BIT607

ASSESSMENT

2

Set A ver. 2.0

BIT607 Web Development
Assessment 2

Weighting

30%

Learning outcomes

2 Apply HCI principles to design and implement a web application using frameworks and techniques that focus on responsive design, user experience (UX), and the usability and accessibility of the application.

4 Implement, evaluate and use a range of tools and techniques required for the development of a web application.

Instructions

Complete and submit your assessment according to the Open Polytechnic's Assessments webpage. This includes information on academic integrity, formatting, word limits and referencing.

- Include your name, student number and the assessment number.
- Number your pages.

Submission

- Submit your assessment in two files:
  - 1 x Microsoft Word document
  - 1 x Zipped web application code files.
- Submit your work through your iQualify course.
- Emailed assessments will not be accepted.
- You will receive an automated notice advising you of your successful submission.

By submitting your assessment, you confirm that it is your own original work.

Introduction

Across your three assessments for BIT607 Web Development, you will design, build and analyse a web application.

In Assessment 2, you will:

- plan and analyse how you will make your web application accessible, responsive and usable
- develop a responsive web application from the requirements and planning you completed in Assessment 1
- discuss two techniques that could be used to improve your web application's load time.

Task 1: Design planning and analysis

Explain **two** things your development work will include to ensure your application is accessible, responsive and usable and has a positive user experience. How would you test to see if this has been achieved?

(Word count guideline: 300 words)

(10 marks)

Task 2: Develop a web application

Part 1: Development

Use only HTML and CSS to develop your web application locally to implement the wireframes created in Assessment 1. Do not use a framework or CMS such as Bootstrap, W3.CSS or WordPress.

It is recommended that you use Visual Studio Code. Ensure you have the following web pages and content:

- Home (an image)
- Services (a table containing services and prices)
- Gallery (a 4 x 4 grid of images)
- Booking (a form containing inputs for owner's name, email address, dog's name, breed and age, and a submit button)

- Contact (a map using Google Maps or similar service, a form containing inputs for name, email address, phone number and comment, and a submit button).

Note: The images must be royalty-free. **Also, the forms are NOT needed to be functional.**

Your completed web application must meet the following requirements:

- Use HTML to implement your web pages' header, nav, section, footer, table and form. The nav must contain Snazzy Paws Grooming. The footer must contain Snazzy Paws Grooming's opening hours.
- Use CSS to style your web pages. These styles must be stored in an external CSS file. Minify the external CSS file once you are happy with the style of your web application. Link the external CSS file in your web pages.
- Follow accessibility guidelines such as headings, alt text, style, colour and ARIA.
- Apply responsive design principles to make the web application mobile responsive.
- Apply clean, intuitive design that considers HCI and UX principles.
- HTML is indented correctly.

(60 marks)

Part 2: Hosting pages

Your manager has asked you to host your web application on GitHub Pages (or any service online or in your local server).

- Create a new repository on GitHub (or a folder on the chosen service online or in your local server).
- Upload your HTML, CSS and images to this repository. Ensure your home web page is named index.html.
- Provide a URL to your web application on GitHub Pages (or any service online or the local URL address in your local server).

(5 marks)

Part 3: Validating your HTML and CSS

- Validate your web application hosted on GitHub Pages (or any service online or in your local server) using an online HTML and CSS validation service.

- Provide a screenshot of each page being validated.

(10 marks)

Part 4: Performance and speed testing

- Performance-test all web pages in your web application on GitHub Pages (or any service online or in your local server) using a service such as Google Page Speed Insights. You can choose any service you can find to achieve it.
- Speed-test the contact web page in your web application (or any service online or in your local server) using a service such as Pingdom Website Speed Test. Test from five locations.
- Provide a screenshot of each page being tested.

(10 marks)

Part 5: Content delivery networks (CDNs) and caching

Explain how CDNs and caching could improve your web application’s load time.

(Word count guideline: 200 words)

(5 marks)

Marking schedule

Task 1: Design planning and analysis	8–10 marks	6.5–7.5 marks	5–6 marks	4–4.5 marks	1–3.5 marks
Explain two things to ensure the application's accessibility, responsiveness, usability and positive user experience.	Detailed explanation of two specific measures for each aspect (accessible, responsive, usable, positive user experience). Clear and comprehensive testing methods to verify achievement.	Clear explanation of two measures for each aspect. Testing methods are clear and address how to verify achievement.	Adequate explanation of two measures for each aspect but may lack detail or specificity. Adequate testing methods to verify achievement.	Explanation of measures for each aspect is somewhat clear but lacks detail and specificity. Testing methods are basic but present.	Explanation of measures for each aspect lacks clarity or is incomplete. Testing methods are unclear or incomplete.
Task 2: Develop a web application Part 1: Development	**8–10 marks**	**6.5–7.5 marks**	**5–6 marks**	**4–4.5 marks**	**1–3.5 marks**
HTML structure and contents	HTML structure accurately implements wireframes for Home, Services, Gallery, Booking and Contact pages with correct use of <header>, <nav>, <section>, <footer>, <table> and <form> elements. Includes all required content as specified.	HTML structure correctly implements wireframes for required pages but with one or two minor errors or omissions in <header>, <nav>, <section>, <footer>, <table> or <form> elements. May have minor issues with content inclusion.	HTML structure adequately implements wireframes for required pages but with minor errors or omissions in <header>, <nav>, <section>, <footer>, <table> or <form> elements. Minor issues or consistencies with content inclusion.	HTML structure partially implements wireframes for required pages, with noticeable errors and omissions in <header>, <nav>, <section>, <footer>, <table> or <form> elements. Content has issues and consistencies with content inclusion.	HTML is structured poorly or is unable to implement wireframes for required pages. Major errors and omissions in <header>, <nav>, <section>, <footer>, <table> or <form> elements. Significant amount of content is missing.
Task 2: Develop a web application Part 1: Development	**4–5 marks**	**3.5 marks**	**2.5–3 marks**	**2 marks**	**1–1.5 marks**
External CSS file	CSS styles are stored in an external file, properly linked in all web pages (<link> tag). File is minified for optimised loading.	CSS styles are stored in an external file and linked in web pages, but not minified. Minor issues with linking or minification.	CSS styles are stored in an external file but with significant errors in linking or minification. Styles are not fully externalised.	CSS styles are partially stored in an external file or not linked properly. Minification is missing. Major errors in linking or styling approach.	CSS styles are not stored in an external file or not linked properly. Styles are embedded within HTML or missing entirely.
Task 2: Develop a web application Part 1: Development	**8–10 marks**	**6.5–7.5 marks**	**5–6 marks**	**4–4.5 marks**	**1–3.5 marks**

Accessibility	Follows accessibility guidelines, including appropriate use of headings (<h1> to <h6>), alt text for images, style and colour contrast, and ARIA attributes where applicable.	Mostly follows accessibility guidelines, but with minor issues in headings, alt text, style contrast or ARIA attributes.	Partial adherence to accessibility guidelines. Noticeable issues in headings, alt text or style contrast or lack of ARIA attributes.	Accessibility guidelines are not consistently followed. Major issues in headings, alt text or style contrast or lack of ARIA attributes.	Minimal adherence to accessibility guidelines. Significant issues in headings, alt text or style contrast or lack of ARIA attributes.
Task 2: Develop a web application Part 1: Development	**12–15 marks**	**10–11.5marks**	**7.5–9.5 marks**	**6–7 marks**	**1–5.5 marks**
Responsive design	Applies responsive design principles effectively, ensuring optimal display and usability across different devices (desktop, tablet, mobile).	Adequately applies responsive design principles with one or two minor issues in adaptation across different devices.	Partial application of responsive design principles. Issues in adaptation across different devices are noticeable.	Responsive design principles are not effectively applied. Major issues in adaptation across different devices.	Poor application of responsive design principles. Design does not adapt or is unusable on different devices.
Task 2: Develop a web application Part 1: Development	**8–10 marks**	**6.5–7.5 marks**	**5–6 marks**	**4–4.5 marks**	**1–3.5 marks**
Clean and intuitive design	Effectively designs a clean, intuitive interface thoroughly considering HCI and UX principles.	Effectively designs a clean, intuitive interface adequately considering HCI and UX principles.	Designs a clean, intuitive interface, evidently considering HCI and UX principles.	Designs a minimal interface with limited consideration of HCI and UX principles.	Designs a cluttered or unintuitive interface with poor consideration of HCI and UX principles.
Task 2: Develop a web application Part 1: Development	**8–10 marks**	**6.5–7.5 marks**	**5–6 marks**	**4–4.5 marks**	**1–3.5 marks**
HTML indentation	HTML code is consistently and correctly indented throughout all web pages, enhancing readability and maintainability.	HTML code is mostly indented. Minor inconsistencies may exist but do not significantly affect readability or maintainability.	HTML code is partially indented, with noticeable inconsistencies that somewhat affect readability or maintainability.	HTML code has noticeable inconsistencies in indentation, affecting readability and maintainability.	HTML code indentation is inconsistent or largely absent, making code difficult to read and maintain.
Task 2: Develop a web application Part 2: Hosting pages	**4–5 marks**	**3.5 marks**	**2.5–3 marks**	**2 marks**	**1–1.5 marks**

Create a new repository on GitHub (or folder on any service online or in your local server).	Repository or folder created correctly. All HTML, CSS and images are uploaded correctly, with the home page named index.html. URL to the web application is provided and works perfectly.	Repository or folder created correctly. Most HTML, CSS and images are uploaded correctly. The home page is named index.html. URL is provided and works with minor issues.	Repository or folder created but with one or two minor issues in accessing. Most HTML, CSS and images are uploaded but with one or two minor issues. The home page is named index.html. URL is provided but may have issues.	Repository or folder created but with significant errors in access settings. Some HTML, CSS and images are uploaded correctly. The home page is named index.html. URL is provided but has issues.	Repository or folder created with many significant errors in access settings or file uploads. The home page might not be named index.html. URL is provided but does not work correctly.
Task 2: Develop a web application Part 3: Validating your HTML and CSS	**8–10 marks**	**6.5–7.5 marks**	**5–6 marks**	**4–4.5 marks**	**1–3.5 marks**
Validate your HTML and CSS in your web application hosted on GitHub Pages (or any service online or in your local server). Provide a screenshot of each validation performed.	All web pages validated using an online HTML and CSS validation service. Clear screenshots of each page being validated are provided, showing no errors or warnings.	All web pages validated with one or two minor issues using an online HTML and CSS validation service. Clear screenshots of each page being validated are provided, showing warnings.	Most web pages validated with several minor issues using an online HTML and CSS validation service. Clear screenshots of each page being validated are provided, showing a few errors and warnings.	Some web pages validated, but with notable issues using an online HTML and CSS validation service. Screenshots are provided but show many errors and warnings.	Few web pages validated with significant issues. Screenshots are unclear or show major errors.
Task 2: Develop a web application Part 4: Performance and speed testing	**8–10 marks**	**6.5–7.5 marks**	**5–6 marks**	**4–4.5 marks**	**1–3.5 marks**
Performance and speed testing web pages in your web application hosted on GitHub Pages (or any service online or in your local server) using Google Page Speed Insights (or any service you can find to achieve it). Provide a screenshot of each test performed.	All web pages performance tested. Contact web page speed tested from five locations using Pingdom Website Speed Test. Screenshots of each page being tested from both services are provided, showing comprehensive results.	All web pages performance tested. Contact web page speed tested from five locations using Pingdom Website Speed Test. Screenshots of each page being tested from both services are provided, showing clear results.	Most web pages performance tested. Contact web page speed tested from five locations using Pingdom Website Speed Test. Screenshots of each page being tested from both services are provided, showing adequate results.	Some web pages performance tested. Contact web page speed tested from fewer than five locations using Pingdom Website Speed Test. Screenshots are provided but may be incomplete or unclear.	Few web pages performance tested. Contact web page speed tested from only one or two locations using Pingdom Website Speed Test or not tested at all. Screenshots are unclear or incomplete.

Task 2: Develop a web application Part 5: Content delivery networks (CDNs) and caching	4–5 marks	3.5 marks	2.5–3 marks	2 marks	1–1.5 marks
Explain how CDNs and caching could improve load time.	Comprehensive and clear explanation of how CDNs and caching work and their impact on improving load time. Provides specific examples and technical details.	Clear explanation of how CDNs and caching work with a good understanding of their impact on load time. Some technical details and examples provided.	Adequate explanation of how CDNs and caching work with a general understanding of their impact on load time. Limited technical details and examples.	Limited explanation of how CDNs and caching work with minimal understanding of their impact on load time. Few technical details or examples provided.	Vague or unclear explanation of how CDNs and caching work with little to no understanding of their impact on load time. Minimal technical details or examples provided.