Front-End Web Development

Unit 16: Walkthrough project

Course Outline

1. Getting Started
2. HTML - Structuring the Web
3. CSS - Styling the Web
4. JavaScript - Dynamic client-side scripting
5. CSS - Making Layouts
6. Introduction to Websites/Web Applications
7. CSS – Advanced
8. Reviewing Progress
9. JavaScript - Modifying the Document Object Model (DOM)
10. Dynamic HTML
11. Web Forms - Working with user data
12. JavaScript - Advanced
13. Building a Web Application with JavaScript
14. Introduction to CSS Frameworks – Bootstrap
15. Other Frameworks, SEO, Web security, Performance
16. **Walkthrough project**

Course Learning Outcomes

- Competently write HTML and CSS code
- Create web page layouts according to requirements using styles
- Add interactivity to a web page with JavaScript
- Access and display third-party data on the web page
- Leverage Bootstrap and Static Site Generator

Assessment

- Final Project - 100% of the grade

  - Design and Build functioning Website using HTML5, CSS (including Bootstrap), JavaScript (browser only)

  ✓ Code will be managed in GitHub
  ✓ Website will be deployed to GitHub Pages
  ✓ All code to follow best practice and be documented

  - Details and How-To-Guide are available on the course page under the section called Assessments

16. Walkthrough Project

In This Unit

Title
Walkthrough Project

Walkthrough Project

Instructions

The goal of the assignment is to demonstrate how you apply course concepts, models, tools and practices in your own work to demonstrate the course learning outcomes:

- Write clean and efficient HTML and CSS code
- Create web page layouts and styling to address customer requirements
- Apply interactivity to a web page with Javascript
- Use Bootstrap and similar web design aids
- Incorporate content/data from third-party REST API’s using AJAX
- Design, construct and deploy real-world static responsive websites

Objective

- Develop a full-featured website using Astro and Svelte (optional), incorporating a functional feature from the ideagiven or a feature of your own
- The key is to ensure proper integration between frontend and the website content, integration with third-party API’s
- deploy the application to GitHub pages
- demonstrate strong communication and software engineering best practice

Potential Ideas

1. **Task Management App**

- Users can create tasks, set due dates, mark tasks as completed, and search, filter tasks based on status or priority
- All data should be stored locally in the browser or local file system

2. **Weather Dashboard**

- Fetch weather data (REST API) based on user input (e.g., location) and display current conditions, forecasts

3. **GitHub Repository Explorer**

- Allow users to search for GitHub repositories and display repository details like stars, forks, and open issues

4. **Movie Database**

- Browse and search movie listings

5. **Interactive Map**

- Allow users to add and view annotated data on an interactive online map using a third-party library, e.g leaflet.js

You may also propose your own idea, but ensure it fits the scope and ambition of these feature ideas. Whether you use your own idea or one of the ideas above, you must develop a feature-rich website

Project Details

Front-end

- **Home Page** - A landing page related to your chosen idea
- **Feature-specific pages** - This will depend on your chosen idea (e.g., display weather data, etc.)
- **Third-party Libraries & Frameworks** - Third-party JavaScript libraries which provide functionality to support your functional idea are allowed but exclude the use of `JQuery`

Front-end Framework (Optional):

- **Integrate Svelte** for the front-end. Use third-party REST API's to be consumed by Svelte

Back-end

- **Server** - Utilise static html for all website content and JSON for any data files

Features

1. Idea-specific features

- Depending on your chosen idea (e.g., CRUD tasks, display weather data, etc.). The use of local data persistence (Browser/File system) is encouraged

2. Responsive Design

- Design elements should fluidly adjust to fit various screen sizes, ensuring a seamless user experience across devices

3. API Integration

- Integrate at least one external API relevant to the chosen project theme. This could be for data retrieval, enhancing functionality, or providing additional features to users
- For instance, a Weather Dashboard might use an API like OpenWeatherMap, while a Movie Database system might interface with The Movie Database API for movie details

Features

4. Agile Development Processes

- Use GitHub for managing all content, code and documentation for the project
- Regular commits with descriptive commit messages. Use feature branches and tags
- Use GitHub features such `Issues`, `Projects`, etc… for managing project requirements and user feedback

5. Deployment

- Continuous deployment to the hosting platform, i.e. `GitHub Pages` or `Render.com`

Final Presentation

- You will record and submit a final presentation of a maximum of 5 minutes along with your project
- This final presentation describes your entire project, its design and main features
- It should take advantage of the previous project review and feedback received throughout the course
- Your video presentation will be graded according to the rubric (https://learn.ucdpa.ie/mod/resource/view.php?id=149176) under the presentation and documentation category

Software to record Presentation

- Screencast software comparision -> https://screenrec.com/screen-recorder/screencast-software/
- OBS Studio -> https://obsproject.com/download

There is a maximum limit in Moodle on the submitted file can be so if required you can use software like Handbrake to reduce the video file size

- https://handbrake.fr/downloads.php

Submission Checklist

- Your code is properly structured and follows best practices
  - Code guide for HTML and CSS -> https://codeguide.co/
  - Code guide for JavaScript -> https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide
  - Project structure for Astro -> https://docs.astro.build/en/basics/project-structure/
  - Project structure for Svelte -> https://kit.svelte.dev/docs/project-structure
  - Best practices for GitHub:
    - https://docs.github.com/en/repositories/creating-and-managing-repositories/best-practices-for-repositories
    - https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/about-readmes
- You have implemented additional features or design enhancements to showcase your creativity and skills
- Include a README file in the root of the GitHub repository which should include the following:
  - The URL to the deployed website/app
  - An overview of the website/app
  - Comments to explain each non-trivial function/method
  - https://medium.com/@vndpal/how-to-create-the-perfect-documentation-for-your-code-344ebe301c22

Submission Checklist

- Your website is accessible and fully functional on the provided URL. Provide a copy of the accessible report for the website
  - https://usabilitygeek.com/10-free-web-based-web-site-accessibility-evaluation-tools/
  - https://www.accessibilitychecker.org/
  - https://accessibe.com/accessscan
- Your website is performant and follows web best practice. Provide a copy of the lighthouse report
- Use https://validator.w3.org/ and https://jigsaw.w3.org/css-validator/ validators to check your `HTML` and `CSS` code
- You double-checked the deployment instructions and any necessary configuration details to ensure a smooth deployment process
- All project documentation and reports (accessibility and performance) should be saved in a “docs” folder in either markdown or html formats. If a report can not be downloaded, include a screenshot of the summary part of the report
- Any additional details or considerations specific to the deployment in your submission
- Verified the hosted website’s functionality and appearance after deployment to ensure it matches your local development version

To submit your project and final presentation:

- Compress all your project files into a single ZIP file, including the website files and the deployment configuration files and clear instructions on how to deploy and access the website on your chosen hosting provider. Do not include the video file in this ZIP
- Upload the video file, uncompressed, along with the ZIP file
- Once the two files are uploaded, submit your assessment

Course Recap

1. ~~Getting Started~~
2. HTML - Structuring the Web
3. CSS - Styling the Web
4. JavaScript - Dynamic client-side scripting
5. CSS - Making Layouts
6. Introduction to Websites/Web Applications
7. CSS – Advanced
8. ~~Reviewing Progress~~
9. JavaScript - Modifying the Document Object Model (DOM)
10. Dynamic HTML
11. Web Forms - Working with user data
12. JavaScript - Advanced
13. Building a Web Application with JavaScript
14. Introduction to CSS Frameworks – Bootstrap
15. Other Frameworks, SEO, Web security, Performance
16. ~~**Walkthrough project**~~

Final Questions

Activity

Breakout

- Join a breakout room
- Continue working on remaining exercises/project/assessment
- Lecturer will visit each room in turn, etc...

Summary

Completed this Week

- Walkthrough of Project
- Course Recap

For next Three Weeks

- Best of luck with the Project
- Post any questions you may have to the discussion board
- Don't forget to attend the Wednesday Coding Sessions