

COMP229 – Advanced Web Application Development

Final Team Project - Value 30%

Part 1 (Project Concept, Landing Page & Site Security) – 28 Marks – Value 10%

- **HomeController related Views:** Landing Page, About Page, Contact Page (3 Marks: GUI, 3 Marks: Functionality).
- Initial Site Security Setup and Views Cleaned up (2 Marks: GUI, 2 Marks: Functionality).
- **Site Structure:** Responsive Site, Site Style, Assets Folder Setup (4 Marks: Site Structure).
- **Internal Documentation:** Comment headers on each page (2 Marks: Internal Documentation).
- **External Document:** Company Logo, Table of Contents, Detailed App Description, Colour and Typography, and Wireframes (6 Marks: External Documentation).
- Initial GitHub Repo Created & Cloud Provider Setup (2 Marks: Version Control, 4 Marks: Cloud Provider).

Due Week # 12 (Friday December 2), 2016 @ midnight

Feature	Description	Marks
GUI / Interface Design	HomeController related Views: Landing Page, About Page, Contact Page Initial Site Security Setup and Views Cleaned up.	5
Functionality	HomeController related Views: Landing Page, About Page, Contact Page Initial Site Security Setup and Views Cleaned up.	5
Site Structure	Responsive Site, Site Style, Assets Folder Setup	4
Internal Documentation	Comment headers on each page	2
External Documentation	Company Logo, Table of Contents, Detailed App Description, Colour and Typography and Wireframes.	6
Version Control	Initial GitHub Repo Created	2
Cloud Deployment	Deploy site to Cloud Service.	4
Total		28

Submission Details for Part 1:

Your submission should include:

1. First Draft of your external document (MS Word or PDF).
2. A link to your GitHub Repo (Part 1 Only).
3. A link to your cloud provider live site (preferably MS Azure).
4. A zip archive of your website's Project files (for Part 1) submitted to blackboard

Please see next page for Project 2 - Part 2

Part 2 (Main Functionality & Database Connectivity) – 36 Marks – Value 10%

- Product Catalog Page **First Draft** (2 Marks: GUI, 3 Marks: Functionality)
- Product Details Page **First Draft** (2 Marks: GUI, 3 Marks: Functionality)
- Shopping Cart Page **First Draft** (4 Marks: GUI, 6 Marks: Functionality).
- **Users** should be able to Browse the **Product Catalog Page**, **Add Items** from the **Product Catalog Page** to the **Shopping Cart**, and **Modify the Shopping Cart** before Checkout (3 Marks: GUI, 3 Marks Functionality).
- Checkout and Order Confirmation Pages **First Draft** (2 Marks: GUI, 3 Marks: Functionality).
- **Site Structure:** Sample Products setup in the Database with appropriate Table structure (2 Marks: Site Structure).
- **Internal Documentation:** ensure you include class and method comment headers for all c# code; all code uses contextual variable names (2 Marks: Internal Documentation).
- **Version Control:** GitHub repo updated with commits during development (1 Mark: Version Control).

Due Week # 13 (Friday December 9), 2016 @ midnight

Feature	Description	Marks
GUI / Interface Design	Product Catalog, Product Details, Shopping Cart, Checkout and Order Confirmation Pages Setup with basic styling – First Draft – to promote functionality. Users should be able to Browse the Product Catalog Page, Add Items from the Product Catalog Page to the Shopping Cart, and Modify the Shopping Cart before Checkout	13
Functionality	Product Catalog, Product Details, Shopping Cart, Checkout and Order Pages Setup with most functionality – First Draft . Users should be able to Browse the Product Catalog Page, Add Items from the Product Catalog Page to the Shopping Cart, and Modify the Shopping Cart before Checkout	18
Site Structure	Sample Products setup in the Database with appropriate Table Structure.	2
Internal Documentation	Class and Method comment headers for all c# code; all code uses contextual variable names	2
Version Control	GitHub Repo Updated with Commits during development	1
Total		36

Submission Details for Part 2:

Your submission should include:

1. A link to your GitHub Repo (Part 2 Only) – recommend creating a separate repo.
2. A link to your cloud provider live site (preferably MS Azure) for Part 2 only – recommend creating a separate site.
3. A zip archive of your website's Project files (for Part 2) submitted to blackboard

Please see next page for Project 2 - Part 3

Part 3 (Finished Version – Fully Styled and Functional) – 26 Marks – Value 10%

- Product Catalog Page **Final Draft** (2 Marks: GUI, 1 Marks: Functionality)
- Product Details Page **Final Draft** (2 Marks: GUI, 1 Marks: Functionality)
- Shopping Cart Page **Final Draft** (4 Marks: GUI, 2 Marks: Functionality).
- Checkout and Order Confirmation Pages **Final Draft** (2 Marks: GUI, 1 Marks: Functionality).
- **Site Security** updated to prevent non-registered users from Checking Out – non-registered users are redirected to appropriately styled Login Page. (1 Mark: GUI, 1 Mark: Functionality).
- Include Additional **Image assets** as required for your site to look stylish and complete (1 Mark: Site Structure).
- Additional Products in database to promote the look and feel of a more complete Store prototype. Validation attributes added to models to improve useability (2 Marks: Site Structure).
- Confirm all code is **error free** through rigorous testing (2 Marks: Site Structure).
- **Internal Documentation:** Include inline comments that describe your GUI Design and Functionality (1 Mark: Internal Documentation).
- **External Documentation** Updated: Screen shots for all pages added, Complete Future Functionality section (2 Marks: External Documentation).
- Final Version of **GitHub repo** that includes updated commits related to final touches and application updates (1 Mark: Version Control)

Due Week #14 (Friday December 16, 2016) @ midnight.

Feature	Description	Marks
GUI / Interface Design	Product Catalog, Product Details, Shopping Cart, Checkout and Order Confirmation Pages cleaned up with appropriate Layout and Style, non-registered Users are directed to appropriately styled Login page when attempting to Checkout – Final Draft	11
Functionality	Product Catalog, Product Details, Shopping Cart, Checkout and Order Pages include full functionality, Site Security prevents non-registered users from checking out – Final Draft	6
Site Structure	Additional Image assets included for products, icons etc. as required. Additional products in database to promote the look and feel of a more complete store prototype. Validation Attributes added to models to improve useability. Confirm all code is error free through rigorous testing	5
Internal Documentation	Include inline comments that describe your GUI Design and Functionality	1
External Documentation	External Document Updated with Web Application Screen Shots; Future Features section completed.	2
Version Control	Final Version of GitHub repo that includes updated commits related to final touches and application updates.	1
Total		26

Submission Details for Part 3:

Your submission should include:

1. Final Draft of your external document (MS Word or PDF).
1. A link to your GitHub Repo (Part 3 Only) – recommend creating a separate repo.
2. A link to your cloud provider live site (preferably MS Azure) for Part 3 only – recommend creating a separate site.
3. A zip archive of your website's Project files (for Part 3) submitted to blackboard

Final Team Project

Maximum Mark: 90

Overview: Working in a group (up to 4 members) or on your own and utilizing your accumulated knowledge of the ASP.NET MVC create a mock e-commerce Web Application. The Web Application will mostly feature a **catalog** of items you will be selling that share a common theme. You may use the “Fantasy” restaurant from Assignment 2 as a starting point for this project. You will be required to create additional pages to support the application such as a **Contact Page**, an **About Page**, and other pages used for Product Management.

Project Instructions:

(29 Marks: GUI, 29 Marks: Functionality, 11 Marks: Site Structure, 5 Marks: Internal Documentation, 8 Marks: External Documentation, 4 Marks: Version Control, 4 Marks: Cloud Hosting)

1. The **Landing Page** and pages that can be accessed as an **Anonymous** user **(14 Marks: GUI, 14 Marks: Functionality)**:
 - a. Anonymous users should be able to access all Pages associated with the **HomeController** including the **Landing Page**, **About Page** and **Contact Page**. Anonymous users should also be able to **Browse** the **Product Catalog Page**, **Add Items from the Product Catalog Page** to the **Shopping Cart**, and **Modify** the **Shopping Cart** before **Checkout** (6 Marks: GUI, 6 Marks: Functionality).
 - b. The store’s products will be displayed on the **Product Catalog Page** in either a **List** or **Card** format. Both formats will display the **name of the product**, a **thumbnail image** of the product, a **short description**, the **Product Price**, and an optional **Add to Cart Button**. If the user clicks the **Name Label** or the **thumbnail image** he will be redirected to the **Product’s Detail Page**. If the optional **Add to Cart Button** exists for each Product, clicking it will add the Product to the **Shopping Cart** (4 Mark: GUI, 4 Marks: Functionality).
 - c. Your **Product Details Page** displays details for a **single Product**. It includes the **Product Name**, a **full-size image** of the Product, A **longer description** of the Product, and the **price** of the Product. A back button (or other control) should also be included to take the user back to the **Product Catalog Page**. An **Add to Cart Button** should also be included. Click on the Add to Cart Button will add the Product to the **Shopping Cart** (4 Marks: GUI, 4 Marks: Functionality).
2. **Shopping Cart** Functionality **(8 Marks: GUI, 8 Marks: Functionality)**:
 - a. The **Shopping Cart** will keep track of all items added from the **Store Catalog**. This does not have to persist between logins, however (3 Marks: GUI, 3 Marks: Functionality).
 - b. The **Shopping Cart** will display the Totals for all items Added, including taxes (2 Mark GUI, 2 Mark: Functionality).
 - c. The **Shopping Cart** will allow users to change the quantity of each Product that has been added to it (2 Marks: GUI, 2 Marks: Functionality).

- d. The **Shopping Cart** will include a **Checkout Button** that will redirect registered users to the Checkout Page. If a user is browsing Anonymously it will redirect them to the Login / Register pages (1 Mark: GUI, 1 Mark: Functionality).
- 3. **Checkout and Order Confirmation Pages** (4 Marks: GUI, 4 Marks: Functionality).
 - a. Registered users may elect to **Checkout** after they are satisfied with their selections. Checking out will allow the user to select a mock **payment option** (e.g. Credit, Debit or COD). It is assumed that any payment option selected is valid for this application. Users will **not be required** to enter credit card or debit card or other sensitive information (2 Marks: GUI, 2 Marks: Functionality).
 - b. Once the **payment option** is selected the user may click a **Finalize Order Button** which will redirect them to the **Order Confirmation Page** that confirms that their order has been accepted and that they will receive their products within a specified timeframe (2 Marks: GUI, 2 Marks: Functionality).
- 4. **User Management and Site security (3 Marks: GUI, 3 Marks: Functionality):**
 - a. **User Registration** must be included. A form will allow the user to enter profile information (**email address, password**), which will be stored in a SQL database structure (1 Marks: GUI, 1 Marks: Functionality).
 - b. The user will be able to **Login, Logout** and **reset** his password (1 Mark: GUI, 1 Mark: Functionality).
 - c. **Site security** will prevent any non-registered users from **checking out** or entering secure areas of the site (1 Mark: GUI, 1 Mark: Functionality).
- 5. **Site Structure. (10 Marks: Site Structure):**
 - a. **Responsive Design.** Your Site must be **responsive**, and adapt to various viewport sizes. **Note:** Using a mobile-first framework such as Bootstrap is highly recommended (1 Mark: Site Structure).
 - b. **Style and Layout.** Your **CSS** rules reside in separate file(s) in their own folder and adhere to style and layout best practices. You may use SASS for additional functionality and layout control (2 Marks: Site Structure).
 - c. **Assets Folder.** Your **images and multimedia assets** are contained in their own subfolder of the **Assets** root folder and appropriately linked to your site. **Note:** will need to provide your own image assets for your site that may include thumbnail images, product images, carousel images, custom icons, etc. (2 Marks: Site Structure).
 - d. **Error Free Code.** All Your Code (C#, HTML, CSS, JavaScript, jQuery, etc.) is error free (2 Marks: Site Structure).
 - e. **SQL Database.** Your MS SQL database must be hosted online. **Note:** It is recommended that you use Microsoft Azure as your service provider (4 Marks: Site Structure).
- 6. **Internal Documentation (comments) - (5 Marks: Internal Documentation):**
 - a. Ensure you include a **comment header** for each of your **C#, HTML, CSS, and JavaScript files** that indicate: The **File name, Author's name(s), Student Number(s), web site name, file description** (2 Marks: Internal Documentation).

- b. Ensure you include a **section header** for all of your **C# Classes, C# Methods, HTML structure, CSS style sections**, and any **JavaScript functions** (1 Marks: Internal Documentation)
 - c. Ensure all your code uses **contextual variable names** that help make the files human-readable (1 Marks: Internal Documentation).
 - d. Ensure you include **inline comments** that describe your GUI Design and Functionality. **Note:** please avoid “over-commenting” (1 Marks: Internal Documentation)
7. Create an **External Document** for your Web App that includes **(8 Marks: External Documentation)**:
- a. A **company Logo** (1 Marks: External Documentation).
 - b. **Table of contents** (1 Marks: External Documentation).
 - c. A **Detailed description** of your Web App including its core functionality (1 Mark: External Documentation).
 - d. **Colour & Typography Section** that includes Colour swatches for your site and documents the fonts you will be using for your navigation, headlines, main body, link text, etc. (1 Mark: External Documentation).
 - e. A **Wireframes Section** that include a wireframe image and appropriate arrows and labels for each page template of your Web App. Wireframes for simple pages such as your About Page and Contact Page will **not be required** for this Web App (2 Marks: External Documentation)
 - f. **Screen Capture Section** that includes Screen Shots (samples) of each of your site’s templates. (1 Marks: External Documentation).
 - g. **Potential Future Functionality** – a section that describes features that could be added to your app but do not as yet exist given the time constraints. (1 Mark: External Documentation).
8. Share your files on **GitHub** to demonstrate Version Control Best Practices and push your site to a cloud host **(4 Marks: Version Control, 4 Marks: Cloud Hosting)**.
- a. Your repository must include **your code** and be well structured (2 Marks: Version Control).
 - b. Your repository must include **commits** that demonstrate the project being updated at different stages of development – each time a major change is implemented (2 Marks: Version Control).
 - c. You must deploy your site to your Cloud Server using **git** (4 Marks: Cloud Hosting).

Please See Next Page for Full Project Rubric

PROJECT RUBRIC

Feature	Description	Marks
GUI / Interface Design	Display elements meet requirements. Appropriate spacing, graphics, colour, and typography used.	29
Functionality	Site deliverables are met and site functions are met. No errors, including submission of user inputs.	29
Site Structure	Your site files are well organized. Your HTML and CSS are kept in separate folders. All external documents and code are appropriately linked to your site. You code is minified where possible. Your code is error free.	11
Internal Documentation	File header present, including site & student name & description. Functions and classes include headers describing functionality & scope. Inline comments and descriptive variable names included.	5
External Documentation	An external document (MS Word or PDF format) has been created that includes a company logo, table of contents, brief site description, etc.	8
Version Control	GitHub commit history demonstrating regular updates.	4
Cloud Deployment	Deploy site to Cloud Service.	4
Total		90

This assignment is weighted **30%** of your total mark for this course.

All Assignments are due at the beginning of class.

Late submissions:

- 10% deducted for each day late.

External code (e.g. from the internet or other sources) can be used for student submissions within the following parameters:

1. The code source (i.e. where you got the code and who wrote it) must be cited in your internal documentation.
2. It encompasses a maximum of 10% of your code (any more will be considered cheating).
3. You must understand any code you use and include documentation (comments) around the code that explains its function.
4. You must get written approval from me via email.