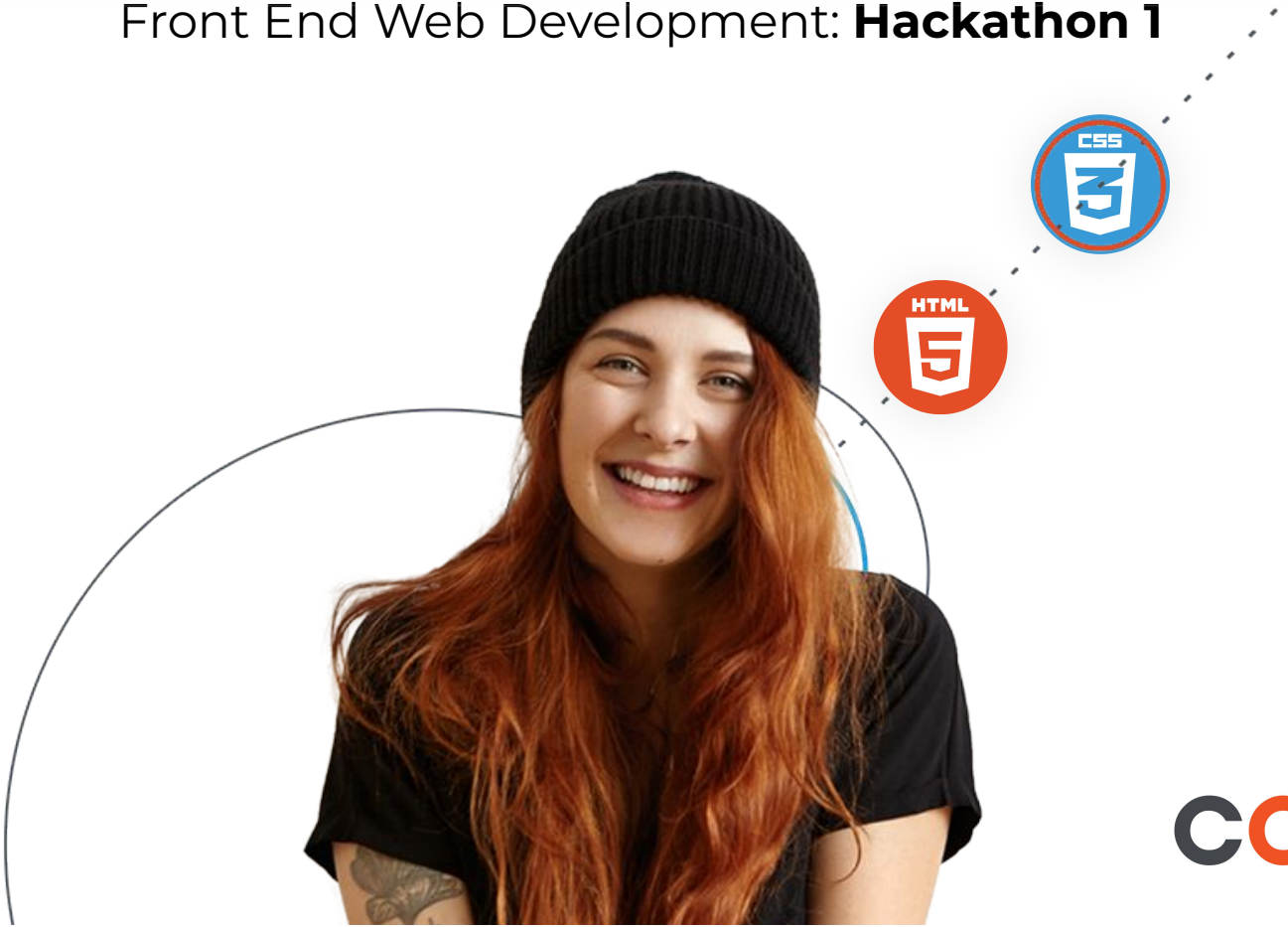
Front End Web Development: **Hackathon 1**



code institute

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Duration: 3 Days

Participant activities: Front-end development, UX Design, and Agile Project Management.

Introduction

Welcome to the 3-day hackathon focused on building static front-end websites using HTML and CSS. Ideally, each team consists of **three** participants who will collaboratively work on their projects. (Allowances will be made for numbers that don't fit the ideal member breakdown). The hackathon aims to enhance participants' skills in front-end web development, user experience design, and agile project management.

Each team member actively contributes to all three days, based on the principle of shared responsibility to ensure a balanced workload and a successful project outcome. The hackathon begins at 9:00, breaks for lunch at 12:00 for one hour and concludes at 5 pm each day to allow participants time for rest and preparation for the following day.

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Hackathon Outcomes

- Design a static Front-End web application using HTML and CSS based on the principles of user experience design, accessibility, and responsivity.
- Use an Agile methodology to plan and design a Front-End Web application.
- Test a Front-End web application through the development, implementation, and deployment stages.
- Maximise future maintainability through documentation, code structure, and organisation.
- Deploy a Front-End web application to a Cloud platform.
- Demonstrate and document the development process through a version control system such as GitHub.

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

High-Level Activities/Ceremonies

1. **Front-End Development**: the website's structure and styling using HTML and CSS.
2. **UX Design**: creating wireframes, and ensuring a user-friendly interface.
3. **Agile Project Management ceremonies and activities**: agile planning process, tracking progress, and ensuring the team meets deadlines.

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Hackathon Team Leads

The team leads for each project will be responsible for creating and managing the Hackathon project repositories, accepting and merging pull requests with assistance from the CI Support Staff and Tutor Support

The Project team leads are for this Hackathon are:

Joe Mellis | Vedran Lajic | Donal McCarthy | Katie Coughlan | Nathan Snoodyk | Nathan Keogh

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Sample Project Ideas (No JS or APIs allowed):

Project Example Idea 1: Travel Destination Guide

- External user's goal: Users want to gather information about popular travel destinations, attractions, and tips for an upcoming trip or out of general interest.
- Site owner's goal: The site’s goal is to provide comprehensive, visually appealing, and valuable travel-related content.
- Potential features to include:
    - Destination categories (e.g., beach, mountain, city destinations)
    - Detailed pages for each destination, with information on attractions, local food, accommodation, and travel tips
    - High-quality images to showcase the beauty of each destination
    - A responsive layout to ensure the site is easily navigable on all devices

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Project Example Idea 2: Recipe Collection

- External user's goal: Users want to find new recipes to try, with clear instructions and ingredient lists.
- Site owner's goal: The site’s goal is to provide a collection of enjoyable, easy-to-follow recipes.
- Potential features to include:
  - Categorization of recipes (e.g., by meal type, cuisine, dietary restrictions)
  - Each recipe page includes an ingredient list, step-by-step instructions, prep/cook time, and high-quality images
  - A responsive design to ensure the site is accessible and user-friendly on all devices

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Project Example Idea 3: Art Portfolio

- External user's goal: Users want to view an artist's work and potentially contact them for commissions or purchases.
- Site owner's goal: The site’s goal is to effectively showcase the artist's work and attract potential clients or buyers.
- Potential features to include:
  - A gallery section with high-quality images of the artist's work, categorized by medium or style if applicable
  - An 'About the Artist’ section with a bio and artistic philosophy
  - Contact information or form for inquiries or purchases
  - A responsive layout to ensure the site is easily navigable on all devices

Hackathon Teams

- **Travel Destination - Luxury Space**
  - *Joe Mellis -Lead*
  - *Cian Thornhill*
  - *Priyanka Savalge*
  - *Ossama Alawamenah*

Hackathon Team 1 | **Team Generative**

- **Art Portfolio - TBA**
  - *Vedran Lajic -Lead*
  - *Tadgh McCarthy*
  - *Jean Baptist Liberge*
  - *Archana Satpute*

Hackathon Team 2 | **Agile Avengers**

- **Recipe Collection - Frontend Drinks**
  - *Donal McCarthy -Lead*
  - *Gordon Meade*
  - *Mel maguire*

Hackathon Team 3 | **The Frontenders**

Hackathon Teams

- **Art Portfolio - Yayoi Kusama**
  - *Katie Coughlan -Lead*
  - *Nicole Elizabeth Rojas Valdes*
  - *Ryan Ferris*

Hackathon Team 4 | **The Code Collectors**

- **Art Portfolio - Brushes & Bytes**
  - *Nathan Snoodyk -Lead*
  - *Mark O'Flynn*
  - *Tuguldur Batsaikhan*

Hackathon Team 5 | **The IT Crowd**

- **Recipes - Bytesize**
  - *Nathan Keogh -Lead*
  - *Arianne Santiago*
  - *Vinicius Olveira*

Hackathon Team 6 | **Curious Coders**

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Day 1: Ideation, Design, Code Repository and Basic Site Structure Implementation:
Outcomes:

Activities & Expected Outcomes:

1. **Ideation Session:** Teams brainstorm and discuss potential project ideas that address real or imagined user needs.
2. **Project Proposal:** Each team creates a project proposal outlining the chosen idea, target audience, and main features.
3. **UX Design:** Participants create wireframes of their website layouts, demonstrating an understanding of user experience design principles.
4. **Agile Planning:** Teams set up a shared repository and project board using GitHub Projects defining user stories and mapping them to the project goals.
5. **Development Environment** and supporting tech in place.

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Activities Per Role Day 1

Agile Project Management

Day 1:

- All members contribute to the ideation session - all team members contribute project ideas.
- Discuss and select the final project idea, ensuring it aligns with the Hackathon guidelines.
- Team Lead creates a team project repository. (Upstream)
- Set up the project board using GitHub Issues/Projects or Trello creating and assigning user stories and supporting tasks.
- Each member except the Team Lead, creates a fork of the Upstream Repository for pull requests and merges.

UX Design

Day 1:

- Conduct user research and gather insights to understand the target audience and their needs.
- Create wireframes for the website's main pages, defining the layout and user flow.
- Determine an overall thematic style that visually represents the website's intended appearance, colours, and typography.
- Collaborate to ensure the feasibility of design concepts.

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Activities Per Role Day 1

Front-End Development

Day 1:

- Collaborate to understand the wireframes and design concepts.
- Create the initial HTML structure for the website, including the header, footer, and navigation menu.
- Set up the CSS file and apply basic styling to achieve the desired layout.
- Ensure responsiveness by using media queries to adjust the design for different screen sizes.
- Create reusable CSS classes for consistent styling across the website.
- Merge any branched changes resolving any conflicts

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Day 2: Further Implementation and Testing: Outcomes

Activities & Expected Outcomes:

1. Front-End Implementation: Each team member continues working on their assigned user stories.
2. Regular Check-ins: Hold regular check-ins to monitor progress on the Kanban board and resolve any issues.
3. Code Review: Teams conduct code reviews to ensure clean code, proper indentation, and consistent naming conventions.
4. Testing: Team members manually test the website on different devices and browsers to ensure responsiveness and bug-free functionality.
5. Submit pull requests and resolve any merge conflicts
6. Create the project README.md files to include a clear rationale for the development of the project, key project goals, and target audience. Include wireframes and supporting UX such as color, typography choices

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Activities Per Role Day 2

Agile Project Management

Day 2:

- Conduct regular check-ins to monitor progress and address any challenges.
- Ensure that each team member's activities are on track and assist in resolving any roadblocks.
- Keep track of completed tasks and update the project board accordingly.
- Document any tests carried out in the README file

UX Design

Day 2:

- Provide design assets, such as images and icons, required for the website's implementation.
- Ensure the design is accurately translated into HTML/CSS.
- Test the website's usability on different devices to ensure a consistent and intuitive user experience.
- Conduct usability testing and gather feedback for further improvements.

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Front-End Development

Day 2:

- Continue working on the HTML and CSS to implement the design details provided by the UX Design.
- Ensure that the website displays correctly on various devices and browsers by performing manual testing.
- Collaborate with the team during code reviews and make necessary improvements.
- Work on any additional features or components needed to enhance the website's functionality and aesthetics.
- Continue project documentation in the README.md file. Include reference any tests carried out
- Merge any branched changes, resolving any conflicts

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Day 3: Deployment and Presentation: Outcomes:

Activities & Expected Outcomes:

1. Deployment: Teams deploy their completed projects to a cloud-based hosting platform like GitHub Pages.
2. Presentation Preparation: Each team prepares a concise presentation showcasing their website's features and design decisions. Each team member will contribute to the presentation
3. Final Judging: Teams present their projects, and judges evaluate them based on User Experience, Design and Responsiveness, Code Quality and Documentation and Agile Process

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Activities Per Role Day 3

Agile Project Management

Day 3:

- Verify that all tasks are completed on the Kanban board and the website is ready for deployment.
- Coordinate the deployment process, making sure the website is hosted on the chosen cloud platform.
- All team members assist in the preparation of the final presentation, ensuring all aspects of the project are covered.
- Update the README.md file with comprehensive documentation, including the rationale behind the project and the agile planning process.

UX Design

Day 3:

- Review the website's design and verify that it aligns with the initial wireframes and mockups.
- All members assist in the preparation of the presentation, highlighting design decisions and user-centric features.
- Update the README.md file with details about the design process and the website's overall user experience.

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Front-End Development

Day 3:

- Finalise the implementation of the website, making any last-minute adjustments or fixes.
- Prepare the project for deployment on the chosen cloud platform.
- Ensure that the HTML/CSS code is well-organised, commented, and follows best practices.
- Update the README.md file with relevant information about the front-end development process.
- Merge any branched changes resolving any conflicts

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Behavioural focus

Solution-Oriented Mindset:

Capture:

During regular daily stand-up meetings, team members should share their achievements, progress, and challenges, focusing on maintaining a positive attitude.

Measure:

At the end of each day, team members should reflect on their input and experiences and plan how they can improve/maintain a solution oriented mindset.

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Task Estimation:

Capture:

Team members must estimate the effort required for each task during the ideation phase

Measure:

Team members track the actual time taken to complete each task and compare it with the initial estimates. This will assess the accuracy of task estimation.

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Personal Accountability:

Capture:

Team members are required to self-report their progress and contributions during daily stand-up meetings and on the project board.

Measure:

At the end of each day, team members must review their planned tasks and compare them with the completed tasks. Personal accountability will be measured based on the consistency and accuracy of reporting progress and task completion.

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Dealing with Distractions:

Capture:

During daily stand-up meetings, each member will observe and discuss any distractions they or other team members encounter.

Measure:

Team members must briefly describe the distractions they faced during the day and explain how they managed to refocus on their tasks. This will provide insights into their ability to handle distractions.

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Problem Solving:

Capture:

Team members are required to document in the README any challenges or obstacles they encounter and the problem-solving approaches they use during the implementation phase.

Measure:

The team will evaluate the effectiveness of problem-solving approaches used by each member based on the outcomes achieved. This assessment will be part of team discussions.

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Consistent Coding Practices:

Capture:

Code reviews are scheduled at the end of each day to ensure coding practices are followed.

Measure:

The success of consistent coding practices is measured based on the feedback received during code reviews.

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Agile Methodology Adoption:

Capture:

The adoption of Agile methodologies is documented through the usage of GitHub Projects tool for managing tasks.

Measure:

The effectiveness of Agile adoption is assessed based on the completion of tasks and the final success of the project as a minimal viable product.

Hackathon: Front-End Web Development: Build a Static Web Application using HTML and CSS

Judging Criteria:

- User Experience: How well the website addresses user needs and provides a pleasant browsing experience.
- Design and Responsiveness: Visual appeal, consistency, and responsiveness across different devices.
- Code Quality: Clean code structure, adherence to naming conventions, and proper use of HTML/CSS.
- Documentation and Agile Process: Quality of README.md file, version control usage, and effective agile planning and management.

Hackathon Schedule

Day 1: Ideation, Site Structuring, UX Design

9:00 am - 10:30 am
- All members:
  - Kick-off meeting and ideation session start.
  - Teams contribute project ideas that align with the hackathon guidelines.

10:30 am - 11:30 am
- Agile Project Management:
  - Setting up the team project repository.
  - Organizing the project board using GitHub Issues/Projects.

11:30 am - 12:30 pm
- UX Design:
  - Initial user research.
  - Sketching rough wireframes for the main pages of the website.

12:30 pm - 1:30 pm: Lunch Break

1:30 pm - 3:30 pm
- Front-End Development:
  - Creation of the basic HTML structure.
  - Initial CSS styling setup.

3:30 pm - 4:30 pm
- UX Design:
  - Wireframe refinement.
  - Deciding the thematic style of the website.

4:30 pm - 5:00 pm
- Review of Day 1 work and Day 2 prep.

Day 2: Further Implementation, Testing, and Code Review

9:00 am - 9:15 am
- Morning stand-up meeting for progress discussion.

9:15 am - 10:15 am
- UX Design:
  - Supplying design assets.
  - Ensuring accuracy in design translation.

10:15 am - 12:30 pm
- Front-End Development:
  - HTML and CSS design implementation continuation.
  - Manual device and browser testing.

12:30 pm - 1:30 pm: Lunch Break

1:30 pm - 2:30 pm
- Front-End Development:
  - Feature/component work.
  - README.md documentation updates.

2:30 pm - 3:30 pm
- UX Design:
  - Usability tests and feedback gathering.

3:30 pm - 4:30 pm
- Agile Project Management:
  - Update checks on the project board.
  - Progress check-ins.

4:30 pm - 5:00 pm
- Review of Day 2 work and Day 3 prep.

Day 3: Deployment and Presentation

9:00 am - 9:15 am
- Morning stand-up meeting for progress discussion.

9:15 am - 10:00 am
- Front-End Development:
  - Finalizing website implementation.
  - Preparing the project for cloud deployment.

10:00 am- 11:00am Presentation preparation

11:00 am - 12:30pm
- Agile Project Management:
  - Task completion verification.
  - Deployment process coordination.
  - Final checks and tweaks.

12:30 pm - 1:30 pm: Lunch Break

1:30 am - 2:00 pm
- Dry run for presentations.

2:00 pm - 5:00 pm
- Final presentations and project evaluations.

Hackathon 1 Outcomes: Detail

O1 Design a static Front-End web application using HTML and CSS based on the principles of user experience design, accessibility and responsivity

1.1 Design a website that incorporates a main navigation menu and a structured layout

1.2 Design a website that meets accessibility guidelines (e.g. contrast between background and foreground colours, non-text elements have planned alt text equivalents to cater for the visually impaired)

1.3 Design the organisation of information on the page following the principles of user experience design (headers are used to convey structure, information is easy to find due to being presented and categorised in terms of priority)

1.4 Ensure that foreground information is never distracted by backgrounds

1.5 Include graphics that are consistent in style and colour

1.6 Design the site to allow the user to initiate and control actions such as the playing of audio/video.

O1 Design a static Front-End web application using HTML and CSS based on the principles of user experience design, accessibility and responsivity

1.1
Design a website that incorporates a main navigation menu and a structured layout
1.2
Design a website that meets accessibility guidelines (e.g. contrast between background and foreground colours, non-text elements have planned alt text equivalents to cater for the visually impaired)
1.3
Design the organisation of information on the page following the principles of user experience design (headers are used to convey structure, information is easy to find due to being presented and categorised in terms of priority)
1.4
Ensure that foreground information is never distracted by backgrounds
1.5
Include graphics that are consistent in style and colour
1.6
Design the site to allow the user to initiate and control actions such as the playing of audio/video.

O2 Test a Front-End web application through the development, implementation and deployment stages

2.1
Create a website of at least three pages, or (if using a single scrolling page) at least three separate page areas to match the design and to meet its stated purpose
2.2
Write custom HTML code that passes through the official W3C validator with no issues.
2.3
Write custom CSS code that passes through the official (Jigsaw) validator with no issues
2.4
Incorporate images that are of sufficient resolution to not appear pixelated or stretched
2.5
Code all external links to open in a separate tab when clicked
2.6
Use CSS media queries across the application to ensure the layout changes appropriately and maintains the page's structural integrity across device screen sizes.
2.7
Use Semantic markup to structure HTML code
2.8
Present the finished website with clearly understandable site-specific content, rather than Lorem Ipsum placeholder text
2.9
Implement clear navigation to allow users to find resources on the site intuitively.

O3 Deploy a Front-End web application to a Cloud platform

3.1
Deploy a final version of the code to a cloud-based hosting platform (e.g. GitHub Pages) and test to ensure it matches the development version
3.2
Use Git & GitHub for version control of an interactive web application up to deployment.
3.3
Remove commented-out code before pushing final changes to version control and deploying.
3.4
Ensure that there are no broken internal links
3.5
Insert screenshots of the finished project in the README

O4 Maximise future maintainability through documentation, code structure and organisation

4.1
Write a README.md file for the web application that explains its purpose, the value that it provides to its users, and the deployment procedure.
4.2
Insert screenshots of the project features, give a brief description of what each feature does and explain its value to the user.
4.3
Attribute all code from external sources to its original source via comments above the code and (for larger dependencies) in the README.
4.4
Clearly separate and identify code written for the website and code from external sources (e.g. libraries or tutorials)
4.5
Organise HTML and CSS code into well-defined and commented sections
4.6
Place CSS code in external files linked to the HTML page in the HEAD element.
4.7
Write code that meets at least minimum standards for readability (consistent indentation, blank lines only appear individually or, at most, in pairs)
4.8
Name files consistently and descriptively, without spaces or capitalisation to allow for cross-platform compatibility.
4.9
Group files in directories by file type (e.g. an assets directory will contain all static files and may be organised into sub-directories such as CSS, images, etc.)

O5 Demonstrate and document the development process through a version control system such as GitHub

5.1
Use consistent and effective markdown formatting to produce a README file in English that is well-structured, easy to follow, and has few grammatical errors.

06 Use an Agile methodology to plan and design a Front End Web application

6.1
Use an Agile tool to manage the planning and implementation of all significant functionality.
6.2
Document and implement all User Stories and map them to the project within an Agile tool