepsContainer), and resets the droppedSteps array and result message.



```
// Reset functionality
resetButton.addEventListener('click', () => {
    // Clear the dropzone
    dropzone.innerHTML = '<p>Drop steps here in order!</p>';

    // Move all step elements back to the original container
    steps.forEach(step => {
        stepsContainer.appendChild(step);
    });

    // Reset the droppedSteps array
    droppedSteps = [];

    // Clear the result message
    result.textContent = '';
});
```

Figure 72: Reset functionality code



Interactive Debugging

Arrange these steps to brush our teeth!

Drop steps here in order!
Squeeze toothpaste onto your toothbrush
Start brushing your teeth for 2 minutes
Spit out toothpaste
Put toothbrush away
Check Order
Reset

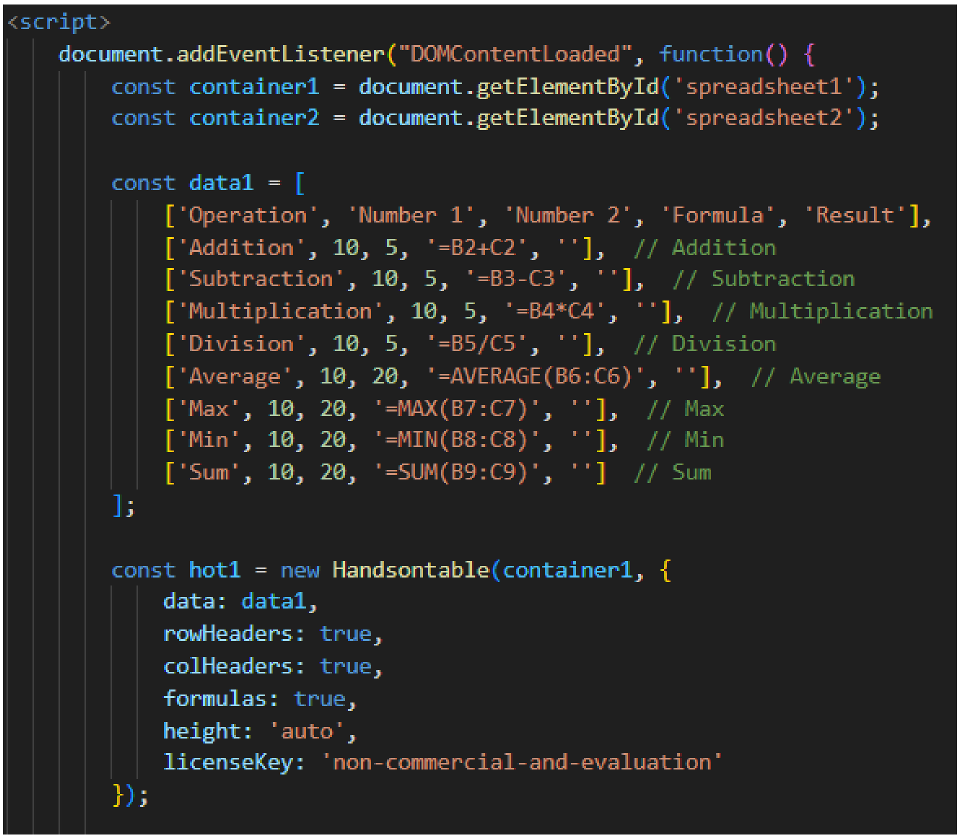
Yes! You have brushed your teeth in the right order!

Figure 73: Correct order sequence as displayed in browser

Interactive Spreadsheet with Handsontable Integration

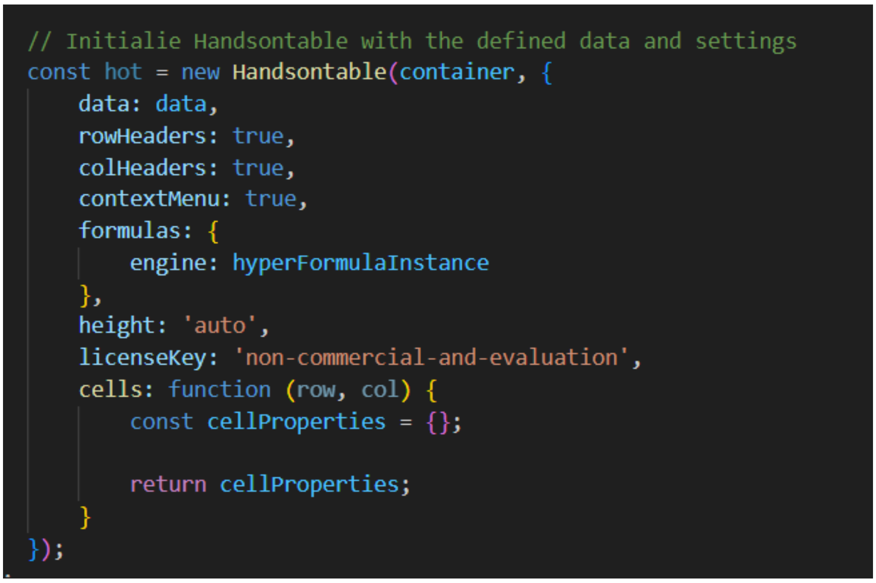
I used Handsontable, a JavaScript-based library, to implement provide interactive data manipulation exercises which enables users to explore spreadsheet functionalities directly within the web application. Using an embedded, interactive Handsontable spreadsheet, users can experiment with different values in the cells, with dynamic updates and real-time calculations within the spreadsheet to simulate practical spreadsheet usage. I placed this in a separate JavaScript file to maintain organisation and avoid any conflicts with other parts of the application.

The Result column dynamically calculates results using formulas written in the spreadsheet. This is enabled by HyperFormula engine. Users can double-click on cells to view or edit formulas directly. Upon reload the table is refreshed to default values.



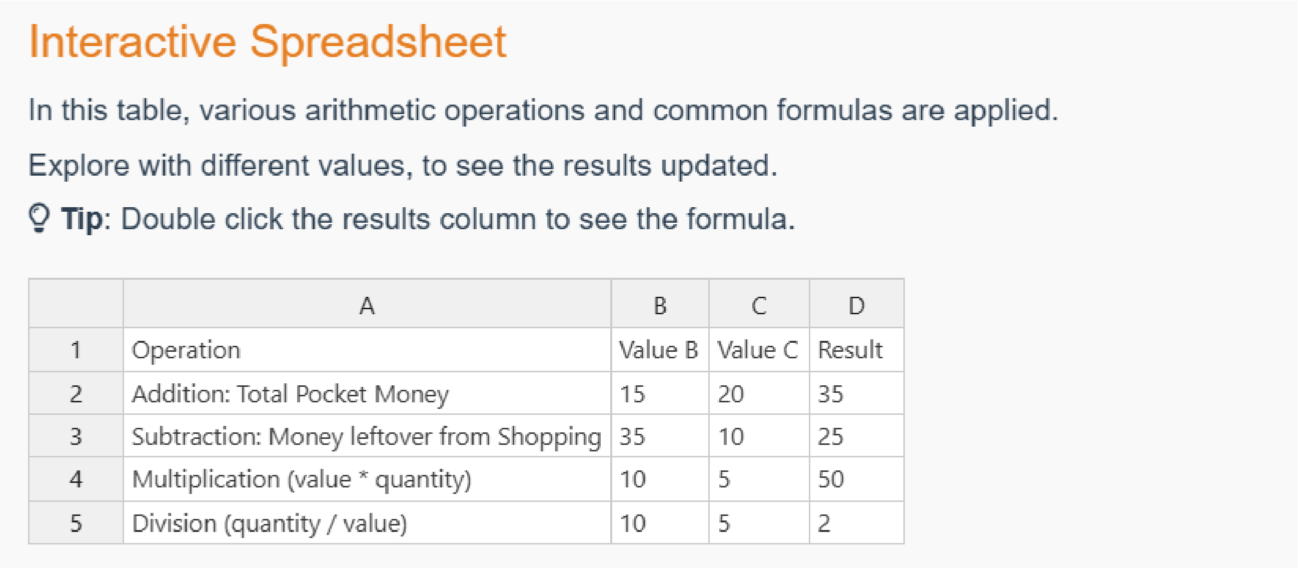
```
<script>
    document.addEventListener("DOMContentLoaded", function() {
        const container1 = document.getElementById('spreadsheet1');
        const container2 = document.getElementById('spreadsheet2');

        const data1 = [
            ['Operation', 'Number 1', 'Number 2', 'Formula', 'Result'],
            ['Addition', 10, 5, '=B2+C2', ''],  // Addition
            ['Subtraction', 10, 5, '=B3-C3', ''],  // Subtraction
            ['Multiplication', 10, 5, '=B4*C4', ''],  // Multiplication
            ['Division', 10, 5, '=B5/C5', ''],  // Division
            ['Average', 10, 20, '=AVERAGE(B6:C6)', ''],  // Average
            ['Max', 10, 20, '=MAX(B7:C7)', ''],  // Max
            ['Min', 10, 20, '=MIN(B8:C8)', ''],  // Min
            ['Sum', 10, 20, '=SUM(B9:C9)', '']  // Sum
        ];

        const hot1 = new Handsontable(container1, {
            data: data1,
            rowHeaders: true,
            colHeaders: true,
            formulas: true,
            height: 'auto',
            licenseKey: 'non-commercial-and-evaluation'
        });
```

Figure 74: Handsontable integration code



```
// Initialie Handsontable with the defined data and settings
const hot = new Handsontable(container, {
    data: data,
    rowHeaders: true,
    colHeaders: true,
    contextMenu: true,
    formulas: {
        engine: hyperFormulaInstance
    },
    height: 'auto',
    licenseKey: 'non-commercial-and-evaluation',
    cells: function (row, col) {
        const cellProperties = {};

        return cellProperties;
    }
});
```

Figure 75: hyperFormulaInstance implemented for dynamic results



Interactive Spreadsheet

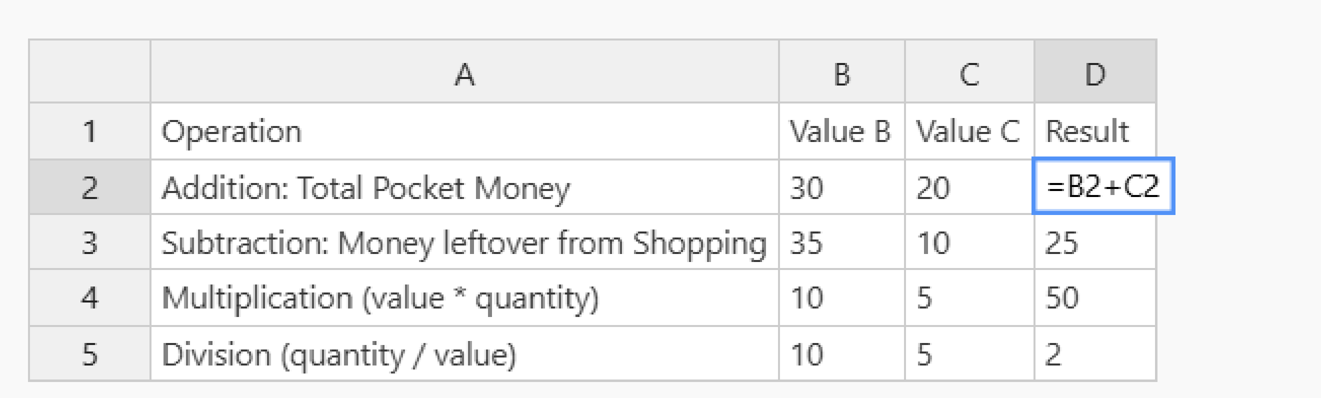
In this table, various arithmetic operations and common formulas are applied.

Explore with different values, to see the results updated.

💡 **Tip**: Double click the results column to see the formula.

	A	B	C	D
1	Operation	Value B	Value C	Result
2	Addition: Total Pocket Money	15	20	35
3	Subtraction: Money leftover from Shopping	35	10	25
4	Multiplication (value * quantity)	10	5	50
5	Division (quantity / value)	10	5	2

Figure 76: Interactive Spreadsheet displayed in browser

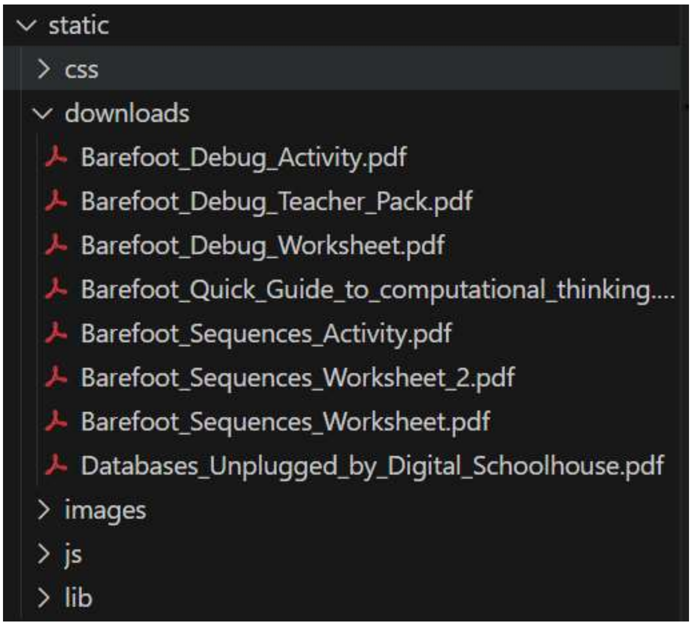


	A	B	C	D
1	Operation	Value B	Value C	Result
2	Addition: Total Pocket Money	30	20	=B2+C2
3	Subtraction: Money leftover from Shopping	35	10	25
4	Multiplication (value * quantity)	10	5	50
5	Division (quantity / value)	10	5	2

Figure 77 Demonstrating use of formulae in Result column. Values can be modified

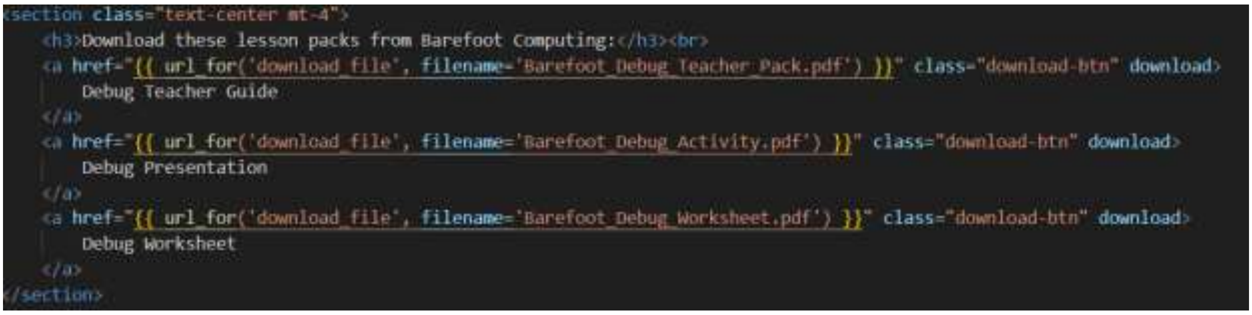
Security

All links to download pdf files, I utilised Flask's URL to generate links for downloads. This enhances security by preventing direct URL manipulations and ensures that users access documents through the intended links only, reinforcing the integrity of the download process. The function to retrieve these files are filtered to only include those with a '.pdf' extension.



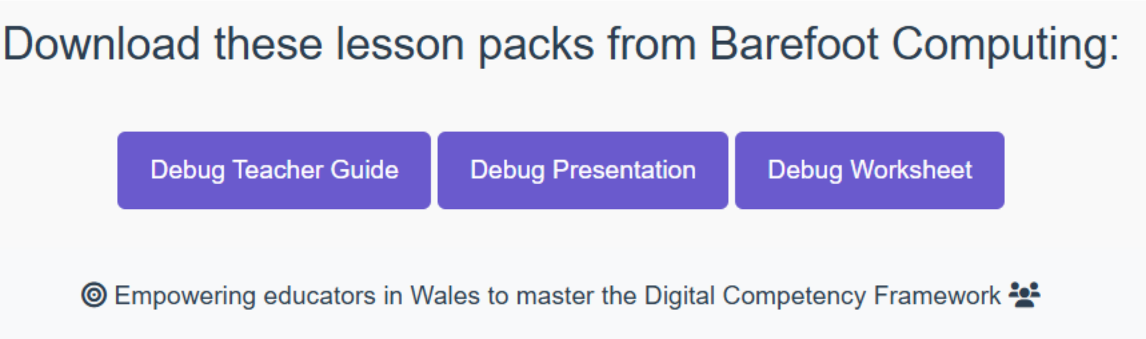
```
∨ static
  > css
  ∨ downloads
      Barefoot_Debug_Activity.pdf
      Barefoot_Debug_Teacher_Pack.pdf
      Barefoot_Debug_Worksheet.pdf
      Barefoot_Quick_Guide_to_computational_thinking....
      Barefoot_Sequences_Activity.pdf
      Barefoot_Sequences_Worksheet_2.pdf
      Barefoot_Sequences_Worksheet.pdf
      Databases_Unplugged_by_Digital_Schoolhouse.pdf
  > images
  > js
  > lib
```

Figure 78: Static folder structure showing downloads sub-folder



```
<section class="text-center mt-4">
    <h3>Download these lesson packs from Barefoot Computing:</h3><br>
    <a href="{{ url_for('download_file', filename='Barefoot_Debug_Teacher_Pack.pdf') }}" class="download-btn" download>
        Debug Teacher Guide
    </a>
    <a href="{{ url_for('download_file', filename='Barefoot_Debug_Activity.pdf') }}" class="download-btn" download>
        Debug Presentation
    </a>
    <a href="{{ url_for('download_file', filename='Barefoot_Debug_Worksheet.pdf') }}" class="download-btn" download>
        Debug Worksheet
    </a>
</section>
```

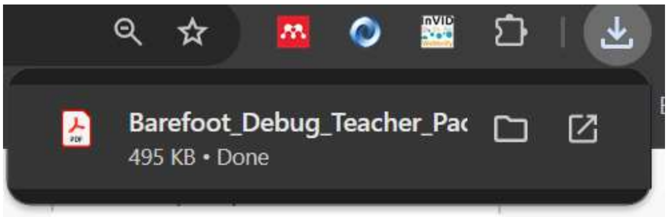
Figure 79: Code to securely download pdfs from static folder



Download these lesson packs from Barefoot Computing:

◎ Empowering educators in Wales to master the Digital Competency Framework

Figure 80: Download buttons displayed on browser



Barefoot_Debug_Teacher_Pac
495 KB • Done

Figure 81: Evidence of successfully downloaded pdfs from web application

Web Development Best Practices

Throughout implementation phases I intended to follow the industry standard best practices for web development. To avoid large-scale errors and issues with the project not functioning properly, I avoided deploying major changes and instead chose to implement smaller elements at a time to ensure the code was functional. The use of GitHub for version control was essential here, as well as backing up local versions to OneDrive. There were few occasions where the code caused problems but by using version control to my advantage, I was easily able to revert my changes and return to a working version of the project. For code quality I followed the software principles of Keep It Simple Stupid and developed code that was reusable where necessary. With many of the features sharing similar functionality, it was simple to reuse a great deal of my code, not only saving myself time in the development process, but making the entire project more unified.

Testing

Regular testing was conducted throughout development, which allowed me to identify and fix bugs, ensuring that functionality aligned with the design. Through implementing incremental changes, this minimised bugs, and I utilised version control for reverting problematic changes. Integration testing verified the interaction between components such as dynamic navigation and search functionality and manual testing identified usability issues which were addressed iteratively.

Conclusion

The implementation phase successfully delivered a functional protype that meets the core requirements while maintaining flexibility for future expansion. By leveraging Flask, Jinja2, and Bootstrap, the project met its requirements while remaining adaptable for future enhancements. Every page of the application is displayed in [Appendix 2].

Evaluation

This chapter evaluates the outcome of the project by assessing the different objectives set out at various stages the of the project.

MVP Revisited

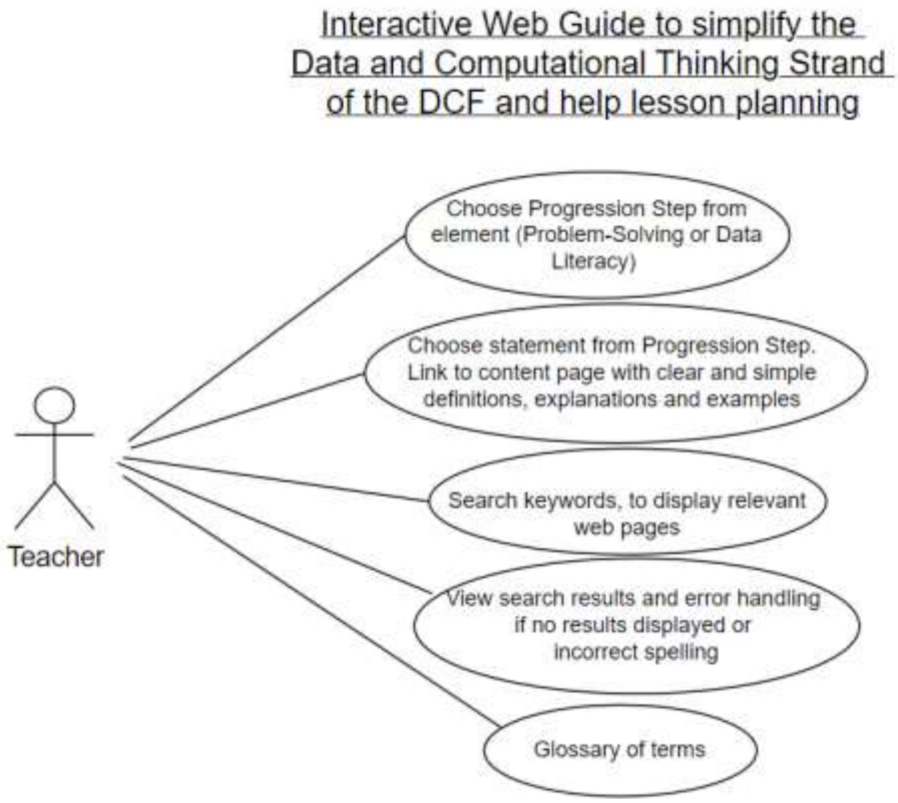

Interactive Web Guide to simplify the Data and Computational Thinking Strand of the DCF and help lesson planning
Choose Progression Step from element (Problem-Solving or Data Literacy)
Choose statement from Progression Step. Link to content page with clear and simple definitions, explanations and examples
Search keywords, to display relevant web pages
View search results and error handling if no results displayed or incorrect spelling
Glossary of terms
Teacher

Figure 82: Minimum Viable Product

Evaluating the objectives set out in the MVP from the Requirements phase, the final product satisfies each objective. The guide's layout ensures that the content for each statement, in each progression step, in each element is clearly displayed. A functional search bar is present on the top of each page, displaying search results, with prompts to handle misspellings or incorrect keywords. The guide contains a dedicated glossary page that contains key terminology that is used throughout the guide.

Evaluating against Acceptance Criteria

The following criteria define the necessary outcomes for the web guide to meet user needs:

Table 2: Evaluating against Acceptance Criteria

Acceptance Criteria	Met / Not Met with Explanation
The web guide should be aligned with the specific expectations and learning objectives of the DCF data and computational thinking strand, as set out in the DCF.	**Met** – every learning objective of the DCT strand is accounted for with its own page in the guide.
The web guide should provide clear and detailed explanations of data and computational thinking skills and concepts.	**Met** – every skill statement is clearly explained, with additional resources referenced for further learning.
The web guide should contain examples of activities for lesson plans to bring concepts to the classroom.	**Met** – every page includes examples and activities, transforming abstract concepts to the classroom.
The web guide should be easy to navigate, and users can easily find information for the topic and progression step that they choose.	**Met** – no page on the guide goes further than 2 layers deep, content is clearly presented, a search bar is enabled for specific keywords.
The web guide should have elements of interactivity to enhance user experience.	**Met** – interactive features include drag and drop activity, interactive spreadsheet, plus other features.
The web guide should contain resources that are relevant to the curriculum with real World examples.	**Met** – resources have been carefully chosen to reflect the curriculum and be relatable to real World examples.

Evaluation against Requirements

Functional Requirements

Table 3: Evaluating against Functional Requirements

Lesson Planning Support Educators can select a specific Progression Step, and the web guide provides a list of relevant learning activities and resources aligned with the selected Progression Step.	Met – content is organised by Progression Step
DCF Integration The DCT is clearly represented within the web guide, and users can easily navigate between Progression Steps and their associated elements. Users can filter or search for activities and resources based on specific DCF elements – skills, knowledge areas, keywords - making it easy to find relevant content for different teaching goals.	Met – search bar
Definitions for Key Terminology Key terminology to be defined and recorded in one place in the form of a Glossary, for easy navigation.	Met – Glossary page, tooltips
Progress Tracking Users can mark sections as complete when finished with the resource.	Met – local storage tracking
Welsh Language Implementation The web application is available in both English and Welsh, with all content accurately translated.	Not Met

Non-Functional Requirements

Table 4: Evaluating against Non-Functional Requirements

Usability The interface is intuitive and user-friendly, and easy to understand with clear instructions. The web guide provides clear visual cues and tooltips to assist users. There is consistent design and styling throughout the web application, with a navigation bar that allows users to easily find what they are looking for.	Met
Error Handling The web guide provides clear error messages and guidance if users encounter issues, ensuring users are redirected and they can resolve problems independently.	Met
Security All documents to be downloaded from the guide will be stored in static folder, making sure that anything downloaded from the web application is safe.	Met

Evaluation against initial Project Objectives

Finally, I will evaluate the project against the objectives established at the onset of this project, from the Introduction chapter. To summarise, I can confidently state that the overarching aim of "*developing an interactive, user-friendly, web-based guide, that simplifies the DCF, provides resources and ideas for lesson plans, to support primary school teachers in Wales, in effectively implementing the DCF into their lesson planning",* has been achieved. I conducted a thorough review of the DCF, researching many points of view and different sources of information, to gain a well-rounded perspective of the challenges faced in implementing the DCF. I gained understanding in the changes that educators have faced in recent years, especially in autonomy regarding curriculum design. Teachers have endured a culture shift in their profession in Wales, from being "passive consumers" enacting a curriculum that is laid out for them, to "curriculum designers" (Evans, G. 2022). Prior assumptions I had regarding challenges in implementing the DCF were solidified though survey responses. These user requirements informed the guide's content; however, it has already been addressed, that the scope of the guide was narrowed and did not contain every requirement addressed.

The design of the guide is intuitive and easy to navigate and incorporates various interactive features for user engagement. Resources have been taken from commonly used platforms, in an effort to consolidate readily available information in one place. Terminology and has been defined and simplified and contained in a dedicated Glossary page. All of the skill statements incorporate practical classroom activities. In terms of evaluating against the Data and Computational Thinking strand, the project can be marked as a success.

Future Work

To further enhance the web guide's functionality and impact, the following areas for future development have been identified:

Expanding the Scope

When I started this project, I was expecting to create a guide that covered the DCF in its entirety. I recognised this was a mammoth task given the projects timescale; however, my initial plans included this scope in its entirety. It wasn't until I conducted deeper research and analysed primary survey findings that I decided to narrow the scope of this project to the Data and Computational Thinking strand. I have highlighted the area of the diagram that is complete. While I am satisfied with the work I have produced, future work would expand content coverage to provide comprehensive support for educators across the DCF.

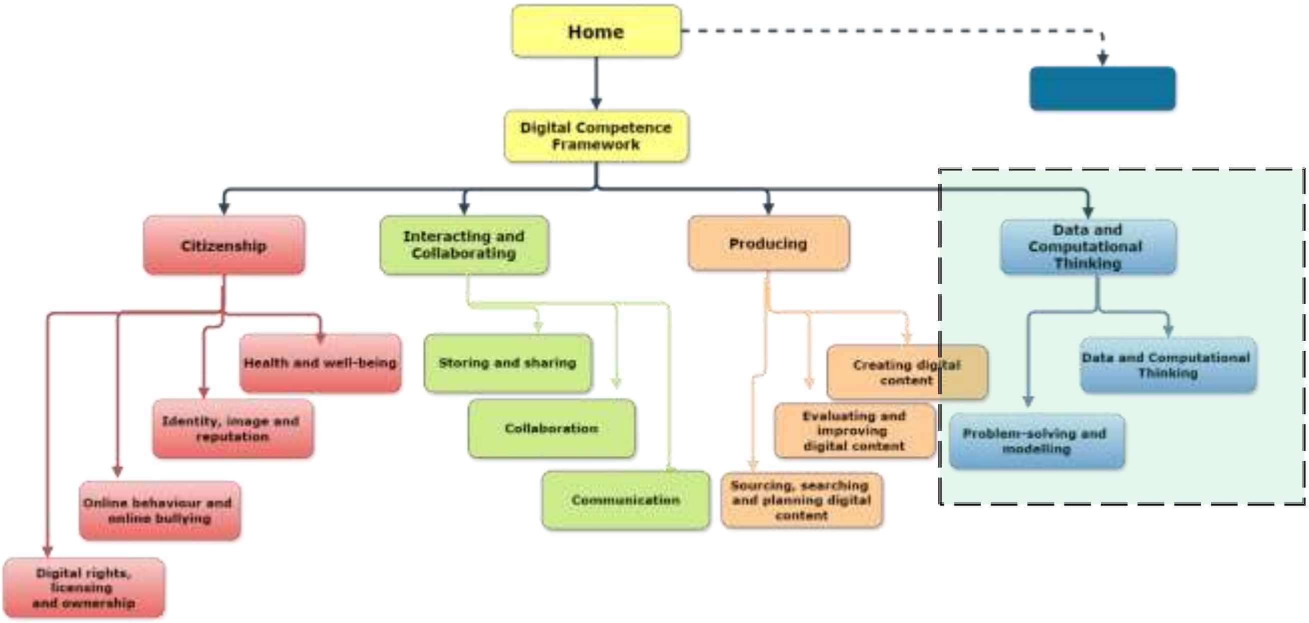

Home
Digital Competence Framework
Citizenship
Interacting and Collaborating
Producing
Data and Computational Thinking
Health and well-being
Identity, image and reputation
Online behaviour and online bullying
Digital rights, licensing and ownership
Storing and sharing
Collaboration
Communication
Creating digital content
Evaluating and improving digital content
Sourcing, searching and planning digital content
Data and Computational Thinking
Problem-solving and modelling

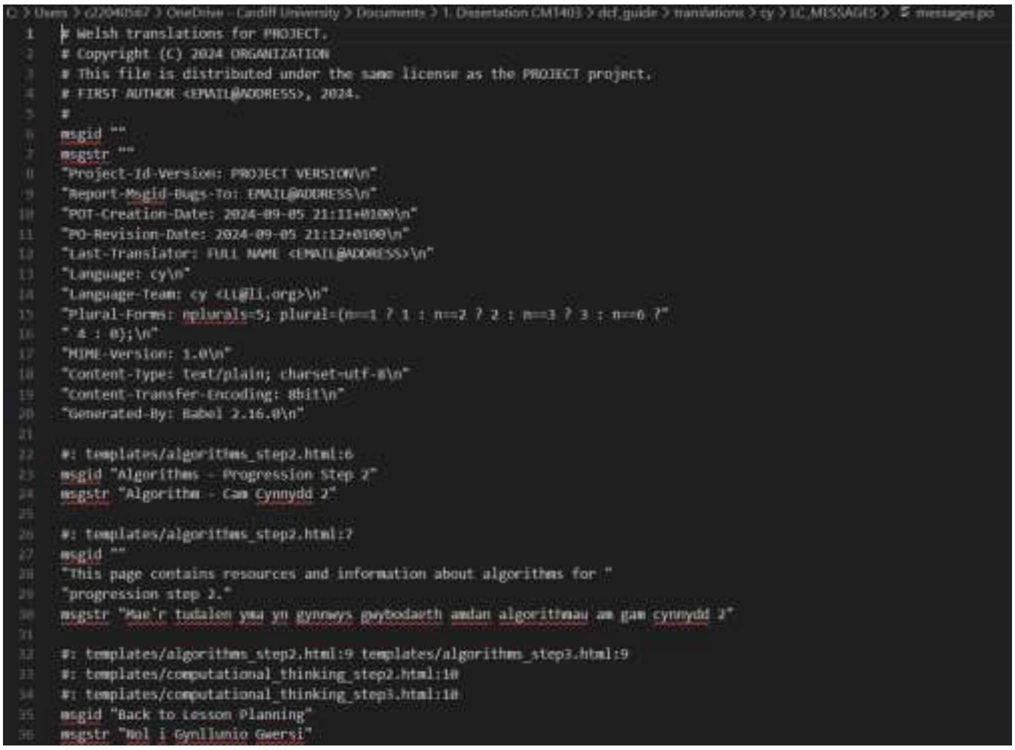
Figure 83: Initial Web Architecture with highlighted section showing what's already been completed

Content expansion would not only cover all four strands of the DCF but would ensure that elements of all six AoLEs are incorporated in examples and lesson activities. Further to this, the guide would incorporate Routes for Learning (RfL) to better support learners with PMLD ensuring inclusivity and adaptability for diverse educational needs.

Enhancing User Experience and Functionality

Welsh language version implementation

Implementing a Welsh language version using tools like Flask-Babel to foster inclusivity and accessibility. While research into using Flask-Babel was conducted, the bilingual feature was not successfully implemented within the project’s timeframe. Future iterations would incorporate a toggle button which could switch seamlessly between English and Welsh language. This would be beneficial not only for fluent Welsh speakers but also for Welsh language learners who could see instantly the translations and make connections between words and meanings. Further to this, even fluent Welsh language speakers can struggle with the Welsh translations of technical terminology. Having the ease to switch between languages allows quicker understanding of terms. This feature would benefit fluent Welsh speakers, Welsh learners and those navigating technical terminology in Welsh, fostering better understanding and accessibility across linguistic groups.


C: > Users > c22040567 > OneDrive - Cardiff University > Documents > 1. Dissertation CM1403 > dcf_guide > translations > cy > LC_MESSAGES > messages.po

```
# Welsh translations for PROJECT.
# Copyright (C) 2024 ORGANIZATION
# This file is distributed under the same license as the PROJECT project.
# FIRST AUTHOR <EMAIL@ADDRESS>, 2024.
#
msgid ""
msgstr ""
"Project-Id-Version: PROJECT VERSION\n"
"Report-Msgid-Bugs-To: EMAIL@ADDRESS\n"
"POT-Creation-Date: 2024-09-05 21:11+0100\n"
"PO-Revision-Date: 2024-09-05 21:12+0100\n"
"Last-Translator: FULL NAME <EMAIL@ADDRESS>\n"
"Language: cy\n"
"Language-Team: cy <LL@li.org>\n"
"Plural-Forms: nplurals=5; plural=(n==1 ? 1 : n==2 ? 2 : n==3 ? 3 : n==6 ?"
" 4 : 0);\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=utf-8\n"
"Content-Transfer-Encoding: 8bit\n"
"Generated-By: Babel 2.16.0\n"

#: templates/algorithms_step2.html:6
msgid "Algorithms - Progression Step 2"
msgstr "Algorithm - Cam Cynnydd 2"

#: templates/algorithms_step2.html:7
msgid ""
"This page contains resources and information about algorithms for "
"progression step 2."
msgstr "Mae'r tudalen yma yn gynnwys gwybodaeth amdan algorithmau am gam cynnydd 2"

#: templates/algorithms_step2.html:9 templates/algorithms_step3.html:9
#: templates/computational_thinking_step2.html:10
#: templates/computational_thinking_step3.html:10
msgid "Back to Lesson Planning"
msgstr "Nol i Gynllunio Gwersi"
```

Figure 84: Snippet of Flask-Babel

Community Features for Collaboration

The new curriculum is designed to empower and give control to teachers, and for this to be reflected in the interactive web guide. The importance of working collaboratively has been a consistent theme in the implementation of the DCF in the curriculum. Digital Pioneer schools have been set up for this purpose, and to further foster this theme of collaboration, in future iterations the guide would include a discussion forum where educators can share ideas, provide feedback, and collaborate on lesson planning. This would foster a sense of community among teachers and encourage collaboration and feedback where teachers can share ideas, ask questions and support each other in implementing the DCF using their experiences.

Hardware Integration resources

Providing resources on using different hardware in the classroom such as Raspberry Pi, Micro:bits and Lego Spike. There is a need for continuous professional development for teachers, while ensuring equitable access to technology for all learners. Micro:bit is a good tool to use for this as it is online simulation available if the physical hardware is not available ensuring that pupils from deprived areas receive the same teaching experience.

Enhanced Interactivity

Adding quizzes and assessments to test user understanding of concepts and enhance user engagement.

Improved Scalability

With further expansion, migrating content to a database will enable better data management, advanced search capabilities and optional user account registration for personalised experiences.

Personal Reflection

This reflection assesses the skills and knowledge gained during the development of this dissertation project. Throughout the project, I encountered various challenges and learning opportunities that have contributed to my personal and professional development.

To begin with, I had limited knowledge of the curriculum in Wales or its recent changes regarding digital literacy. Through extensive research and creative thinking, I successfully addressed every skill statement outlined in the DCT strand.

This project significantly enhanced my confidence and development. It made me appreciate how far I have come from the beginning of my masters, from having no prior experience with web development and limited digital competency myself. This conversion course has exposed new terminology and concepts that I have had to quickly grasp and understand. This project has been a significant learning experience, contributing to both my personal and professional growth. Through countless hours of designing, coding and problem-solving, I have honed my technical skills in areas such as routing, rendering templates and creating interactive web features. Additionally, beyond technical development, the project demanded extensive research into the Welsh curriculum and a thorough understanding of educators needs, to ensure that the guide was both practical, relevant and impactful.

I attempted to implement a Welsh language translation using Flask-Babel following Miguel Grinberg's online guide (Grinberg, M. 2013) but ultimately, with the time constraints, I felt it was too insecure to include in my project. Flask babel involved manually translating messages in .mo and .po. As a Welsh speaker this wasn't an issue for me however, there are many computational terms that don't have a fitting translation, which can prove challenging and to be confusing across Welsh speakers.

The final project is the result of numerous iterations. I have developed a deeper understanding of the importance of digital skills in education and the varying levels of technical skills among educators.

The most time-consuming aspect of the project, and one of the biggest challenges was designing and refining the lesson pages. With eleven pages to create, I aimed to avoid redundancy while ensuring content is relevant to the DCF statements. Additionally, I prioritised using relatable, real-World examples for learners.

One of the biggest challenges I faced was determining the appropriate content for the guide. After reviewing various sources and lesson plans, I gained a general understanding of learner progression in steps 2 and 3. However, I was concerned that some terminology and concepts, particularly those related to computing and programming, might be too advanced for certain age groups. Ultimately, I decided to trust my judgment and provide educators with slightly more advanced information. This approach allows educators to tailor their lessons to the specific needs and abilities of their students.

The challenges encountered during the project have strengthened my time management and creative problem-solving skills. I am grateful for the opportunity to apply my knowledge and skills to a current and real-world problem and contribute to the ongoing development of digital education in Wales.

In conclusion, this dissertation project has been a valuable learning experience. It has shaped my approach to future projects and my overall professional development.

References

- AltexSoft. 2023. *Acceptance Criteria for User Stories in Agile: Purposes, Formats, Examples, and Best Practices.* Available at: https://www.altexsoft.com/blog/acceptance-criteria-purposes-formats-and-bestpractices/ [Accessed: 1 October 2024].
- Atlassian. 2024. *Waterfall Methodology: A Comprehensive Guide.* Available at: https://www.atlassian.com/agile/project-management/waterfall-methodology [Accessed: 10 October 2024].
- Atlassian. 2024. *Agile Project Management – What is it and how to get started?* Available at: https://www.atlassian.com/agile/project-management [Accessed: 10 October 2024].
- BBC. 2015. *Digital skills in schools as 'vital' as pupil literacy*. Available at: https://www.bbc.co.uk/news/uk-wales-33080589 [Accessed: 2 July 2024].
- Carroll, F. *et al.* (2023). *The journey to making "Digital Technology" education a Community Learning Venture. Education Sciences*, 13(5), p. 428. doi:10.3390/educsci13050428.
- Donaldson, G. (2015). *Successful Futures, Independent Review of Curriculum and Assessment Arrangements in Wales:* Crown Copyright.
- Estyn. 2016. *Digital technology in primary school life.* Available at: https://estyn.gov.wales/improvement-resources/digital-technology-in-primary-school-life/ [Accessed: 1 September 2024].
- Estyn. 2018. *Preparing for the Digital Competence Framework (DCF).* Crown Copyright. Available at: www.estyn.gov.wales [Accessed: 1 September 2024].
- Estyn. 2019. *Digital competence: Improving pupils' digital skills.* Available at: https://estyn.gov.wales/blog/digital-competence-improving-pupils-digital-skills/ [Accessed: 10 August 2024].
- Evans, G. *Back to the future? Reflections on three phases of education policy reform in Wales and their implications for teachers*. J Educ Change 23, 371–396 (2022). https://doi.org/10.1007/s10833-021-09422-6
- Grinberg, M. 2013. *The flask mega-tutorial, part XIV: I18n and L10N (2012), miguelgrinberg.com.* Available at: https://blog.miguelgrinberg.com/post/the-

flask-mega-tutorial-part-xiv-i18n-and-l10n-legacy [Accessed: 28 September 2024].

- Lewis, H. 2015. *Written Statement - Announcement of Digital Pioneer Schools* , *Written Statement - Announcement of Digital Pioneer Schools* Available at: https://www.gov.wales/written-statement-announcement-digital-pioneer-schools [Accessed: 10 June 2024].
- MozDevNet. 2024. *The web and web standards - learn web development: MDN, MDN Web Docs.* Available at: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/The_web_and_web_standards [Accessed: 17 September 2024].
- Nania, J. (2019) et al *No Longer Optional: Employer Demand for Digital Skills*
- Pencoed Primary School. 2024. *The Digital Competence Framework*. Available at: https://www.pencoedprimary.co.uk/the-digital-competence-framework-dcf/#:~:text=The%20DCF%20teaches%20children%20how,and%20enhance%20their%20computational%20thinking [Accessed: 1 September 2024].
- St Patricks RC. 2024*. Curriculum for Wales 2022.* Available at: https://www.stpatricks-primary.co.uk/curriculum-for-wales-2022/ [Accessed: 1 September 2024].
- Technocamps. 2024. *Annual Report.* Available at: https://www.technocamps.com/en/about-technocamps/annual-report/ [Accessed: 1 July 2024].
- Technocamps. 2024. *Technoteach.* Available at: https://www.technocamps.com/en/technoteach/ [Accessed: 1 September 2024].
- Welsh Government. 2015. *Successful Futures Looking at the Curriculum and Assessment Arrangements in Wales. A summary of Professor Graham Donaldson's report.*
- Welsh Government. 2018a. *Digital Competence Framework Guidance*
- Welsh Government. 2018b. *Digital Competence Framework. Your questions answered.* Digital ISBN 978 1 78903 888 0. Available at: gov.wales/curriculumforwales

- Welsh Government. 2022. *Cross-curricular skills frameworks.* Available at: https://hwb.gov.wales/curriculum-for-wales/cross-curricular-skills-frameworks/#digital-competence-framework [Accessed: 22 June 2024].
- Welsh Government. *Digital Competence Framework* (2020) *Hwb*. Available at: https://hwb.gov.wales/curriculum-for-wales/cross-curricular-skills-frameworks/digital-competence-framework/ [Accessed: 25 June 2024].
- Wiegers, K.E. and Beatty, J., 2013. *Software requirements*. Pearson Education.

Bibliography

- BBC. 2015. *School shake-up within eight years, says education minister.* Available at: https://www.bbc.co.uk/news/uk-wales-33087859 [Accessed: 2 July 2024].
- BBC Bitesize. 2024. 'Computing'. Available at: https://www.bbc.co.uk/bitesize/subjects/zvnrq6f [Accessed: 20 August 2024].
- Burns, J. 2015. *Digital skills should be core subjects, says report.* Available at: https://www.bbc.co.uk/news/education-31501917#:~:text=Children%20should%20be%20taught%20%22digital,Lords%20Digital%20Skills%20Committee%20warns. [Accessed: 4 July 2024].
- Dauncey, M. 2015. *Donaldson Review: The 'purposes' and content of a Curriculum for Wales.* Available at: https://research.senedd.wales/research-articles/donaldson-review-the-purposes-and-content-of-a-curriculum-for-wales/#:~:text=The%20four%20purposes%20are%20that,of%20Wales%20and%20the%20world [Accessed: 2 August 2024].
- Digital Schoolhouse. 2020. *Resources*. Available at: https://www.digitalschoolhouse.org.uk/resources [Accessed: 25 June 2024].
- Gatley, J. 2020. *Can the New Welsh Curriculum achieve its purposes?* Available at: https://bera-journals.onlinelibrary.wiley.com/doi/epdf/10.1002/curj.26?src=getftr [Accessed: 5 July 2024].
- Interaction Design Foundation. No date. *Wireframes.* Available at: https://www.interaction-design.org/literature/topics/wireframing [Accessed: 20 July 2024].

- Mark Otto, J.T. (no date) *Bootstrap*, *Bootstrap · The most popular HTML, CSS, and JS library in the world.* Available at: https://getbootstrap.com/ [Accessed: 27 June 2024].
- McCavitt. 2021. *How Wales brings digital literacy to every student in the country.* Available at: https://blog.adobe.com/en/publish/2021/08/02/how-wales-brings-digital-literacy-every-student-country#:~:text=The%20Welsh%20Government%20introduced%20a,and%20math%20in%20the%20classroom. [Accessed: 18 July 2024].
- Meace S., Carroll F., & Kop R. 2022. '*Promoting Teacher Confidence With Technology Through Risk Taking and Organisational Changes: A Welsh Perspective', The Barcelona Conference on Education 2022: Official Conference Proceedings* (pp. 269-281) doi.org/10.22492/issn.2435-9467.2022.23
- Misheva, G. 2021. *The Digital Competence Framework (digcomp)*, *Digital Skills and Jobs Platform*. Available at: https://digital-skills-jobs.europa.eu/en/actions/european-initiatives/digital-competence-framework-digcomp [Accessed: 25 June 2024].
- Moller, F. 2020. *Technocamps: Igniting computation in the new curriculum*, *Education Wales*. Available at: https://educationwales.blog.gov.wales/2020/08/05/technocamps-igniting-computation-in-the-new-curriculum/ [Accessed: 25 June 2024].
- Pallets. 2010. *Welcome to Flask, Welcome to Flask - Flask Documentation.* Available at: https://flask.palletsprojects.com/en/stable/ [Accessed: 28 June 2024].
- Stem Learning No date. *KS2 Algorithms.* Available at: https://www.stem.org.uk/resources/community/collection/359714/ks2-algorithms [Accessed: 19 July 2024].
- Web Dev Students. 2023. *Frameworks and Bootstrap Columns.* Available at: https://webdevstudents.com/modules/module-11/ [Accessed: 20 July 2024].
- Welsh Government, 2015*. Written Statement - Announcement of Digital Pioneer Schools.* Available at: https://www.gov.wales/written-statement-announcement-digital-pioneer-schools [Accessed: 2 July 2024].

- Welsh Government. 2017. *Barefoot boost for computing science in Welsh primary schools.* Available at: https://www.gov.wales/barefoot-boost-computing-science-welsh-primary-schools [Accessed: 2 July 2024].
- Welsh Government. 2020. *Technocamps: Igniting Computation in the New Curriculum.* Available at: https://educationwales.blog.gov.wales/2020/08/05/technocamps-igniting-computation-in-the-new-curriculum/ [Accessed: 2 July 2024].
- Welsh Government. 2023. *Planning for Progression*. Available at: https://sites.google.com/hwbcymru.net/planningforprogression/home [Accessed: 26 July 2024].

Appendix 1

Survey of Teacher's Digital Skills

This survey is intended for educators in primary schools in Wales only

Elinor Morris

MorrisE23@cardiff.ac.uk

Section One

Introduction (Participation Information Sheet)

You are being invited to take part in a research project. Before you decide whether or not to take part, it is important for you to understand why the research is being undertaken and what it will involve. Please take your time to read the following information carefully and discuss it with others, if you wish.

This is a shortened version of the information sheet, to read the full information sheet please follow this link:

CMT403_ComputingDissertation_Ethics_PIS_C22040567_v1.docx (sharepoint.com)

https://cf.sharepoint.com/:w:/r/teams/ElinorsPlan/Shared Documents/Ethics + Survey/CMT403_ComputingDissertation_Ethics_PIS_C22040567_v1.docx?d=w86b04acfe51f4f8cacb9051d425aeb74&csf=1&web=1&e=0INXUX

1. What is the purpose of this research project?

This student research project is part of the Cardiff University CMT403 Computing Dissertation module. The project aims to improve the digital skills of primary school educators, to better equip and enable the implementation of the Digital Competency Framework in the new Welsh curriculum. The information collected from this survey will be used to gain insight into user's preferences and to develop a set of functional requirements for a website.

2. Why have I been invited to take part?

You have been invited because you have experience of working in primary schools in Wales. Your insight as an educator in primary school is invaluable to the design of this project, to create a piece that is useful and valuable in your lesson planning and digital literacy.

3. Do I have to take part?

No, your participation in this research project is entirely voluntary and it is up to you to decide whether or not to take part. If you decide to take part, you must complete a consent form provided on the following page. If you decide not to take part, you do not have to explain your reasons and it will not affect your legal rights. You are free to withdraw your consent to participate in the research project at any time, without giving a reason, even after completing the consent form.

4. What will taking part involve?

There are a series of questions that address demographics, digital skills and experiences in classrooms. Please complete the survey as accurately as possible, it will take about 7 minutes.

5. Further information and contact details:

Should you have any questions relating to this research project, please contact me via email:

Elinor Morris

MorrisE23@cardiff.ac.uk

Thank you for reading this

Consent Form

Please answer all questions

1. I confirm that I have read the information sheet for the above research project.

Yes

2. I confirm that I have understood the information sheet and that I have had the opportunity to ask questions and that these have been answered satisfactorily.

Yes

3. I understand that my participation is voluntary, and I am free to withdraw at any time without giving a reason.

Yes

4. I understand that this study is part of a module and anonymised data I provide may be included as part of coursework.

Yes

5. I agree to take part in this research project.

Yes, I agree

Demographic Information

6. What is your age range?

18 - 25 years

26 - 35 years

36 - 45 years

46 - 55 years

56 - 65 years

66+ years

7. What is the highest level of education you've completed?

Further Education i.e. A-Levels, BTEC etc.

Undergraduate degree

Postgraduate qualification incl. PGCE

Professional Experience

8. How long have you been teaching?

One year or less

1 - 5 years

6 - 10 years

11 - 20 years

21 years +

9. What year group have you most recently been teaching?

Year 3

Year 4

Years 3 + 4

Year 5

Year 6

Years 5 + 6

10. Have you received any professional development training on digital skills in the past year? If yes, please describe briefly

11. How would you describe your computer proficiency?

Proficient

Advanced

Intermediate

Basic

Fundamental

12. How would you rate your overall confidence with using digital tools in your classroom

Very Confident

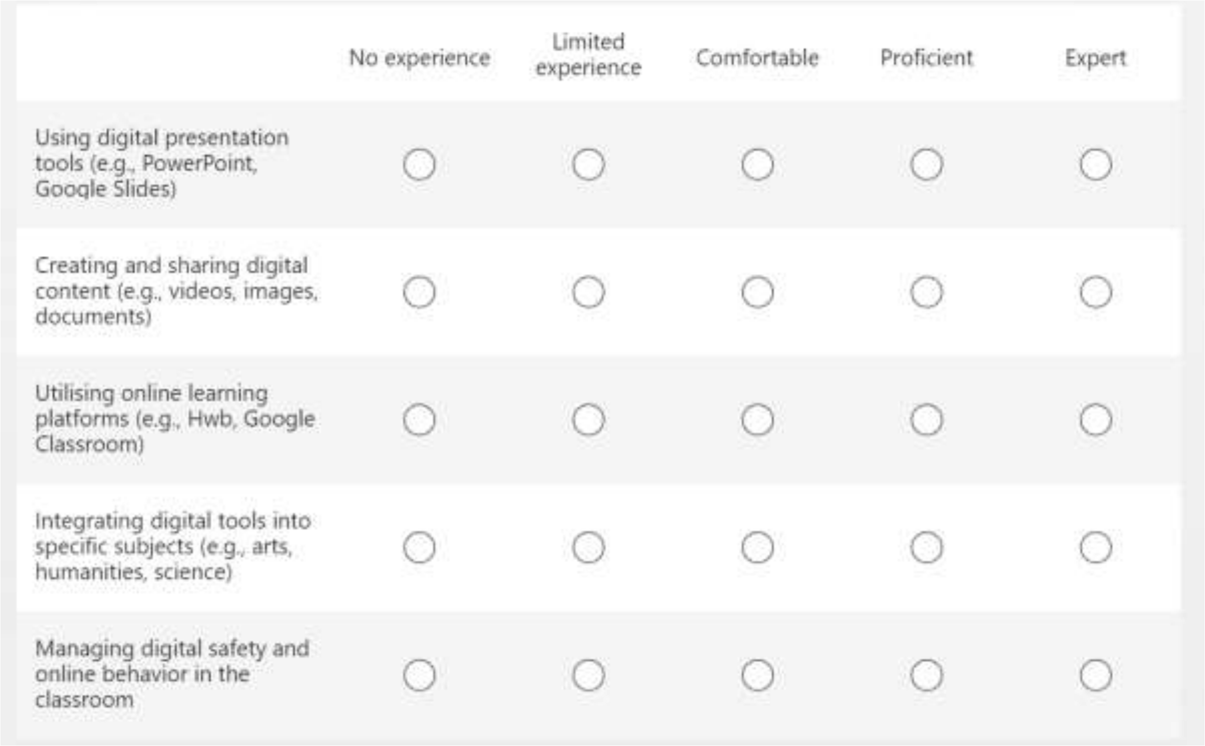
Confident to use new equipment and find answers independently

With training happy to use

Prefer not to use new tools, causes anxiety

Not at all confident

13. Please rate your current level of skill in each of the following areas:



	No experience	Limited experience	Comfortable	Proficient	Expert
Using digital presentation tools (e.g., PowerPoint, Google Slides)	○	○	○	○	○
Creating and sharing digital content (e.g., videos, images, documents)	○	○	○	○	○
Utilising online learning platforms (e.g., Hwb, Google Classroom)	○	○	○	○	○
Integrating digital tools into specific subjects (e.g., arts, humanities, science)	○	○	○	○	○
Managing digital safety and online behavior in the classroom	○	○	○	○	○

14. What are the main challenges you face when trying to integrate digital tools into your teaching? (Select all that apply)

Lack of time for planning and preparation

Insufficient training or support

Difficulty finding relevant and age-appropriate resources

Concerns about digital safety and online behaviour

Other

15. What types of digital skills would you like to develop further to enhance your teaching? (Select all that apply)

Creating engaging and interactive lessons across the curriculum

Incorporating coding and computational thinking

Troubleshooting basic technical issues

Staying up to date with educational technology trends

Other

Needs and Challenges

16. What features would you find most helpful in an interactive guide for learning digital skills? (Select all that apply)

Step-by-step tutorials and video demonstrations

Lesson plans and activities aligned with the Welsh curriculum

Opportunities for self-paced learning and reflection

Interactive quizzes and assessments to check understanding

A teaching community forum for sharing ideas and asking questions

Other

Welsh Curriculum and DCF

17.How familiar are you with the Digital Competence Framework?

0 1 2 3 4 5 6 7 8 9 10

Not at all Familiar
Very Familiar

18. Do you understand the terminology used in the DCF?

Yes, all of it is clear to me

Most of it, some new terms I have had to learn

I find it confusing but after asking others and searching the Hwb I have better understanding

Not really, most of it is not familiar to me

Not at all, I really struggle

19. Are there any phrases or terms that you don’t fully understand? If yes, please provide details

20. Do you understand the terminology used in the Area of Learning of Science and Technology?

Yes, mostly

Not without searching for resources or speaking to others

Not really

21. Are there any terms or concepts in particular that you don't fully understand?

22. In which areas of the DCF do you feel you need the most support? (Select all that apply)

Citizenship

Interacting and collaborating

Producing

Data and computational thinking

Other

23. Are you confident using a Micro:bit?

Yes, I have used them many times and applied them across the curriculum

I know the basics

I would benefit greatly from training

24. Are there any specific areas of the Welsh curriculum where you would like to see more digital resources and support?

Additional Information

25. Is there anything else you would like to share about your experiences or needs regarding digital skills in your teaching?

Outreach

26. Are you aware of Technocamps?

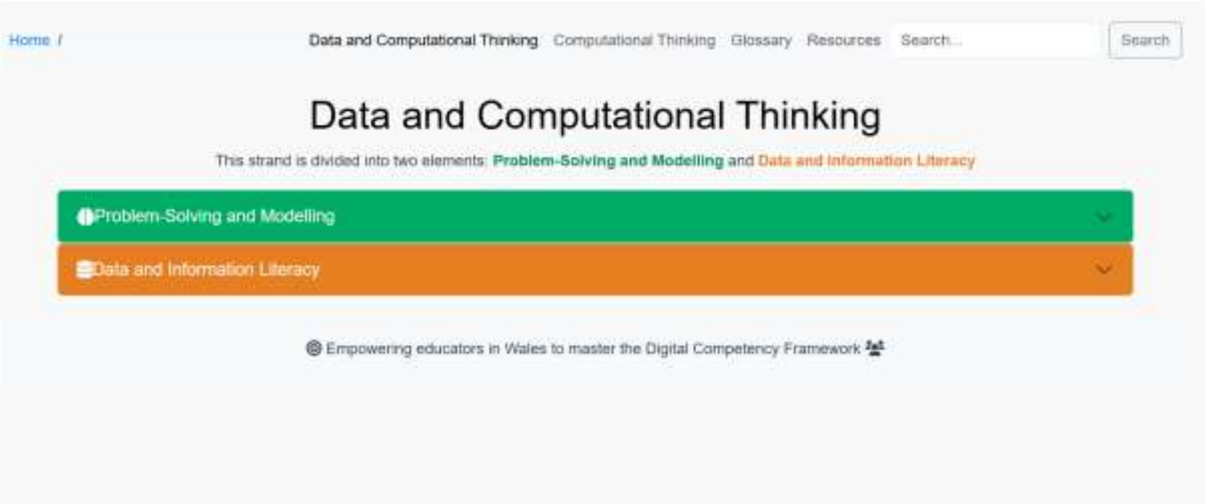
Yes, and we have been in contact and received free workshops / school plans to contact Technocamps

Know the name, but not much else

Never heard of Technocamps

Appendix 2

Interactive Guide

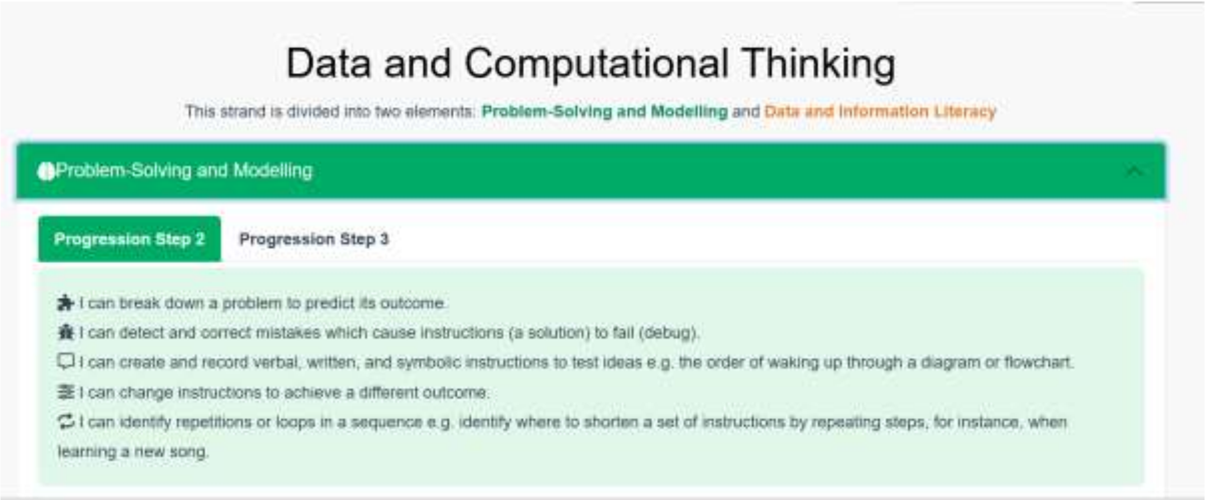

Home /
Data and Computational Thinking
Computational Thinking
Glossary
Resources
Search...
Search

Data and Computational Thinking

This strand is divided into two elements: **Problem-Solving and Modelling** and **Data and Information Literacy**

Problem-Solving and Modelling
Data and Information Literacy

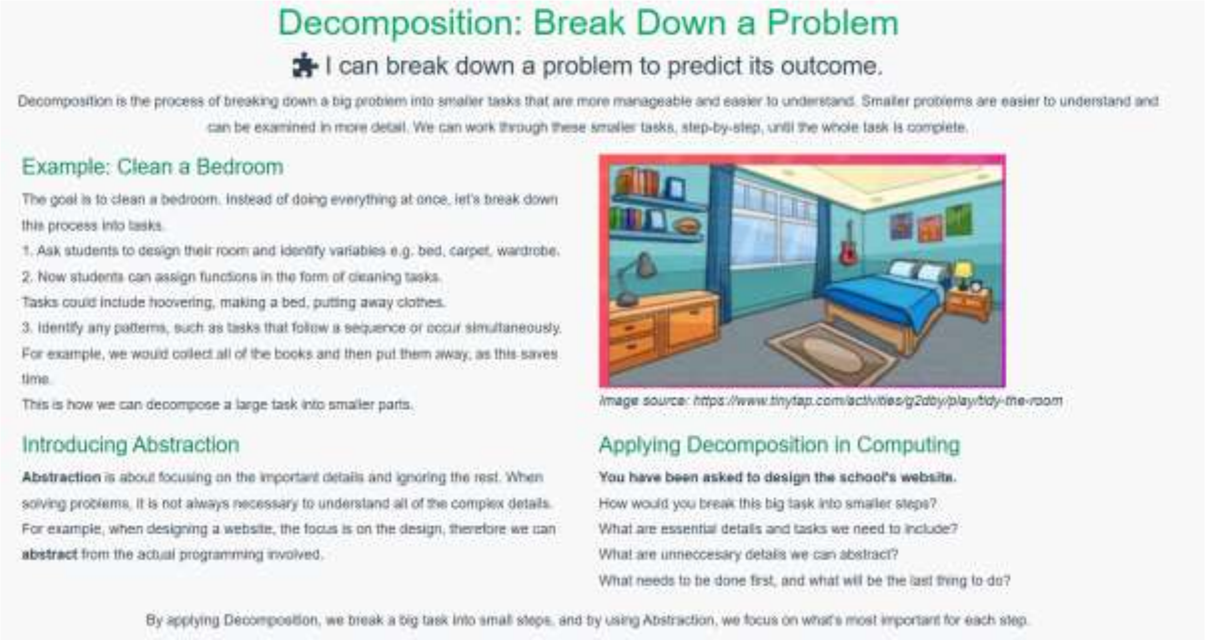
Empowering educators in Wales to master the Digital Competency Framework



Data and Computational Thinking

This strand is divided into two elements: **Problem-Solving and Modelling** and **Data and Information Literacy**

I can break down a problem to predict its outcome.
I can detect and correct mistakes which cause instructions (a solution) to fail (debug).
I can create and record verbal, written, and symbolic instructions to test ideas e.g. the order of waking up through a diagram or flowchart.
I can change instructions to achieve a different outcome.
I can identify repetitions or loops in a sequence e.g. identify where to shorten a set of instructions by repeating steps, for instance, when learning a new song.



Decomposition: Break Down a Problem

I can break down a problem to predict its outcome.

Decomposition is the process of breaking down a big problem into smaller tasks that are more manageable and easier to understand. Smaller problems are easier to understand and can be examined in more detail. We can work through these smaller tasks, step-by-step, until the whole task is complete.

Example: Clean a Bedroom

The goal is to clean a bedroom. Instead of doing everything at once, let's break down this process into tasks.
1. Ask students to design their room and identify variables e.g. bed, carpet, wardrobe.
2. Now students can assign functions in the form of cleaning tasks.
Tasks could include hoovering, making a bed, putting away clothes.
3. Identify any patterns, such as tasks that follow a sequence or occur simultaneously.
For example, we would collect all of the books and then put them away, as this saves time.
This is how we can decompose a large task into smaller parts.

Image source: https://www.tinytap.com/activities/g2dby/play/tidy-the-room

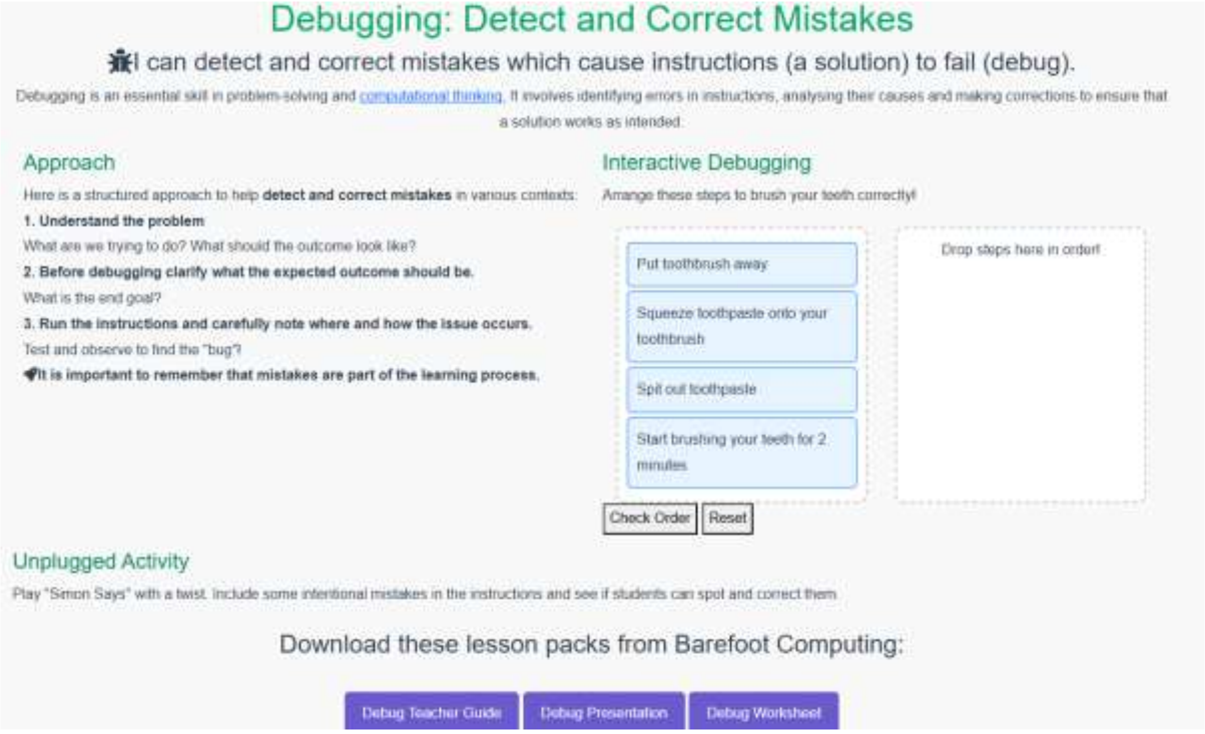
Introducing Abstraction

Abstraction is about focusing on the important details and ignoring the rest. When solving problems, it is not always necessary to understand all of the complex details. For example, when designing a website, the focus is on the design, therefore we can **abstract** from the actual programming involved.

Applying Decomposition in Computing

You have been asked to design the school's website.
How would you break this big task into smaller steps?
What are essential details and tasks we need to include?
What are unneccesary details we can abstract?
What needs to be done first, and what will be the last thing to do?

By applying Decomposition, we break a big task into small steps, and by using Abstraction, we focus on what's most important for each step.



Debugging: Detect and Correct Mistakes

I can detect and correct mistakes which cause instructions (a solution) to fail (debug).

Debugging is an essential skill in problem-solving and computational thinking. It involves identifying errors in instructions, analysing their causes and making corrections to ensure that a solution works as intended.

Approach

Here is a structured approach to help **detect and correct mistakes** in various contexts:

1. Understand the problem

What are we trying to do? What should the outcome look like?

2. Before debugging clarify what the expected outcome should be.

What is the end goal?

3. Run the instructions and carefully note where and how the issue occurs.

Test and observe to find the "bug"?

It is important to remember that mistakes are part of the learning process.

Interactive Debugging

Arrange these steps to brush your teeth correctly!

Put toothbrush away

Squeeze toothpaste onto your toothbrush

Spit out toothpaste

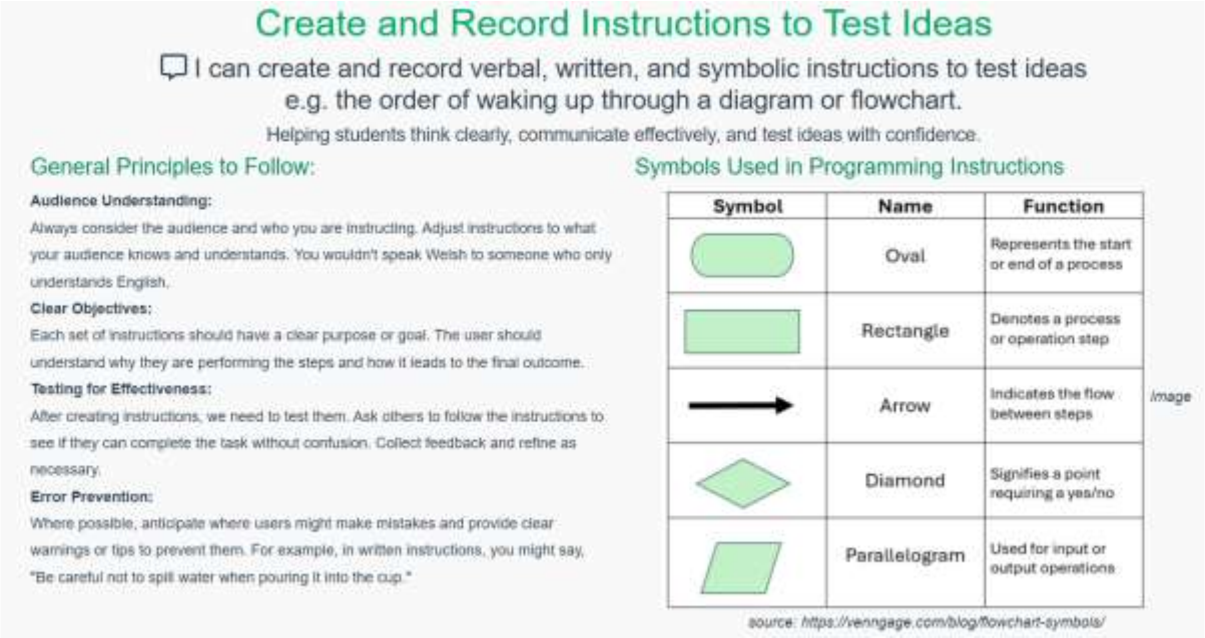
Start brushing your teeth for 2 minutes

Drop steps here in order!

Unplugged Activity

Play "Simon Says" with a twist. Include some intentional mistakes in the instructions and see if students can spot and correct them.

Download these lesson packs from Barefoot Computing:



Create and Record Instructions to Test Ideas

I can create and record verbal, written, and symbolic instructions to test ideas e.g. the order of waking up through a diagram or flowchart.

Helping students think clearly, communicate effectively, and test ideas with confidence.

General Principles to Follow:

Audience Understanding:

Always consider the audience and who you are instructing. Adjust instructions to what your audience knows and understands. You wouldn't speak Welsh to someone who only understands English.

Clear Objectives:

Each set of instructions should have a clear purpose or goal. The user should understand why they are performing the steps and how it leads to the final outcome.

Testing for Effectiveness:

After creating instructions, we need to test them. Ask others to follow the instructions to see if they can complete the task without confusion. Collect feedback and refine as necessary.

Error Prevention:

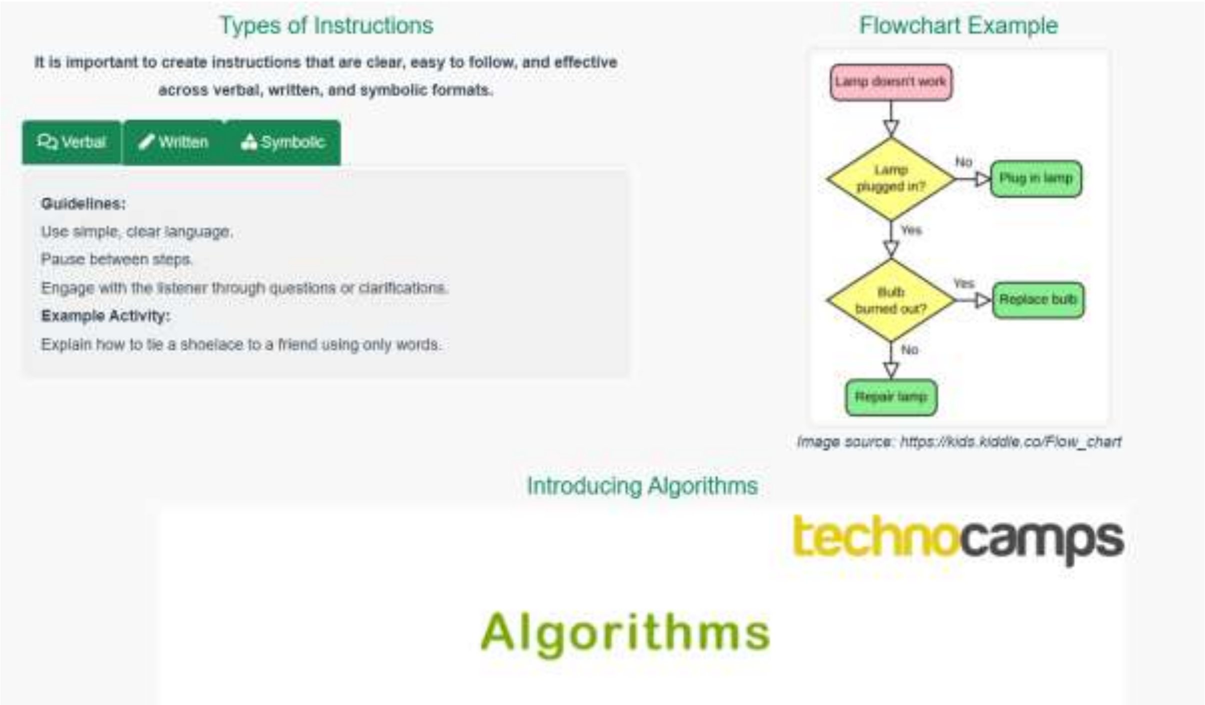
Where possible, anticipate where users might make mistakes and provide clear warnings or tips to prevent them. For example, in written instructions, you might say, "Be careful not to spill water when pouring it into the cup."

Symbols Used in Programming Instructions

Symbol	Name	Function
⬭	Oval	Represents the start or end of a process
▭	Rectangle	Denotes a process or operation step
→	Arrow	Indicates the flow between steps
◇	Diamond	Signifies a point requiring a yes/no
▱	Parallelogram	Used for input or output operations

Image

source: *https://venngage.com/blog/flowchart-symbols/*



Types of Instructions

It is important to create instructions that are clear, easy to follow, and effective across verbal, written, and symbolic formats.

Guidelines:

Use simple, clear language.

Pause between steps.

Engage with the listener through questions or clarifications.

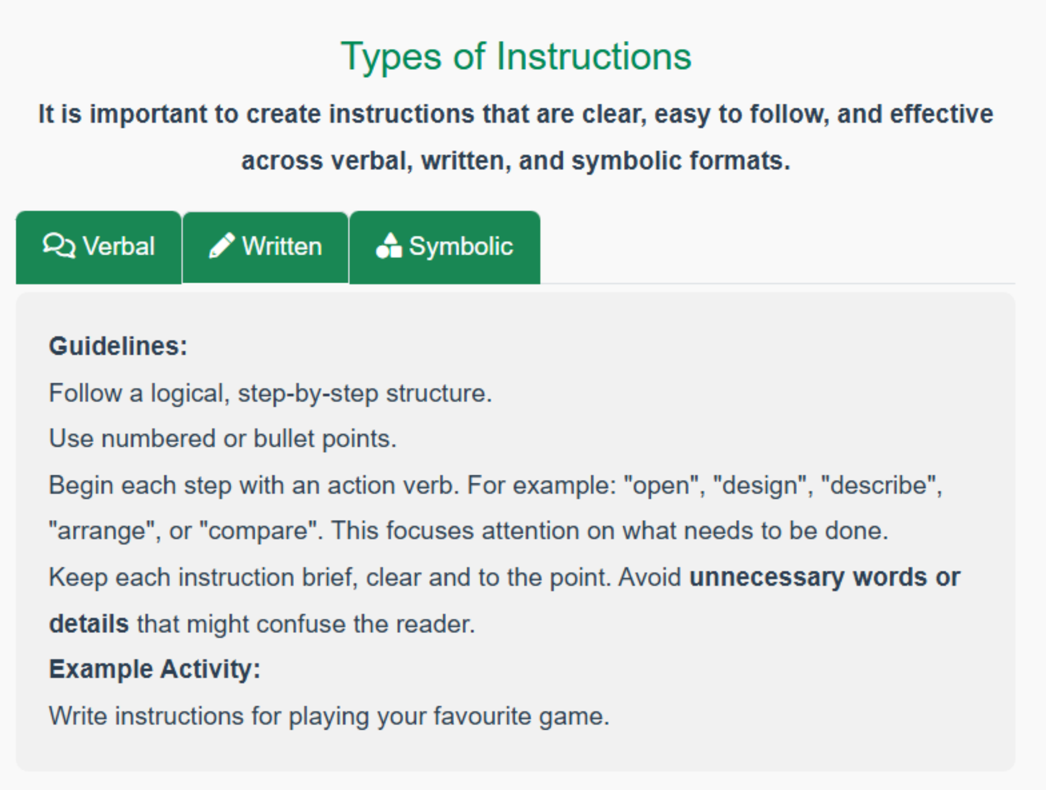
Example Activity:

Explain how to tie a shoelace to a friend using only words.

Flowchart Example

Image source: https://kids.kiddle.co/Flow_chart

Introducing Algorithms



Types of Instructions

It is important to create instructions that are clear, easy to follow, and effective across verbal, written, and symbolic formats.

Guidelines:

Follow a logical, step-by-step structure.

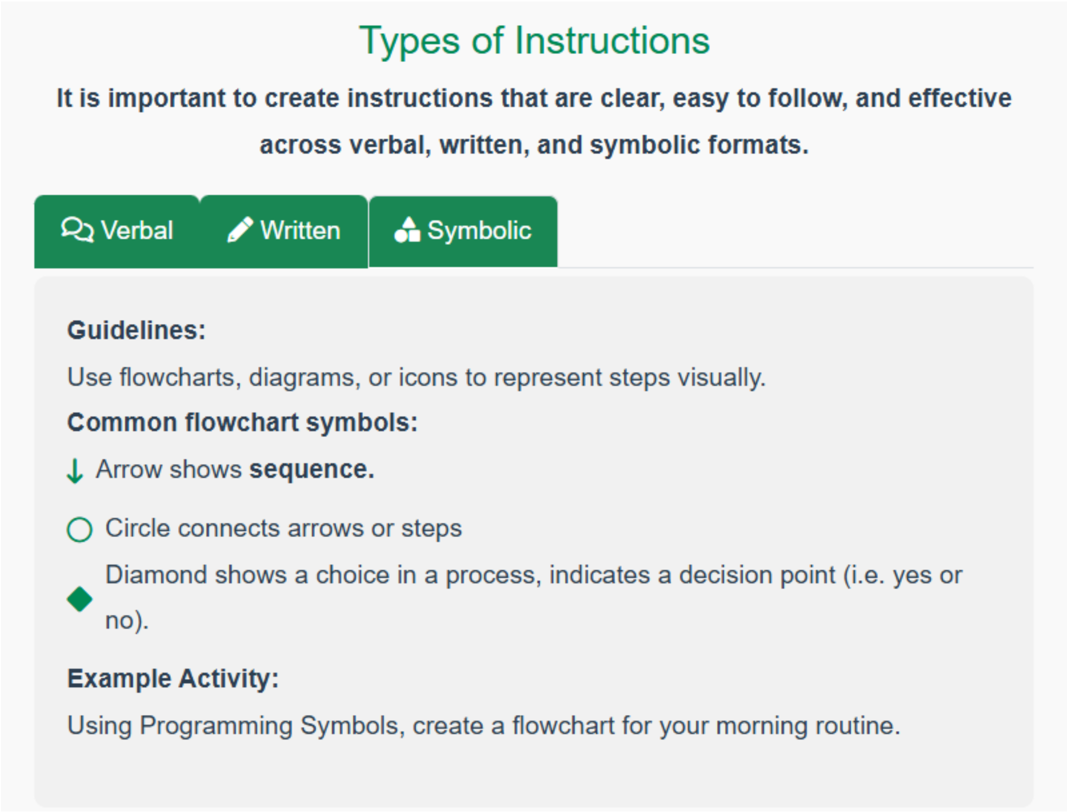
Use numbered or bullet points.

Begin each step with an action verb. For example: "open", "design", "describe", "arrange", or "compare". This focuses attention on what needs to be done.

Keep each instruction brief, clear and to the point. Avoid **unnecessary words or details** that might confuse the reader.

Example Activity:

Write instructions for playing your favourite game.



Types of Instructions

It is important to create instructions that are clear, easy to follow, and effective across verbal, written, and symbolic formats.

Verbal | Written | Symbolic

Guidelines:

Use flowcharts, diagrams, or icons to represent steps visually.

Common flowchart symbols:

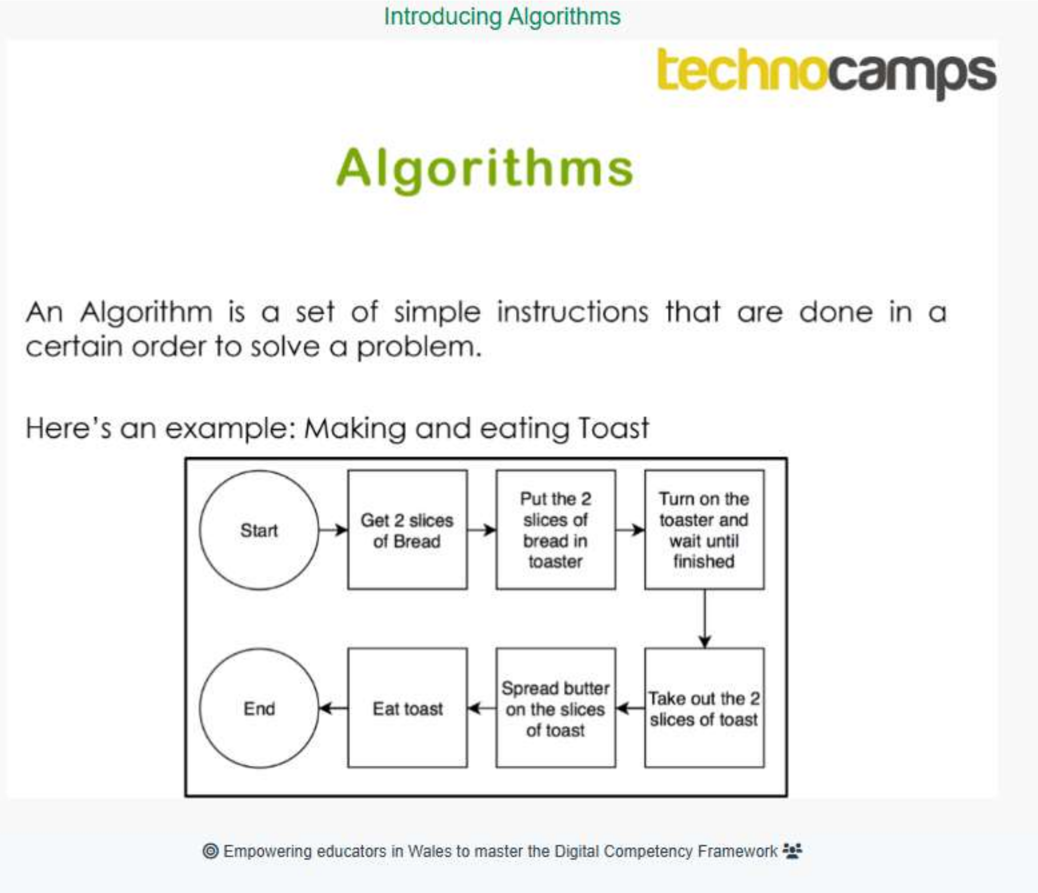
↓ Arrow shows **sequence.**

○ Circle connects arrows or steps

◆ Diamond shows a choice in a process, indicates a decision point (i.e. yes or no).

Example Activity:

Using Programming Symbols, create a flowchart for your morning routine.



Introducing Algorithms

technocamps

Algorithms

An Algorithm is a set of simple instructions that are done in a certain order to solve a problem.

Here's an example: Making and eating Toast

Start
Get 2 slices of Bread
Put the 2 slices of bread in toaster
Turn on the toaster and wait until finished
Take out the 2 slices of toast
Spread butter on the slices of toast
Eat toast
End

◎ Empowering educators in Wales to master the Digital Competency Framework

Change Instructions to Achieve a Different Outcome

I can change instructions to achieve a different outcome.

Encourage learners to experiment with and modify the steps in an algorithm to explore how different instructions impact the final outcome.

Iteration: Improving Through Repetition

Use iteration to refine and improve solutions.

Definition:

Iteration involves repeating a process while making incremental improvements based on prior results. It's not mere repetition — each iteration builds on the last.

Example

Repetition:

Typing the word "allow" repeatedly: "allow, allow, allow."

Iteration:

Typing variations that evolve: "allow, arrow, arrowhead."

Activity:

Design a basic word ladder puzzle where learners must iteratively change one letter in a word to reach a target word

(e.g., "cat" → "cot" → "dot" → "dog").

Use a flowchart or pseudo-code to demonstrate how iterative problem-solving works in programming.

Order in Mathematics: Understanding PEMDAS

The order of operations is important in mathematics.

PEMDAS: Parentheses, Exponents, Multiplication and Division, Addition and Subtraction

Consider addition and multiplication in the following example.

2 + 3 * 2

Following the order of operations, we do the multiplication first:

So, Step One: 3 * 2 = 6

Then we add 2

Step Two: 2 + 6 = 8

However, the order changes when brackets/parenthese are included

(2 + 3) * 2

Now, the brackets/parenthese are done first

So, Step One: 2 + 3 = 5

Then, we multiply by 2

Step Two: 5 * 2 = 10

Two different results, using the same data, with different operations.

Directions: Evaluating Effectiveness

🧭 Experiment with different ways of giving instructions to understand effectiveness 🧭

Scenario: Giving directions to a partner or robot.

↕ Is it more effective to say "Move forward 3 steps" or "Go North 3 steps"?

Discuss which method is clearer, in which situations e.g. indoors versus on a map.

⇄ Turn Left or move West? Turn Right or head East?

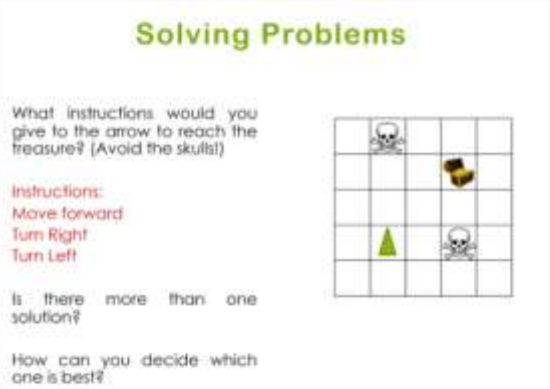
The Shortest Path vs The Longest Path

Find efficient or creative ways to solve navigation problems.

Activity:

Shortest Path to Treasure: Give instructions to guide the arrow to the treasure in the shortest number of moves.

Longest Path Challenge: Reverse the task, to maximize the distance traveled before reaching the treasure.



Solving Problems

What instructions would you give to the arrow to reach the treasure? (Avoid the skulls!)

Instructions:
Move forward
Turn Right
Turn Left

Is there more than one solution?

How can you decide which one is best?

Instructing a Robot

Find efficient or creative ways to solve navigation problems.

Activity:

Consider how you would instruct a robot differently to a person.

How do robots think?

🤖 **HINT:** Like a computer! 🖥

technocamps

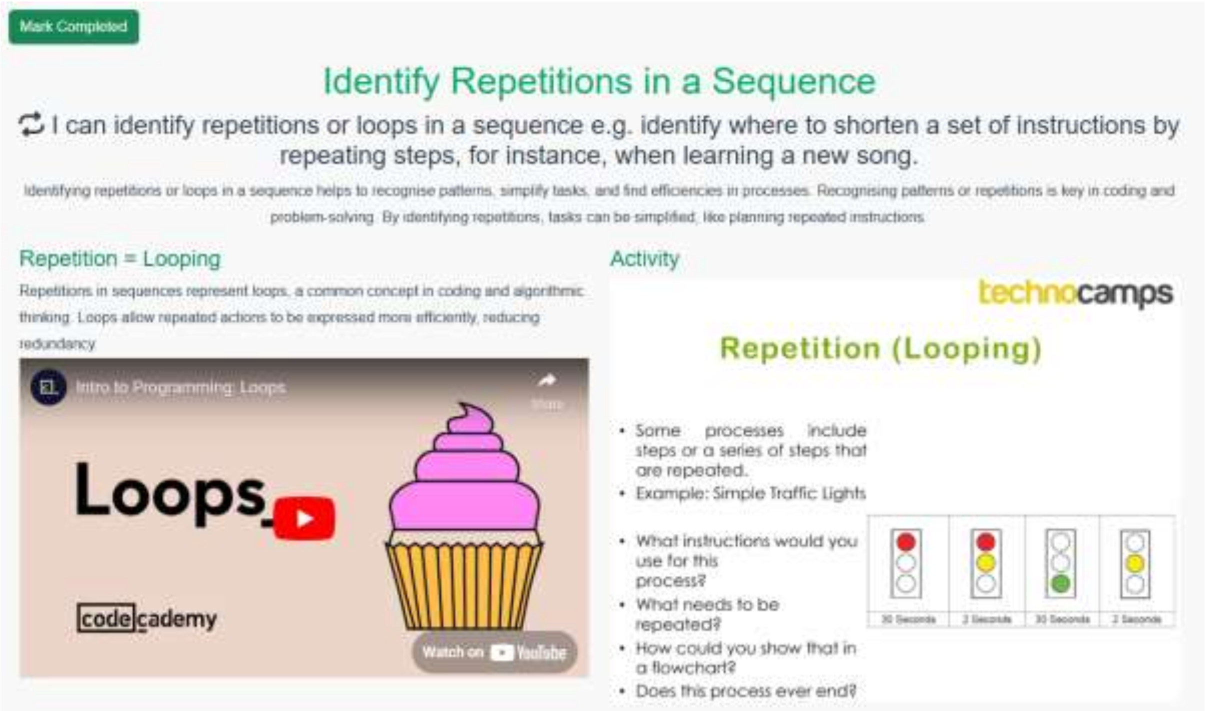
If I was a robot, how would you instruct me to ...

Walk Forward

Find the nearest wall

Move in a square

🌐 Empowering educators in Wales to master the Digital Competency Framework 👥



Mark Completed

Identify Repetitions in a Sequence

I can identify repetitions or loops in a sequence e.g. identify where to shorten a set of instructions by repeating steps, for instance, when learning a new song.

Identifying repetitions or loops in a sequence helps to recognise patterns, simplify tasks, and find efficiencies in processes. Recognising patterns or repetitions is key in coding and problem-solving. By identifying repetitions, tasks can be simplified, like planning repeated instructions.

Repetition = Looping

Repetitions in sequences represent loops, a common concept in coding and algorithmic thinking. Loops allow repeated actions to be expressed more efficiently, reducing redundancy.

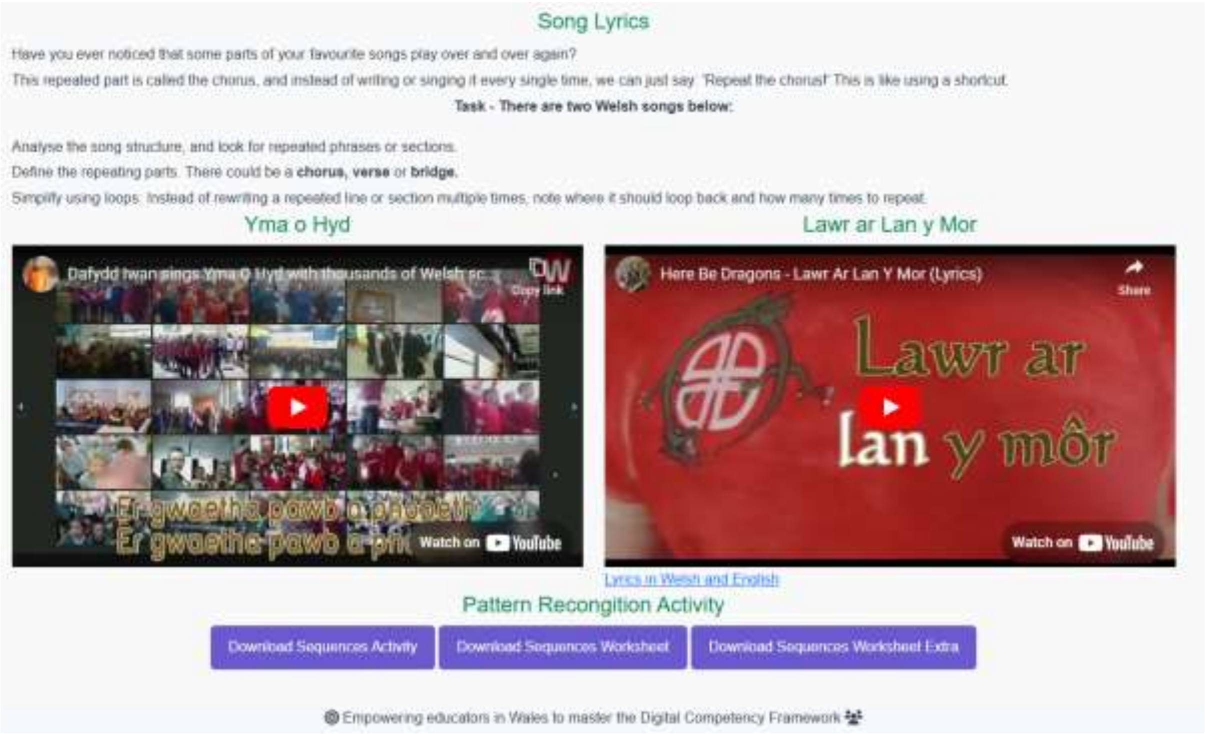
Activity

technocamps

Repetition (Looping)

- Some processes include steps or a series of steps that are repeated.
- Example: Simple Traffic Lights
- What instructions would you use for this process?
- What needs to be repeated?
- How could you show that in a flowchart?
- Does this process ever end?

🟥	🟥🟨	🟩	🟨
30 Seconds	2 Seconds	30 Seconds	2 Seconds



Song Lyrics

Have you ever noticed that some parts of your favourite songs play over and over again?

This repeated part is called the chorus, and instead of writing or singing it every single time, we can just say 'Repeat the chorus!' This is like using a shortcut.

Task - There are two Welsh songs below:

Analyse the song structure, and look for repeated phrases or sections.

Define the repeating parts. There could be a **chorus, verse** or **bridge.**

Simplify using loops: Instead of rewriting a repeated line or section multiple times, note where it should loop back and how many times to repeat.

Yma o Hyd

Dafydd Iwan sings Yma O Hyd with thousands of Welsh sc...
DW
Copy link
Er gwaetha pawb a phopeth
Er gwaetha pawb a'r pm
Watch on YouTube

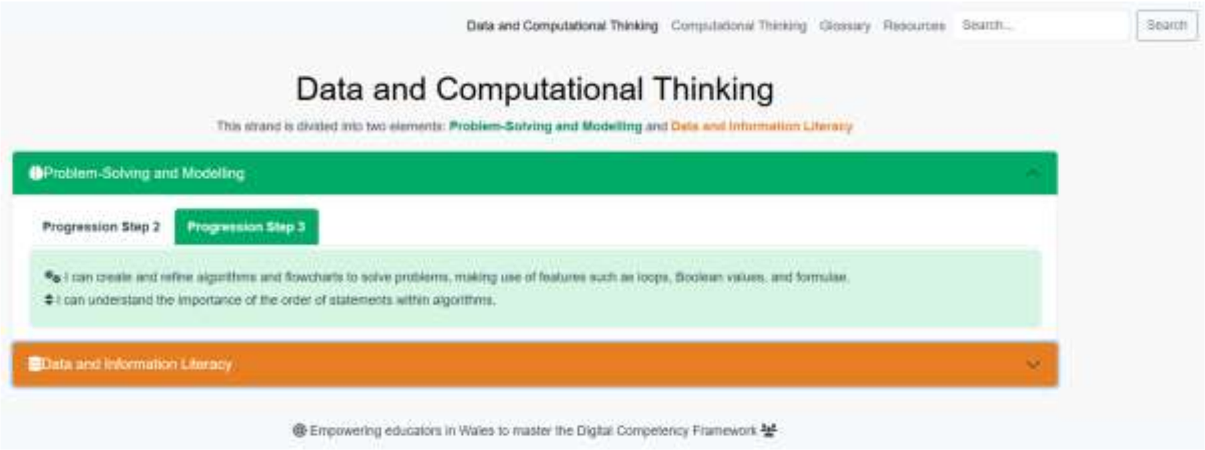
Lawr ar Lan y Mor

Lyrics in Welsh and English

Pattern Recognition Activity

Download Sequences Activity | Download Sequences Worksheet | Download Sequences Worksheet Extra

Empowering educators in Wales to master the Digital Competency Framework



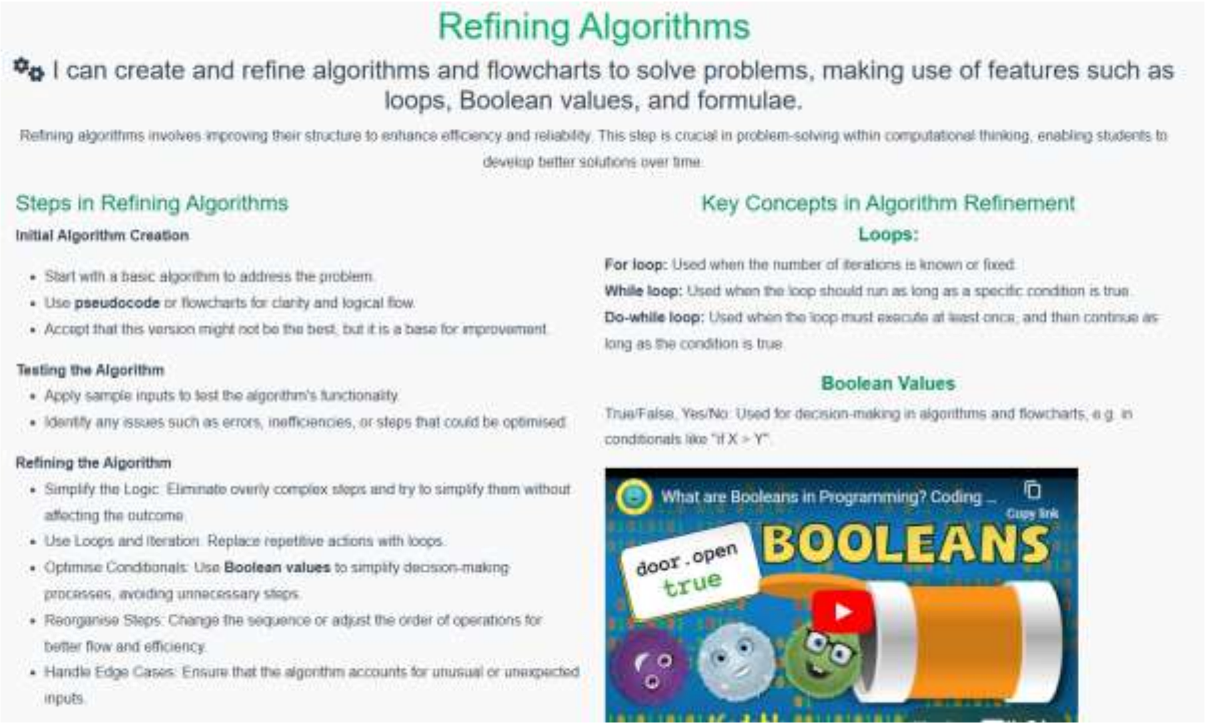
Data and Computational Thinking

This strand is divided into two elements: **Problem-Solving and Modelling** and **Data and Information Literacy**

I can create and refine algorithms and flowcharts to solve problems, making use of features such as loops, Boolean values, and formulae.

I can understand the importance of the order of statements within algorithms.

Empowering educators in Wales to master the Digital Competency Framework



Refining Algorithms

I can create and refine algorithms and flowcharts to solve problems, making use of features such as loops, Boolean values, and formulae.

Refining algorithms involves improving their structure to enhance efficiency and reliability. This step is crucial in problem-solving within computational thinking, enabling students to develop better solutions over time.

Steps in Refining Algorithms

Initial Algorithm Creation

- Start with a basic algorithm to address the problem.
- Use **pseudocode** or flowcharts for clarity and logical flow.
- Accept that this version might not be the best, but it is a base for improvement.

Testing the Algorithm

- Apply sample inputs to test the algorithm's functionality.
- Identify any issues such as errors, inefficiencies, or steps that could be optimised.

Refining the Algorithm

- Simplify the Logic: Eliminate overly complex steps and try to simplify them without affecting the outcome.
- Use Loops and Iteration: Replace repetitive actions with loops.
- Optimise Conditionals: Use **Boolean values** to simplify decision-making processes, avoiding unnecessary steps.
- Reorganise Steps: Change the sequence or adjust the order of operations for better flow and efficiency.
- Handle Edge Cases: Ensure that the algorithm accounts for unusual or unexpected inputs.

Key Concepts in Algorithm Refinement

Loops:

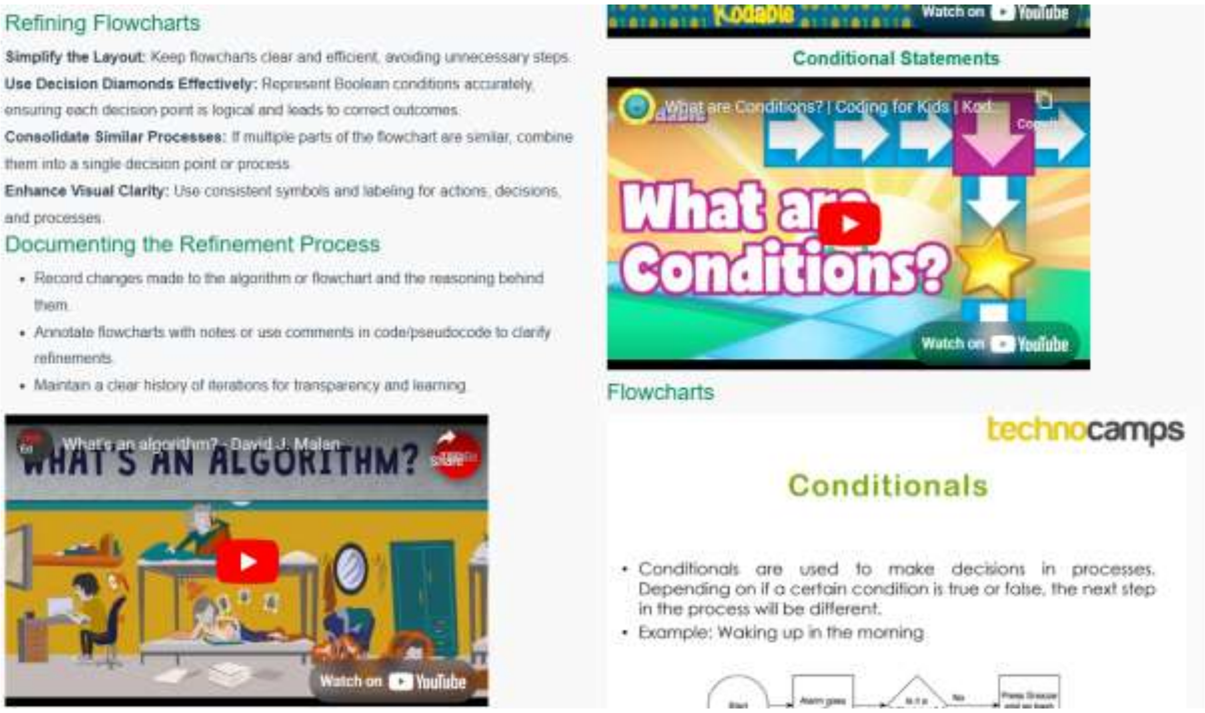
For loop: Used when the number of iterations is known or fixed.

While loop: Used when the loop should run as long as a specific condition is true.

Do-while loop: Used when the loop must execute at least once, and then continue as long as the condition is true.

Boolean Values

True/False, Yes/No: Used for decision-making in algorithms and flowcharts, e.g. in conditionals like "if X > Y".



Refining Flowcharts

Simplify the Layout: Keep flowcharts clear and efficient, avoiding unnecessary steps.

Use Decision Diamonds Effectively: Represent Boolean conditions accurately, ensuring each decision point is logical and leads to correct outcomes.

Consolidate Similar Processes: If multiple parts of the flowchart are similar, combine them into a single decision point or process.

Enhance Visual Clarity: Use consistent symbols and labeling for actions, decisions, and processes.

Documenting the Refinement Process

- Record changes made to the algorithm or flowchart and the reasoning behind them.
- Annotate flowcharts with notes or use comments in code/pseudocode to clarify refinements.
- Maintain a clear history of iterations for transparency and learning.

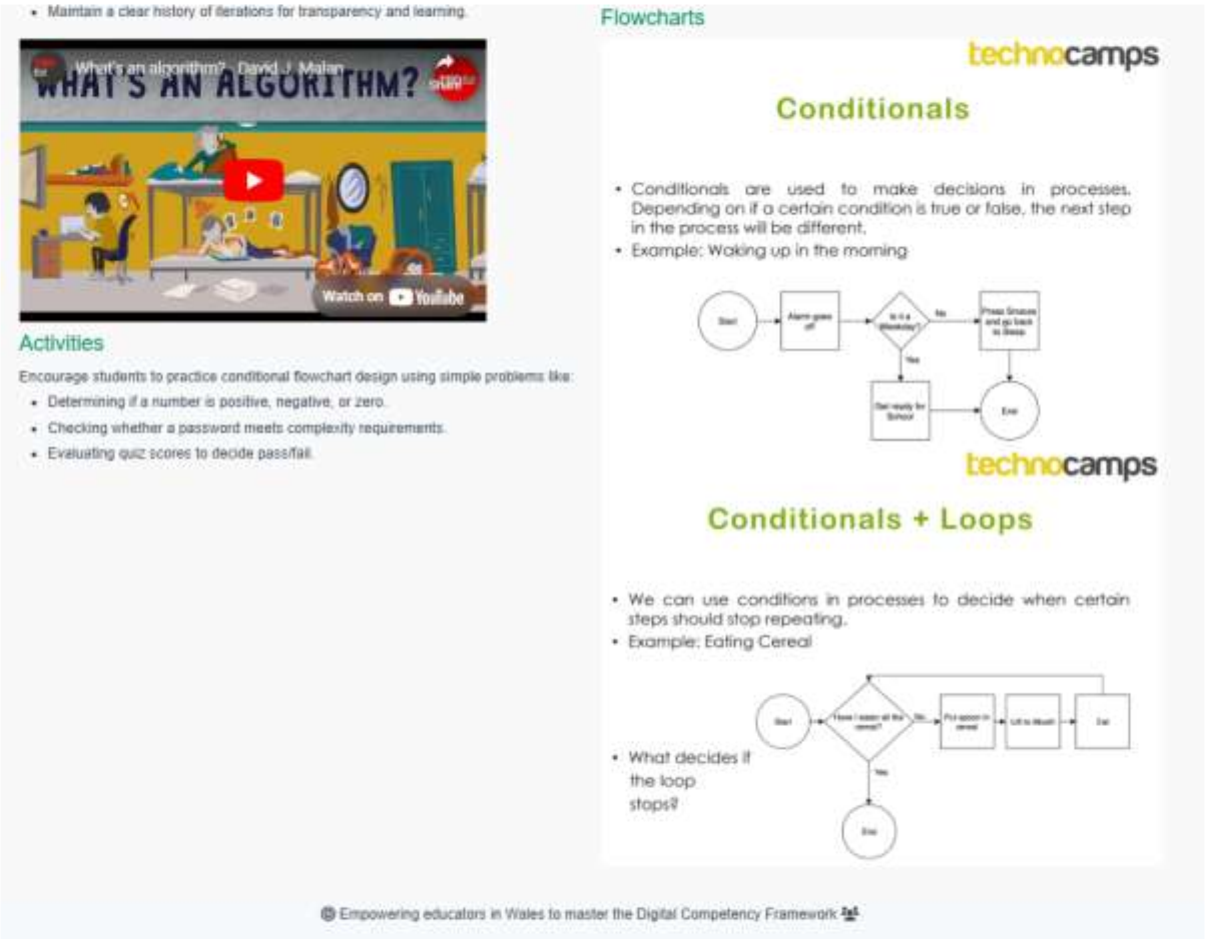
Conditional Statements

Flowcharts

technocamps

Conditionals

- Conditionals are used to make decisions in processes. Depending on if a certain condition is true or false, the next step in the process will be different.
- Example: Waking up in the morning



- Maintain a clear history of iterations for transparency and learning.

Activities

Encourage students to practice conditional flowchart design using simple problems like:

- Determining if a number is positive, negative, or zero.
- Checking whether a password meets complexity requirements.
- Evaluating quiz scores to decide pass/fail.

Flowcharts

technocamps

Conditionals

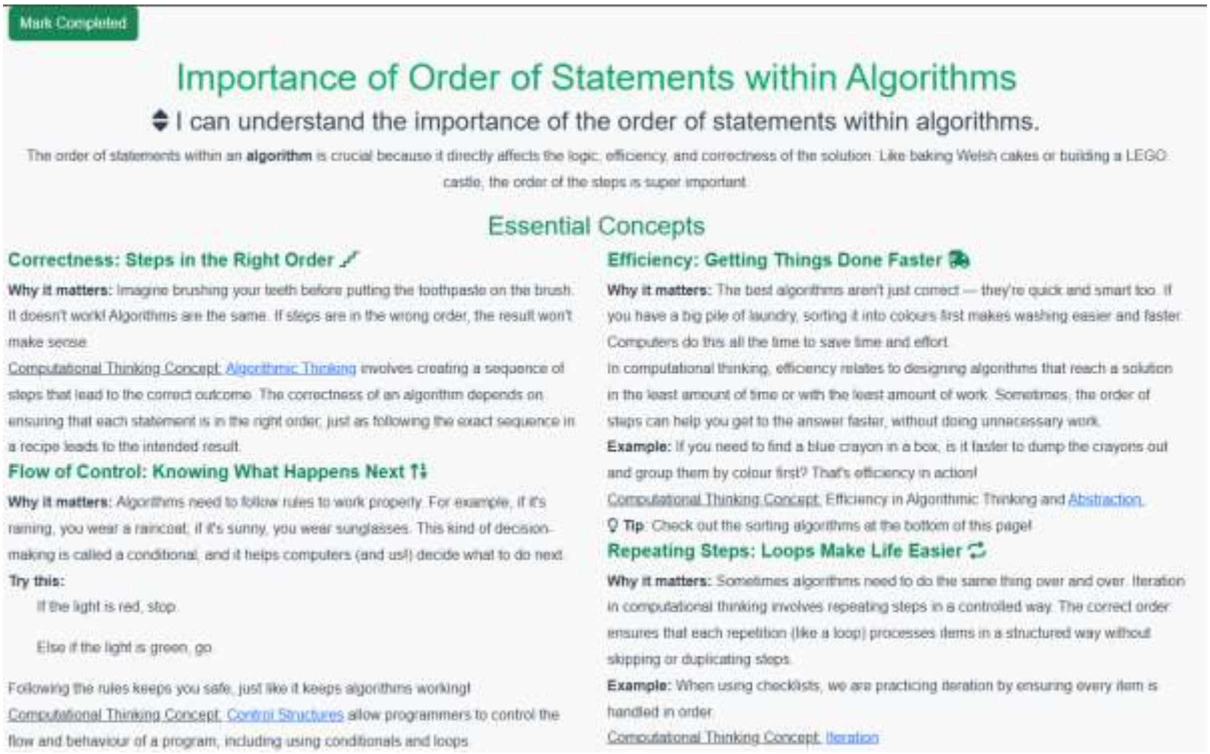
- Conditionals are used to make decisions in processes. Depending on if a certain condition is true or false, the next step in the process will be different.
- Example: Waking up in the morning

technocamps

Conditionals + Loops

- We can use conditions in processes to decide when certain steps should stop repeating.
- Example: Eating Cereal
- What decides if the loop stops?

🌐 Empowering educators in Wales to master the Digital Competency Framework 👨‍🏫



Importance of Order of Statements within Algorithms

⬍ I can understand the importance of the order of statements within algorithms.

The order of statements within an **algorithm** is crucial because it directly affects the logic, efficiency, and correctness of the solution. Like baking Welsh cakes or building a LEGO castle, the order of the steps is super important.

Essential Concepts

Correctness: Steps in the Right Order

Why it matters: Imagine brushing your teeth before putting the toothpaste on the brush. It doesn't work! Algorithms are the same. If steps are in the wrong order, the result won't make sense.

Computational Thinking Concept: Algorithmic Thinking involves creating a sequence of steps that lead to the correct outcome. The correctness of an algorithm depends on ensuring that each statement is in the right order, just as following the exact sequence in a recipe leads to the intended result.

Flow of Control: Knowing What Happens Next ⇅

Why it matters: Algorithms need to follow rules to work properly. For example, if it's raining, you wear a raincoat; if it's sunny, you wear sunglasses. This kind of decision-making is called a conditional, and it helps computers (and us!) decide what to do next.

Try this:

If the light is red, stop.

Else if the light is green, go.

Following the rules keeps you safe, just like it keeps algorithms working!

Computational Thinking Concept: Control Structures allow programmers to control the flow and behaviour of a program, including using conditionals and loops.

Efficiency: Getting Things Done Faster

Why it matters: The best algorithms aren't just correct — they're quick and smart too. If you have a big pile of laundry, sorting it into colours first makes washing easier and faster. Computers do this all the time to save time and effort.

In computational thinking, efficiency relates to designing algorithms that reach a solution in the least amount of time or with the least amount of work. Sometimes, the order of steps can help you get to the answer faster, without doing unnecessary work.

Example: If you need to find a blue crayon in a box, is it faster to dump the crayons out and group them by colour first? That's efficiency in action!

Computational Thinking Concept: Efficiency in Algorithmic Thinking and Abstraction.

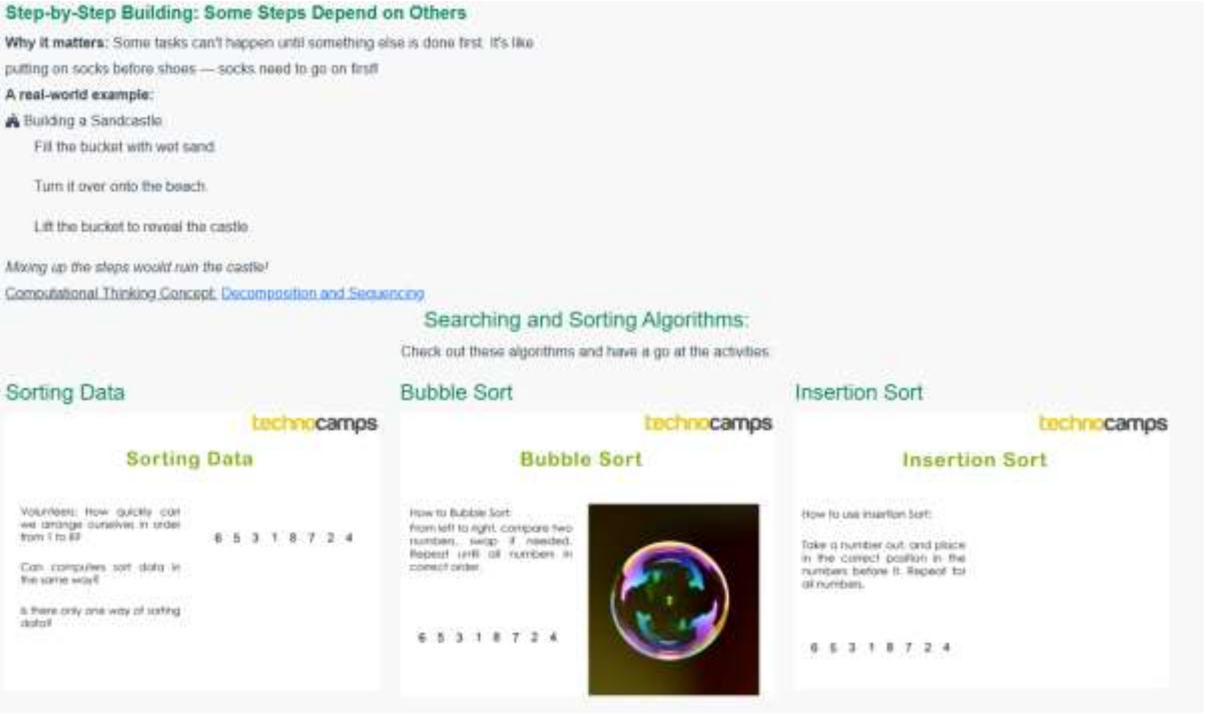
💡 **Tip**: Check out the sorting algorithms at the bottom of this page!

Repeating Steps: Loops Make Life Easier 🔁

Why it matters: Sometimes algorithms need to do the same thing over and over. Iteration in computational thinking involves repeating steps in a controlled way. The correct order ensures that each repetition (like a loop) processes items in a structured way without skipping or duplicating steps.

Example: When using checklists, we are practicing iteration by ensuring every item is handled in order.

Computational Thinking Concept: Iteration



Step-by-Step Building: Some Steps Depend on Others

Why it matters: Some tasks can't happen until something else is done first. It's like putting on socks before shoes — socks need to go on first!

A real-world example:

🏰 Building a Sandcastle

Fill the bucket with wet sand.

Turn it over onto the beach.

Lift the bucket to reveal the castle.

Mixing up the steps would ruin the castle!

Computational Thinking Concept: Decomposition and Sequencing

Searching and Sorting Algorithms:

Check out these algorithms and have a go at the activities.

Sorting Data

Sorting Data

Volunteers: How quickly can we arrange ourselves in order from 1 to 8?

6 5 3 1 8 7 2 4

Can computers sort data in the same way?

Is there only one way of sorting data?

Bubble Sort

Bubble Sort

How to Bubble Sort:

From left to right, compare two numbers, swap if needed. Repeat until all numbers in correct order.

6 5 3 1 8 7 2 4

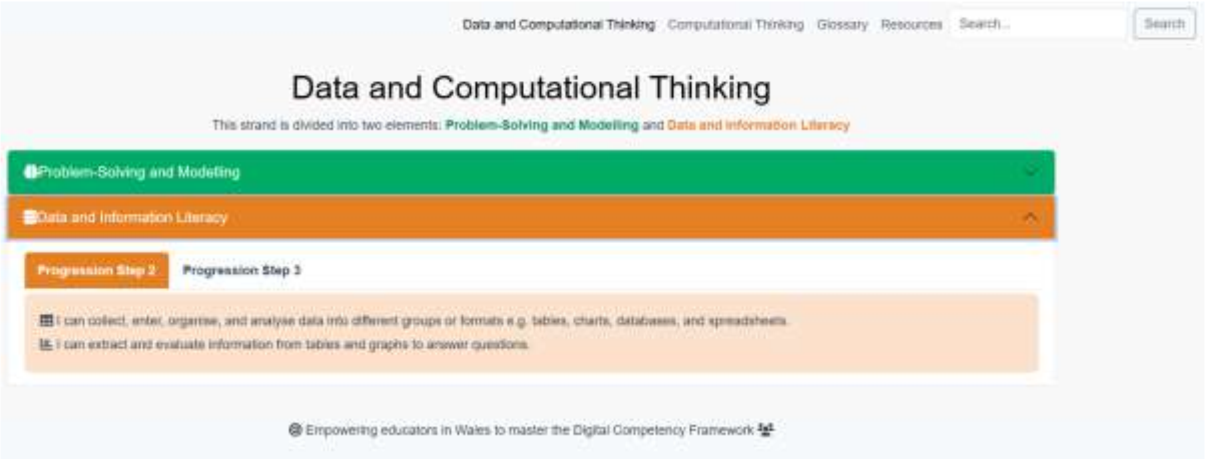
Insertion Sort

Insertion Sort

How to use Insertion Sort:

Take a number out and place in the correct position in the numbers before it. Repeat for all numbers.

6 5 3 1 8 7 2 4



Data and Computational Thinking

This strand is divided into two elements: **Problem-Solving and Modelling** and **Data and Information Literacy**

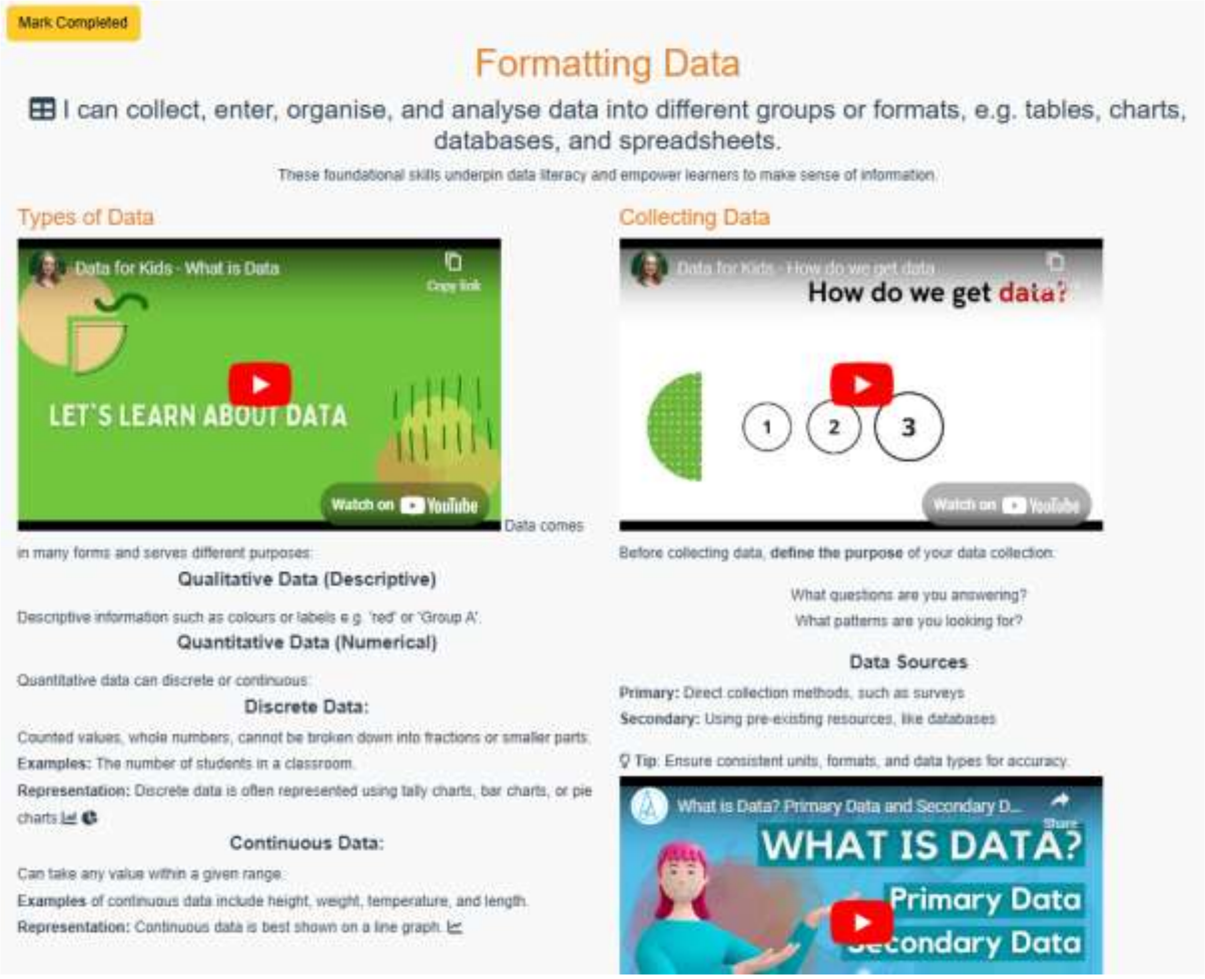
Problem-Solving and Modelling

Data and Information Literacy

I can collect, enter, organise, and analyse data into different groups or formats e.g. tables, charts, databases, and spreadsheets.

I can extract and evaluate information from tables and graphs to answer questions.

Empowering educators in Wales to master the Digital Competency Framework



Formatting Data

I can collect, enter, organise, and analyse data into different groups or formats, e.g. tables, charts, databases, and spreadsheets.

These foundational skills underpin data literacy and empower learners to make sense of information.

Types of Data

Data comes in many forms and serves different purposes:

Qualitative Data (Descriptive)

Descriptive information such as colours or labels e.g. 'red' or 'Group A'.

Quantitative Data (Numerical)

Quantitative data can discrete or continuous:

Discrete Data:

Counted values, whole numbers, cannot be broken down into fractions or smaller parts.

Examples: The number of students in a classroom.

Representation: Discrete data is often represented using tally charts, bar charts, or pie charts

Continuous Data:

Can take any value within a given range.

Examples of continuous data include height, weight, temperature, and length.

Representation: Continuous data is best shown on a line graph.

Collecting Data

Before collecting data, **define the purpose** of your data collection:

What questions are you answering?

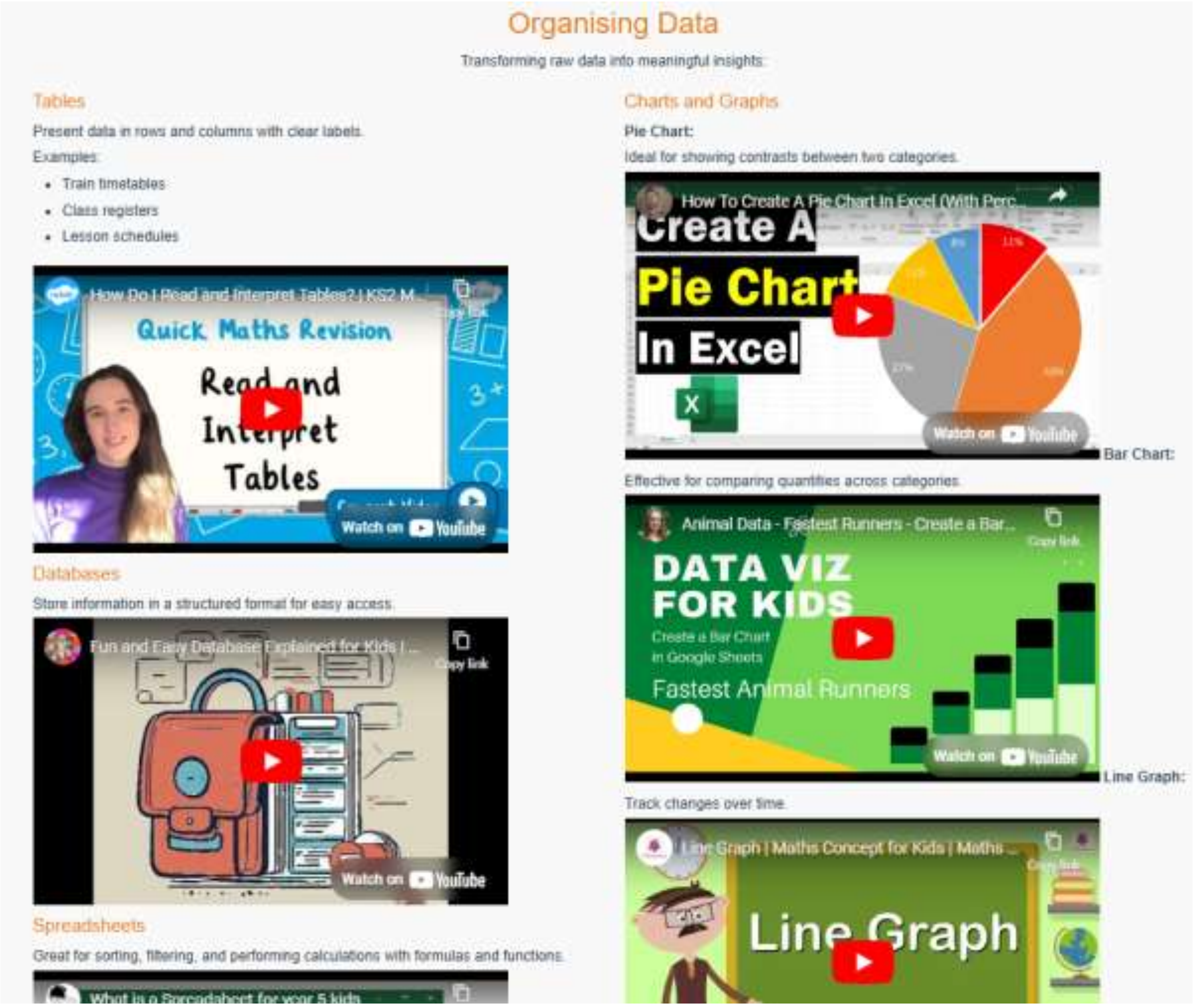
What patterns are you looking for?

Data Sources

Primary: Direct collection methods, such as surveys

Secondary: Using pre-existing resources, like databases

Tip: Ensure consistent units, formats, and data types for accuracy.



Organising Data

Transforming raw data into meaningful insights:

Tables

Present data in rows and columns with clear labels.

Examples:

- Train timetables
- Class registers
- Lesson schedules

How Do I Read and Interpret Tables? | KS2 M...
Quick Maths Revision
Read and Interpret Tables
Watch on YouTube

Databases

Store information in a structured format for easy access.

Fun and Easy Database Explained for Kids | ...
Copy link
Watch on YouTube

Spreadsheets

Great for sorting, filtering, and performing calculations with formulas and functions.

What is a Spreadsheet for year 5 kids

Charts and Graphs

Pie Chart:

Ideal for showing contrasts between two categories.

How To Create A Pie Chart In Excel (With Perc...
Create A Pie Chart In Excel
Watch on YouTube

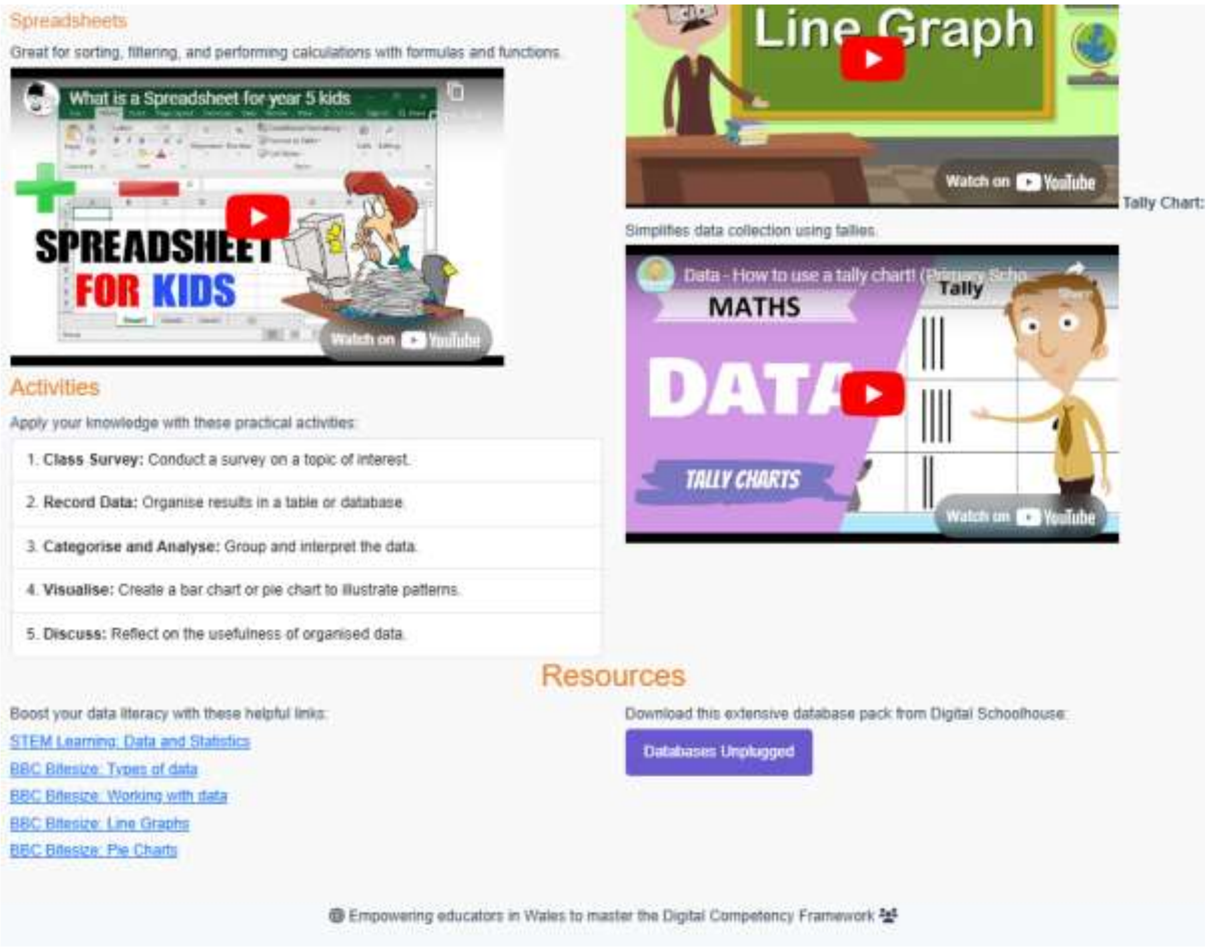
Bar Chart:

Effective for comparing quantities across categories.

Animal Data - Fastest Runners - Create a Bar...
Copy link
DATA VIZ FOR KIDS
Create a Bar Chart in Google Sheets
Fastest Animal Runners
Watch on YouTube

Line Graph:

Track changes over time.



Spreadsheets

Great for sorting, filtering, and performing calculations with formulas and functions.

Activities

Apply your knowledge with these practical activities:

1. **Class Survey:** Conduct a survey on a topic of interest.
2. **Record Data:** Organise results in a table or database.
3. **Categorise and Analyse:** Group and interpret the data.
4. **Visualise:** Create a bar chart or pie chart to illustrate patterns.
5. **Discuss:** Reflect on the usefulness of organised data.

Tally Chart:

Simplifies data collection using tallies.

Resources

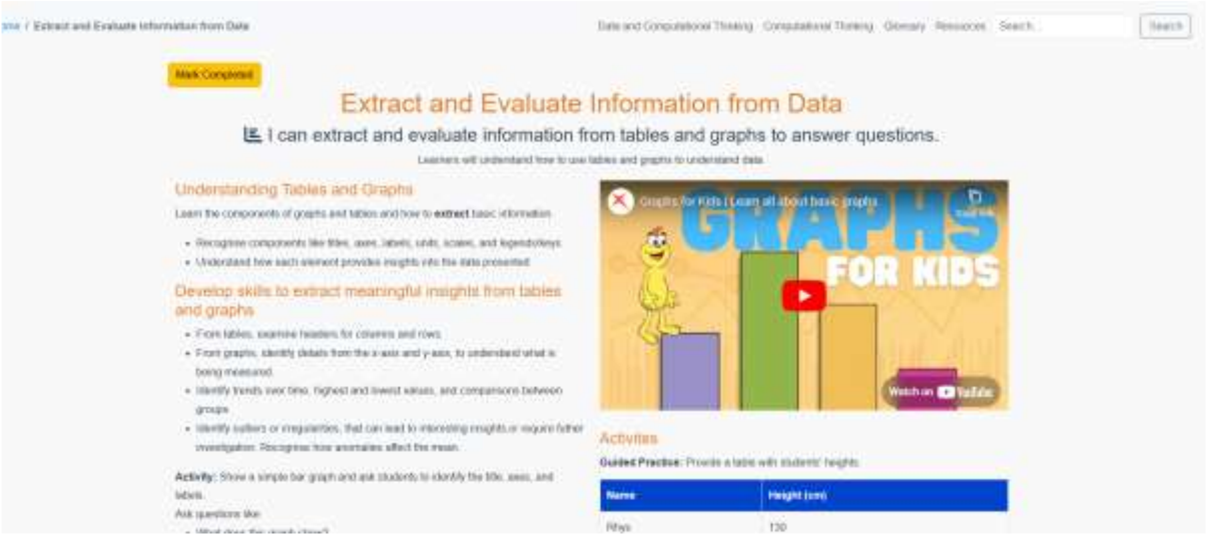
Boost your data literacy with these helpful links:

STEM Learning: Data and Statistics
BBC Bitesize: Types of data
BBC Bitesize: Working with data
BBC Bitesize: Line Graphs
BBC Bitesize: Pie Charts

Download this extensive database pack from Digital Schoolhouse:

Databases Unplugged

🌐 Empowering educators in Wales to master the Digital Competency Framework 👥



ome / Extract and Evaluate Information from Data
Data and Computational Thinking
Computational Thinking
Glossary
Resources
Search...
Search
Mark Completed
Graphs for Kids | Learn all about basic graphs
GRAPHS FOR KIDS
Watch on YouTube

Extract and Evaluate Information from Data

📈 I can extract and evaluate information from tables and graphs to answer questions.

Learners will understand how to use tables and graphs to understand data.

Understanding Tables and Graphs

Learn the components of graphs and tables and how to **extract** basic information.

- Recognise components like titles, axes, labels, units, scales, and legends/keys.
- Understand how each element provides insights into the data presented.

Develop skills to extract meaningful insights from tables and graphs

- From tables, examine headers for columns and rows.
- From graphs, identify details from the x-axis and y-axis, to understand what is being measured.
- Identify trends over time, highest and lowest values, and comparisons between groups.
- Identify outliers or irregularities, that can lead to interesting insights or require futher investigation. Recognise how anomalies affect the mean.

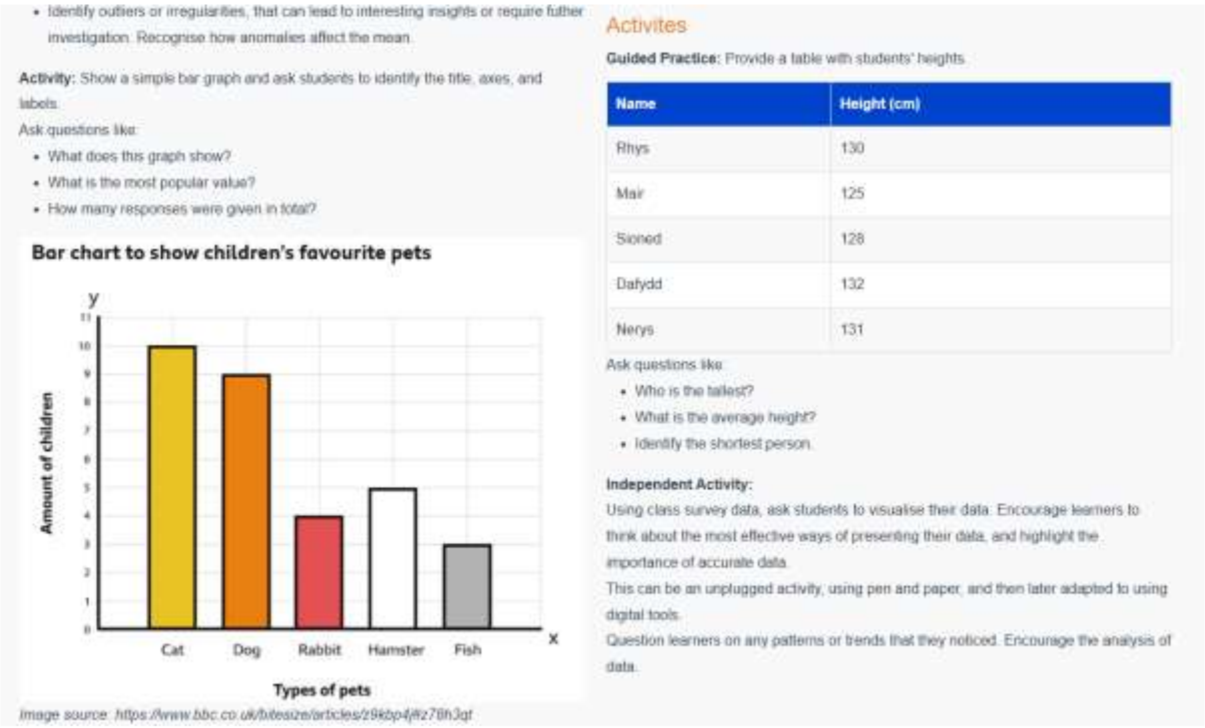
Activity: Show a simple bar graph and ask students to identify the title, axes, and labels.
Ask questions like:

- What does the graph show?

Activities

Guided Practice: Provide a table with students' heights.

Name	Height (cm)
Rhys	130



- Identify outliers or irregularities, that can lead to interesting insights or require futher investigation. Recognise how anomalies affect the mean.

Activity: Show a simple bar graph and ask students to identify the title, axes, and labels.

Ask questions like:

- What does this graph show?
- What is the most popular value?
- How many responses were given in total?

Bar chart to show children's favourite pets

Image source: https://www.bbc.co.uk/bitesize/articles/z9kbp4j#z78h3qt

Activites

Guided Practice: Provide a table with students' heights.

Name	Height (cm)
Rhys	130
Mair	125
Sioned	128
Dafydd	132
Nerys	131

Ask questions like:

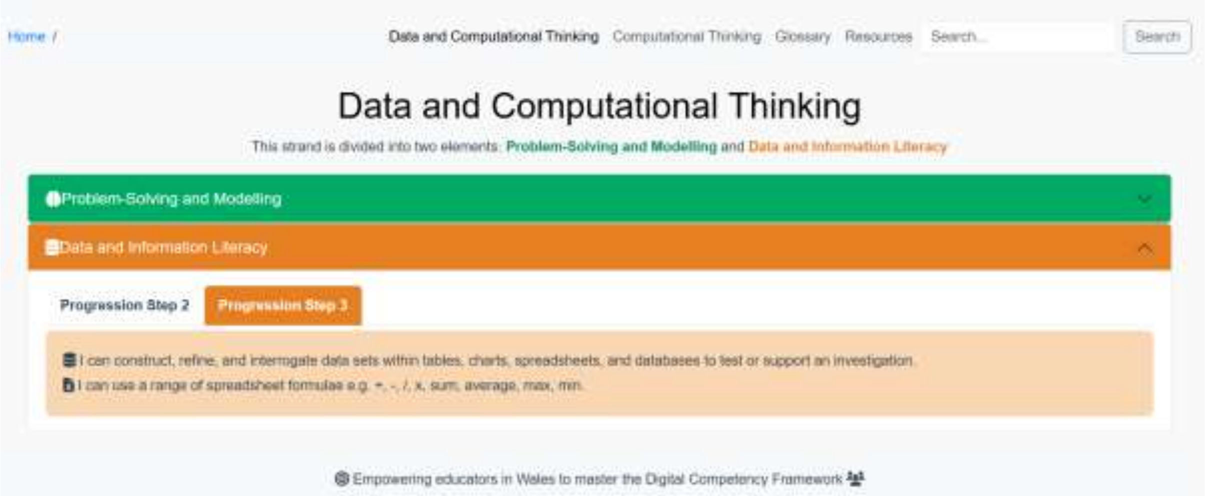
- Who is the tallest?
- What is the average height?
- Identify the shortest person.

Independent Activity:

Using class survey data, ask students to visualise their data. Encourage learners to think about the most effective ways of presenting their data, and highlight the importance of accurate data.

This can be an unplugged activity, using pen and paper, and then later adapted to using digital tools.

Question learners on any patterns or trends that they noticed. Encourage the analysis of data.



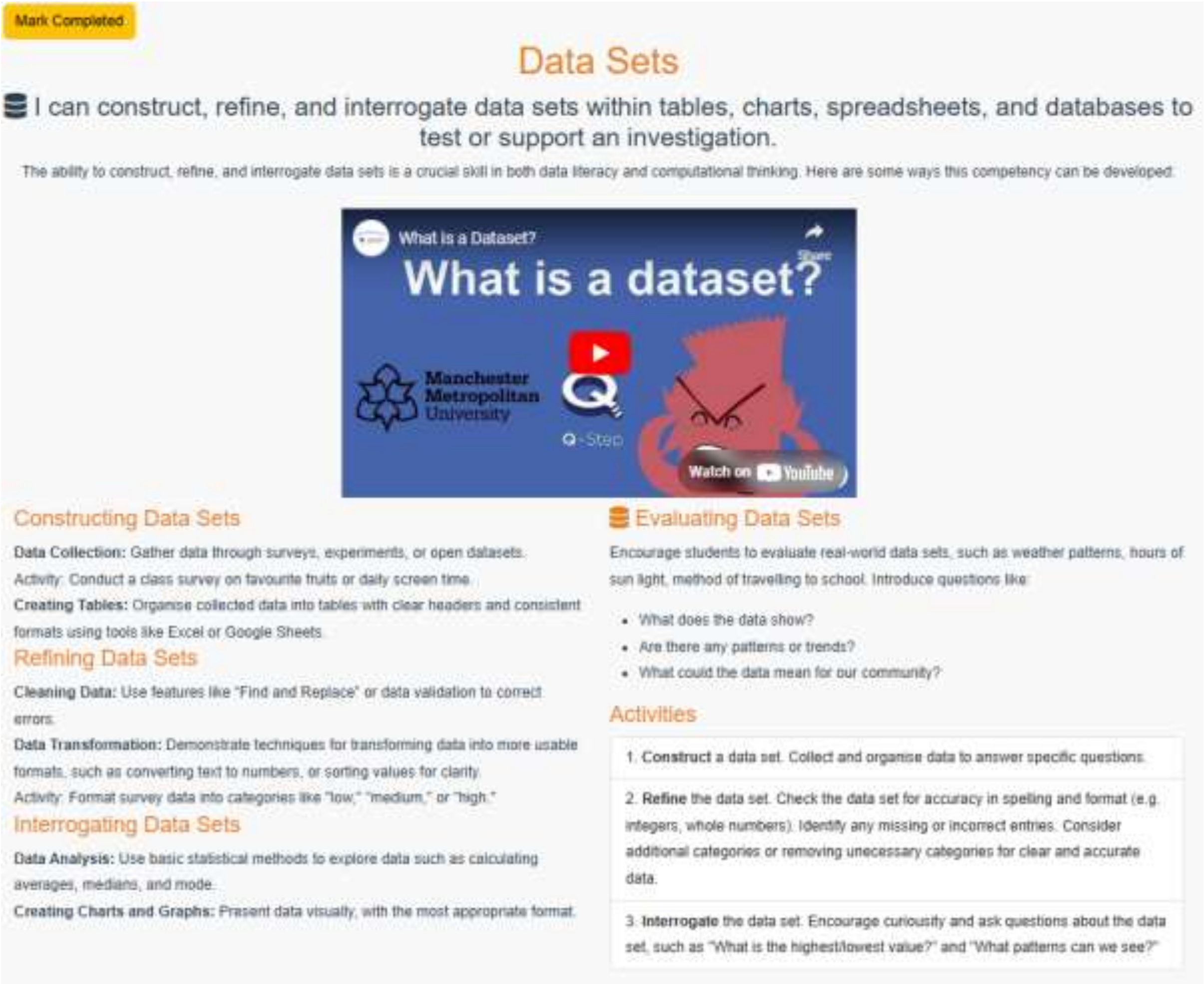
Data and Computational Thinking

This strand is divided into two elements: **Problem-Solving and Modelling** and **Data and Information Literacy**

I can construct, refine, and interrogate data sets within tables, charts, spreadsheets, and databases to test or support an investigation.

I can use a range of spreadsheet formulas e.g. +, -, /, x, sum, average, max, min.

Empowering educators in Wales to master the Digital Competency Framework



Mark Completed

Data Sets

I can construct, refine, and interrogate data sets within tables, charts, spreadsheets, and databases to test or support an investigation.

The ability to construct, refine, and interrogate data sets is a crucial skill in both data literacy and computational thinking. Here are some ways this competency can be developed:

Constructing Data Sets

Data Collection: Gather data through surveys, experiments, or open datasets.

Activity: Conduct a class survey on favourite fruits or daily screen time.

Creating Tables: Organise collected data into tables with clear headers and consistent formats using tools like Excel or Google Sheets.

Refining Data Sets

Cleaning Data: Use features like "Find and Replace" or data validation to correct errors.

Data Transformation: Demonstrate techniques for transforming data into more usable formats, such as converting text to numbers, or sorting values for clarity.

Activity: Format survey data into categories like "low," "medium," or "high."

Interrogating Data Sets

Data Analysis: Use basic statistical methods to explore data such as calculating averages, medians, and mode.

Creating Charts and Graphs: Present data visually, with the most appropriate format.

Evaluating Data Sets

Encourage students to evaluate real-world data sets, such as weather patterns, hours of sun light, method of travelling to school. Introduce questions like:

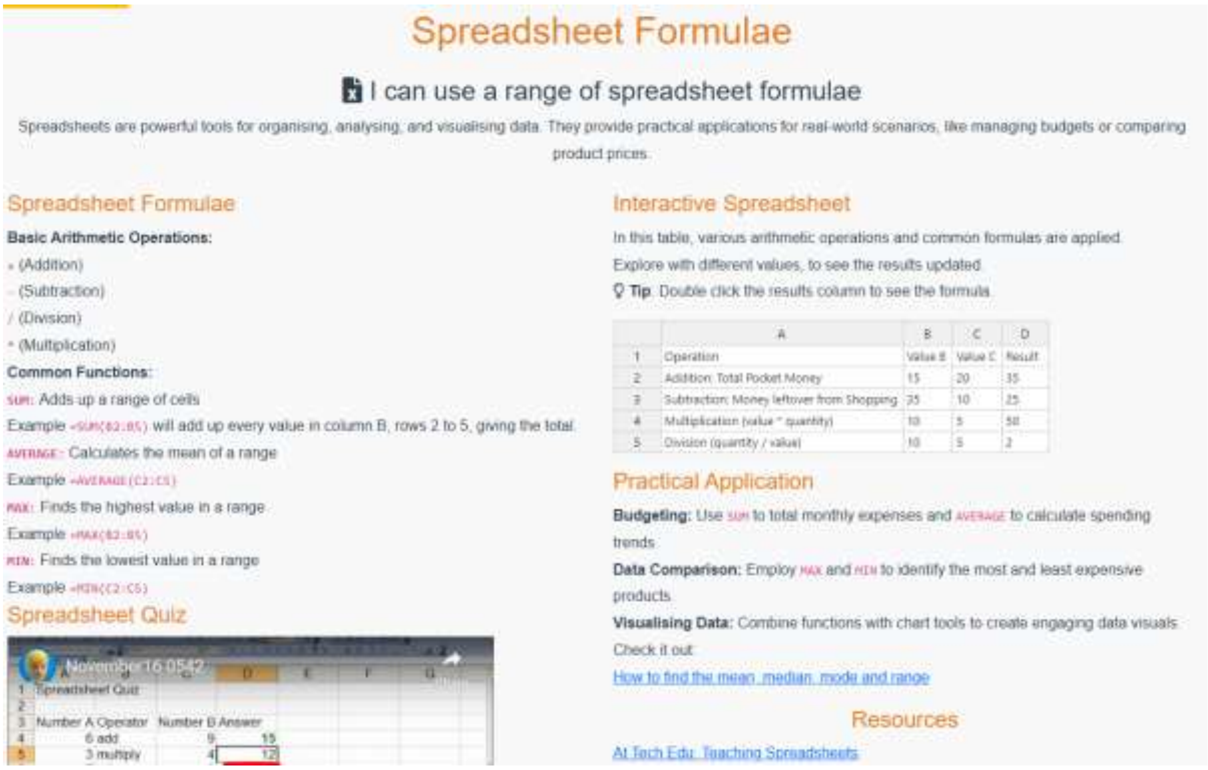
- What does the data show?
- Are there any patterns or trends?
- What could the data mean for our community?

Activities

1. **Construct** a data set. Collect and organise data to answer specific questions.

2. **Refine** the data set. Check the data set for accuracy in spelling and format (e.g. integers, whole numbers). Identify any missing or incorrect entries. Consider additional categories or removing unecessary categories for clear and accurate data.

3. **Interrogate** the data set. Encourage curiousity and ask questions about the data set, such as "What is the highest/lowest value?" and "What patterns can we see?"



Spreadsheet Formulae

I can use a range of spreadsheet formulae

Spreadsheets are powerful tools for organising, analysing, and visualising data. They provide practical applications for real-world scenarios, like managing budgets or comparing product prices.

Spreadsheet Formulae

Basic Arithmetic Operations:

+ (Addition)

- (Subtraction)

/ (Division)

* (Multiplication)

Common Functions:

SUM: Adds up a range of cells

Example =SUM(B2:B5) will add up every value in column B, rows 2 to 5, giving the total.

AVERAGE: Calculates the mean of a range

Example =AVERAGE(C2:C5)

MAX: Finds the highest value in a range

Example =MAX(B2:B5)

MIN: Finds the lowest value in a range

Example =MIN(C2:C5)

Spreadsheet Quiz

November 16 0542

	A	B	C	D
1	Spreadsheet Quiz			
2				
3	Number A	Operator	Number B	Answer
4	6	add	9	15
5	3	multiply	4	12

Interactive Spreadsheet

In this table, various arithmetic operations and common formulas are applied.

Explore with different values, to see the results updated.

Tip: Double click the results column to see the formula.

	A	B	C	D
1	Operation	Value B	Value C	Result
2	Addition: Total Pocket Money	15	20	35
3	Subtraction: Money leftover from Shopping	35	10	25
4	Multiplication (value * quantity)	10	5	50
5	Division (quantity / value)	10	5	2

Practical Application

Budgeting: Use SUM to total monthly expenses and AVERAGE to calculate spending trends.

Data Comparison: Employ MAX and MIN to identify the most and least expensive products.

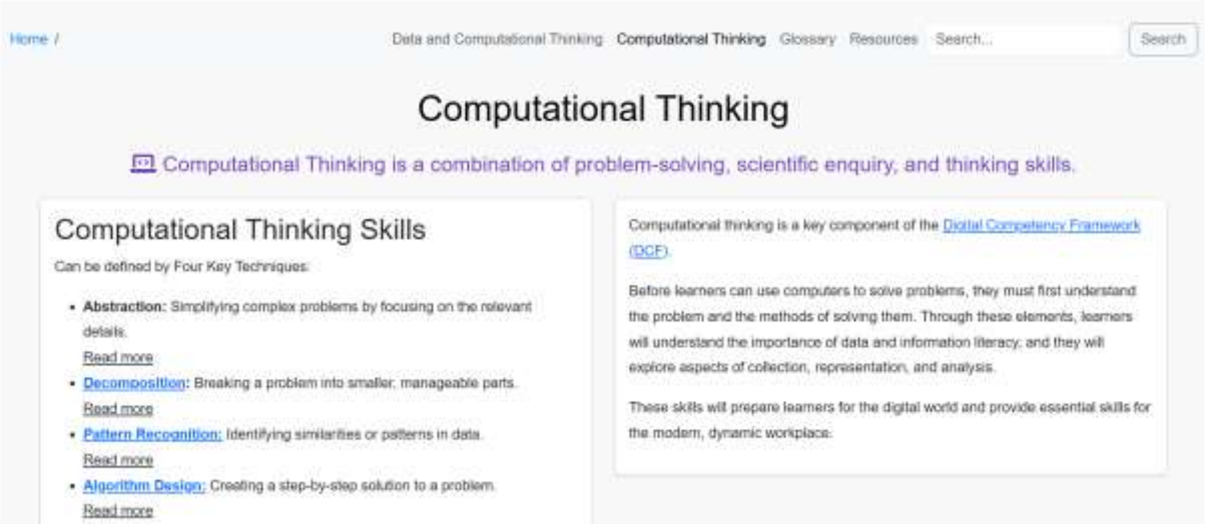
Visualising Data: Combine functions with chart tools to create engaging data visuals.

Check it out:

How to find the mean, median, mode and range

Resources

At Tech Edu: Teaching Spreadsheets



Computational Thinking

Computational Thinking is a combination of problem-solving, scientific enquiry, and thinking skills.

Computational Thinking Skills

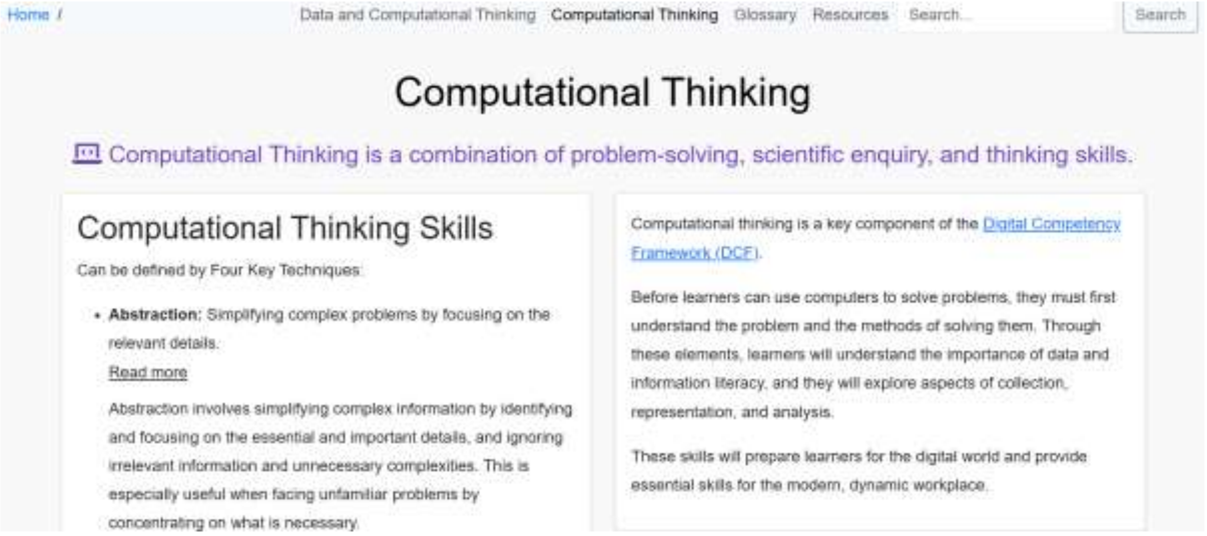
Can be defined by Four Key Techniques:

- **Abstraction:** Simplifying complex problems by focusing on the relevant details.
  Read more
- **Decomposition:** Breaking a problem into smaller, manageable parts.
  Read more
- **Pattern Recognition:** Identifying similarities or patterns in data.
  Read more
- **Algorithm Design:** Creating a step-by-step solution to a problem.
  Read more

Computational thinking is a key component of the Digital Competency Framework (DCF).

Before learners can use computers to solve problems, they must first understand the problem and the methods of solving them. Through these elements, learners will understand the importance of data and information literacy, and they will explore aspects of collection, representation, and analysis.

These skills will prepare learners for the digital world and provide essential skills for the modern, dynamic workplace.



Computational Thinking

Computational Thinking is a combination of problem-solving, scientific enquiry, and thinking skills.

Computational Thinking Skills

Can be defined by Four Key Techniques:

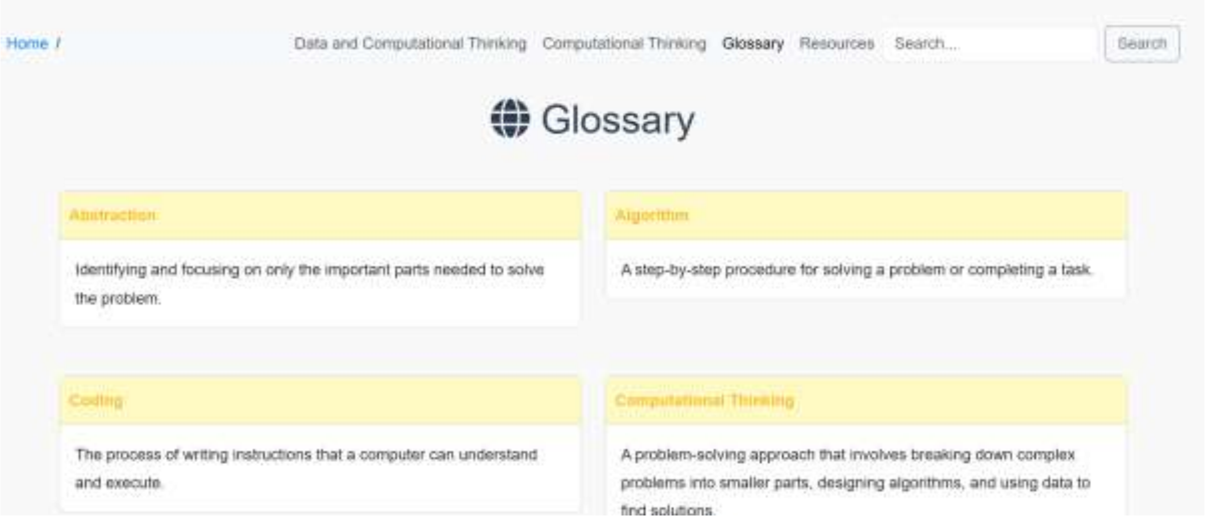
- **Abstraction:** Simplifying complex problems by focusing on the relevant details.
  Read more

  Abstraction involves simplifying complex information by identifying and focusing on the essential and important details, and ignoring irrelevant information and unnecessary complexities. This is especially useful when facing unfamiliar problems by concentrating on what is necessary.

Computational thinking is a key component of the Digital Competency Framework (DCF).

Before learners can use computers to solve problems, they must first understand the problem and the methods of solving them. Through these elements, learners will understand the importance of data and information literacy, and they will explore aspects of collection, representation, and analysis.

These skills will prepare learners for the digital world and provide essential skills for the modern, dynamic workplace.



Glossary

Abstraction

Identifying and focusing on only the important parts needed to solve the problem.

Algorithm

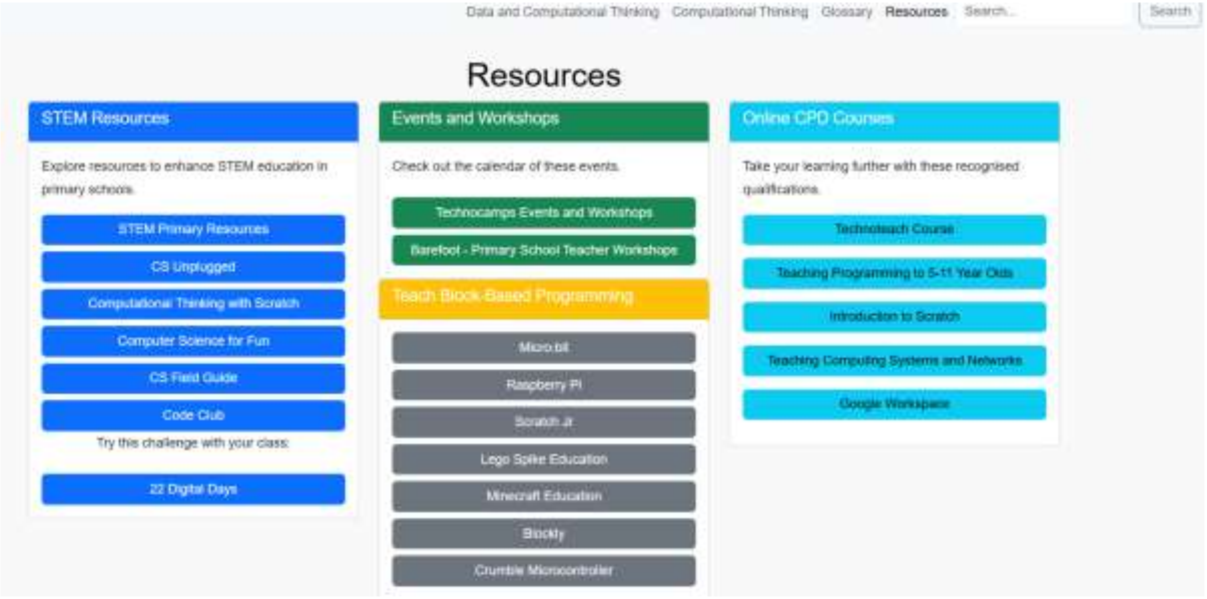
A step-by-step procedure for solving a problem or completing a task.

Coding

The process of writing instructions that a computer can understand and execute.

Computational Thinking

A problem-solving approach that involves breaking down complex problems into smaller parts, designing algorithms, and using data to find solutions.



Resources

STEM Resources

Explore resources to enhance STEM education in primary schools.

Try this challenge with your class:

Events and Workshops

Check out the calendar of these events.

Technocamps Events and Workshops
Barefoot - Primary School Teacher Workshops

Teach Block-Based Programming

Online CPD Courses

Take your learning further with these recognised qualifications.

Technoteach Course
Teaching Programming to 5-11 Year Olds
Introduction to Scratch
Teaching Computing Systems and Networks
Google Workspace