Homework 02 – HTML, CSS, JavaScript, and Responsive Web Design
Out of 48 Marks

DUE: 2022-05-08 before 23:59

IMPORTANT NOTES:
- This is an individual assignment.
- Homework assignments are based on assessment objectives. If an objective has been achieved a mark will be allocated.

INSTRUCTIONS:
- In this assignment, you will be given high-level requirements and can implement them in a context you see fit.
- **Source Code:** Zip your source code files together and name it **uXXXXXXXX_HW02.zip**, where the XXXXXXXX is your student number, e.g., u12345678_HW02.zip.
- **Video Demo:** DO NOT Zip your Video Demo. Name the video demo **uXXXXXXXX_HW02.mp4**, where the XXXXXXXX is your student number, e.g., u12345678_HW02.mp4.
- **Upload instructions will be amended after the Version Control Session.**
- The notes are posted so student who are keen can start to work on the assignment until the upload instructions are updated after the session on Version Control has been presented.
- For uploading your Video demo, follow the instructions on the "How to upload a video to YouTube Guide.pdf" document posted on the Study Guide Page.
- Share the YouTube link as a URL on ClickUP for the video submission.
- For Visual Studio Code applications (which are small), you can still post the source code to ClickUP.
- When starting to work with Visual Studio 2019 Community Edition, follow the instructions on the "How to use GitHub in Visual Studios 2019 Guide.pdf" document posted on the Study Guide Page.
- You can upload your source code to GitHub for this assignment however it is not compulsory. As long as we have your source code, that is what is of importance. If you this assignments source code to GitHub, please share the details in the Source Code Upload section for the assignment.

NOTE ON DEMO SOFTWARE
- Please change the compression ratio and Frames Per Second (FPS) of the desktop recoding software that you use to reduce the file size of your demo. These details have been noted at the top of the homework submission page.
- Please use desktop recoding software as suggested on the homework submission page.
- DO NOT use your phone to record your demo as it is extremely difficult to follow what is going on in your demo and it creates unnecessarily large files making the video demo upload problematic. We suggested desktop recording software as it simply streamlines the process and creates smaller files that are easier to upload.

SUBMISSION DEADLINE: 2022-05-08 before 23:59
- There shall be no extensions to the deadline.
- If homework submissions are uploaded too late, then upload errors WILL happen.
- Do not wait to the last minute to complete the assignment.
- Start working on the assignment as soon as it is posted.
- Verify the completeness of your upload.
- There are multiple upload opportunities enabled if your upload is incomplete.
- Incomplete uploads will be considered unsubmitted work.
- E-mail submissions WILL NOT be accepted.
- Late submissions WILL NOT be accepted.
- **NO EXCEPTIONS WILL BE MADE FOR ANYONE.**

IMPORTANT

- We download your program source code to make sure that you do not upload empty programs.
- In the past we have encountered people demoing other people's programmes. That is considered academic dishonesty and there are severe consequences if you are found to have committed academic dishonestly. Please refer to the student code of conduct regarding the consequences of academic dishonestly.

1. TOOLS, SKILLS, KNOWLEDGE REQUIRED:

- To complete this homework assignment, you should make use of HTML, CSS, JavaScript, and Responsive web design.
- You will need to make use of Bootstrap for the content and Layout of the pages.
- You will need to make use of Font Awesome Icons for all the icons that need to be used in this assignment.

2. USE CASE:

- For this assignment you will be creating a movie ticket booking system. You will need to code two main sections:
    - Home page: The home page will show all the movies available for booking.
    - Cart page: The cart page will show all the movie tickets that have been added to the cart in a table format. You will need to code functionality to allow the user to increase and decrease the quantity of tickets per movie and the functionality to remove a movie from the table.
- You will need to make use of an **array** of movie **objects** that needs to be in the following form:

```javascript
let movies = [
  {
    id: number, //a unique identifier for this movie
    title: string, //the title of this movie
    director: string, //the name of the director of this movie
    runtime: string, //the runtime of this movie
    release_year: number, //the year this movie was released
    description: string, //a description of this movie
    poster_url: string, //a URL or file path to an image of the poster for this movie
    cinema_number: number, //the number of the cinema this movie will be showing in
    ticket_price: number, //the price of a single ticket for this movie
    tickets_in_cart: number //the number of tickets that have been booked for this movie
  }
];
```

- You will need to include at least four movies in the object. This object should be hard coded inside your JavaScript file. The movies you choose to include is up to you.
- You will need to make use of **localStorage** to persist all the movie tickets that have been added to the cart. This means that the data on the page is not lost when the page is refreshed. The following **keys** and **values** will need to be stored in **localStorage**:
    - moviesInCart – the **value** of this **key** will be the movie objects that have been added to the cart
    - totalCost – the **value** of this **key** should be the total cost of all the movie tickets that have been booked.
    - numberInCart – the **value** of this **key** should be the total number of tickets that have been added to the cart.
- You will need to make use of **Bootstrap Components** and **Layout** in this assignment to make all the pages **responsive**.
- You will need to make use of **Font Awesome Icons** in the assignment.

3. STANDARD REQUIREMENTS:

The following is a list of <u>mandatory requirements</u>. See the images provided for examples of what your assignment could look like.
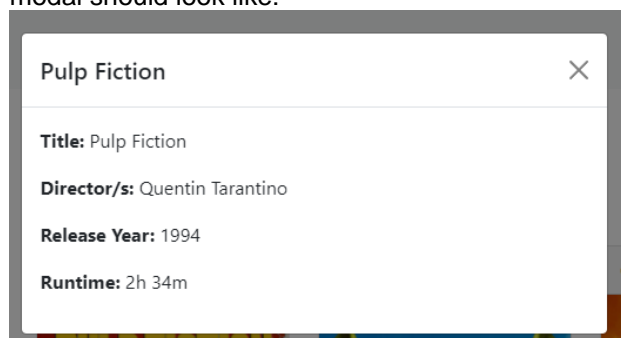
- **Nav bar**
  - HTML and Bootstrap Requirements
    - You must make use of a **Bootstrap NavBar**.
    - You will need to show the same NavBar on all pages of the movie ticket booking system.
    - You need a 'home' image that will return to the home page on click on the right side of the NavBar.
    - You need a **Font Awesome icon** on the left side of the NavBar that will go the Cart page on click.
    - You need an element to show the number of movie tickets added to the cart next to the cart icon on the left side of the NavBar.
  - JavaScript Requirements
    - If the user clicks on the 'cart' icon on the left of the NavBar they should be taken to the cart page.
    - If the user clicks on the 'home' image on the right of the NavBar they should be returned to the home page.

See the following image for an example of what the NavBar could look like:

- **Home page**
  - HTML and Bootstrap Requirements
    - A heading for the home page.
    - Bootstrap Cards should be used to show the movies available for booking.
    - The Bootstrap Grid system should be used to complete the **layout** of the page and make the page **responsive**. This means that the cards will re-align themselves as the size of the display gets smaller. See the example video provided to see what this should look like.
  - JavaScript Requirements
    - The movie cards as seen in the image of the home page below should not be hard coded. You must make use of **JavaScript** and the **movie object array** (as discussed in the Use Case section of the document) to display all the cards **dynamically** on the page.
    - The movie cards should contain the following sections:
      - Header
        - The headers should display the **cinema number** of the movie.
      - Image
        - The image should the **poster** of the movie (this should not be hard coded but read from the **movie object array**)
      - Body
        - The body should display the **title**, **description,** and the **price** of a single ticket of the movie.
      - Footer
        - The footer should have two bootstrap buttons:
          - Show Details button
            - If the user clicks on the 'Show Details' button of one of the cards, a Bootstrap **modal** should be shown that will show that movie's information. See the image below to see what the modal should look like:

- Book Ticket button
  - If the user clicks on the 'Book Ticket' button, the **number** next to the cart icon should be **incremented** and the **localStorage** should be **updated** to include the three **keys** and **values** as discussed in the Use case section of this document. Each time the 'Book Ticket' button is clicked these **values** should be **updated**.

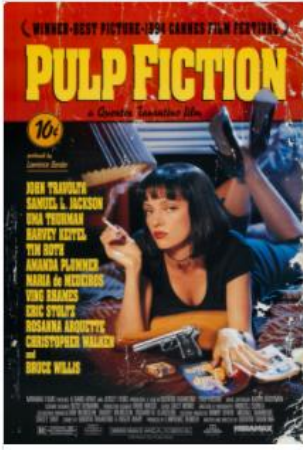See the image below for an example of what the home page could look like:

- **Cart Page**
    - HTML and Bootstrap Requirements
        - A heading for the cart page
        - A Bootstrap table should be used on the cart page with the following **columns**:
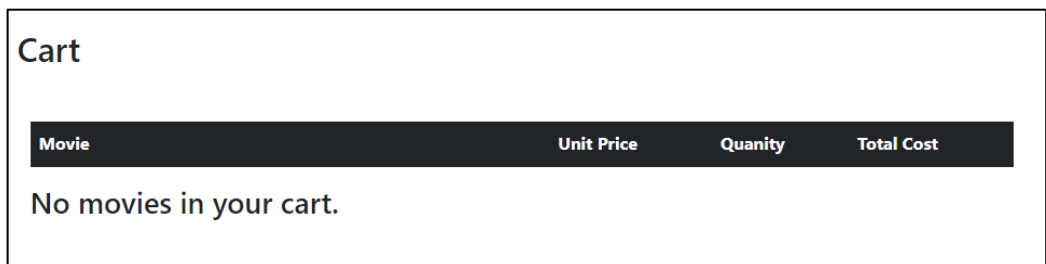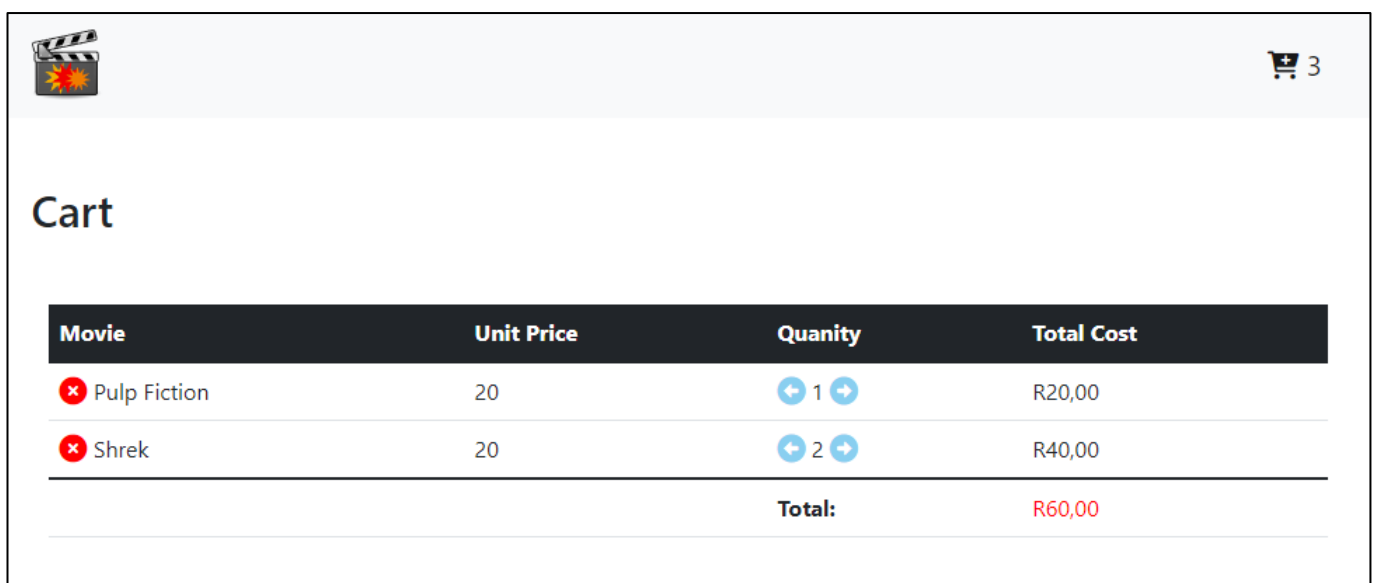   - Movie
            - Unit Price
            - Quantity
            - Total Cost (this is the total cost of the number of tickets for the specific movie)
        - A footer in the table to display the **Total** of **all** the movie tickets in the cart
        - A **Font Awesome Icon** that can be used to **remove** a movie from the cart (see the image below of the cart page to see what is expected)
        - An **Increase** and **Decrease Font Awesome Icon** that can be used to increase and decrease the **quantity** of movie tickets in the cart (see the image below of the cart page to see what is expected)
    - JavaScript Requirements
        - If there are no movies tickets in the cart a message should be displayed to inform the user. See the image below for an example of what this could look like:

- If the user clicks on the **remove icon** (the red cross as seen in the image of the cart page below) the entire movie with all it's tickets should be removed from the **table**. This also means:
    - The **cart number** next to the cart icon in the NavBar must be **updated** accordingly.
    - The **Total** in the footer of the table should be **updated** accordingly.
- If the user clicks on the increase or decrease **icons** (the blue arrows as seen in the image of the cart page below) the quantity in the **table** should be increased or decreased accordingly. This also means that:
    - The **cart number** next to the cart icon in the NavBar must be updated accordingly.
    - The **Total Cost** column in the table should be updated accordingly.
    - The **Total** in the footer of the table should be updated accordingly.
    - If the **quantity** reaches zero, the entire movie should be **removed** from the cart (i.e., the table).

See the image below for an example of what the cart page could look like:

4. RESEARCH / READING REQUIREMENTS

The following is required reading and / or research.

- localStorage
- Bootstrap Grid System
- Bootstrap NavBar
- Bootstrap Card
- Bootstrap Modal
- Bootstrap Table
- Font Awesome Icons

Homework 02 – HTML, CSS, JavaScript, and Responsive Web Design
Out of 48 Marks

DUE: 2022-05-08 before 23:59

VIDEO INSTRUCTIONS:
- Make sure that everything is running when you start recording the video. The video should not be longer than **13** minutes showing the items in the Checklist.
- <mark>**Simply scrolling through the program will be considered not presenting a demo which would equal to you immediately losing 50% of your mark.**</mark> **You need to** <u>explain your reasoning</u>**.** It is part of what is known as "reflective learning" and helps you solve problems when you encounter then during coding. It is extremely useful. Do not neglect your own leaning process by skipping out on this important step
- When showing something in the Checklist, show us your code and explain why you did it in that way. If you prefer to demo everything first then explain code, you can do it that way, but we need an explanation of why you coded it that way and not just functionality. The explanation does not need to be too in-depth.
- If something did not work in your code, in the video explain to us what you wanted to do and what you wanted to achieve with your approach. **We will not mark your code that you show us unless you explain this**.

IMPORTANT NOTES:
- Both the Video Demo and Homework Source Code should be submitted in the correct upload area. <mark>**If the Video demo or the Homework Source Code is missing, you will get a zero.**</mark>
- If files are upload to the wrong upload area, we will not go and look for the upload. Uploads should be submitted correctly as indicated online. Incorrect uploads will lead to a zero being allocated.
- If you are caught for plagiarism, we will give you 0 and you will be reported for plagiarism immediately. There are no more warnings this year. We will audit historical assignments throughout the semester. You are here to learn, the more you know, the better for you.

Checklist Use the checklist as a script that would allow you to sequence your demonstration.	MAX
• **Home – Functionality** • Adding movie to localStorage (4: 4 marks for adding movie object to localStorage) • Adding the number of items in the cart to localStorage (2) • Adding the total cost of the movie tickets in the cart to localStorage (2) • Displaying movie cards dynamical (3: 3 marks for dynamically displaying movie cards using the products object i.e., do not hardcode the cards) • Displaying number of TOTAL movie tickets next to the cart icon (3: 1 mark for displaying the total next to the cart icon, 2 marks for incrementing the number when another movie ticket is added) • Passing and displaying the correct movie information in the modal (do not hard code the information) (4: 4 marks for opening the modal and displaying the correct movie's information in the card on the correct button click) • Clicking on the "home" icon in navbar returns to the home index (1) • Clicking on the cart icon opens the cart index (1) **Demonstration time allocation = 5 minutes**	**20**
1. **Cart – Functionality** • Total price displayed in table (1: 1 mark for reading the total from localStorage) • Movie tickets added to cart displayed correctly in table (4: 2 Marks for retrieving correct information from localStorage, 2 marks for displaying the information in the table correctly) • Remove movie booking from table on click of icon (3: 1 mark for removing the information from the table, 1 mark for updating the price total in the table, 1 mark for updating cart total in nav bar)	**15**

Checklist Use the checklist as a script that would allow you to sequence your demonstration.	MAX
• Increase and decrease the quantity of a movie ticket (if the quantity is zero then the item should be removed from the table) (7: 1 for increasing quantity, 1 for decreasing quantity, 1 for mark updating cart total, 1 mark for updating the total cost on increase and decrease, 2 marks for removing the item form the table when quantity is zero, 1 mark for updating the localStorage on each change) **Demonstration time allocation = 5 minutes**	
2. Bootstrap – Styling • Bootstrap navbar (cart icon and "home icon" displayed on opposite sides as shown) (2) • Bootstrap card used (2: 2 marks for using bootstrap header, footer and bootstrap buttons in card) • Bootstrap grid system used to make cards responsive when resizing page (watch demo video example) (4: 2 marks for alignments of cards, 2 marks for cards resizing correctly) • Bootstrap modal (1: 1 mark for using bootstrap modal) • Bootstrap table used on cart index (2: 2 marks for using bootstrap table) • Font awesome icons used (2: 2 marks for using font awesome icons in table and in navbar for the cart) **Demonstration time allocation = 3 minutes**	**13**
TOTAL MARK ALLOCATION	**48**