**COMP 2511 Assignment #3:   
Creating a Responsive Web Page**

*Due March 23, 2017 by midnight*

*Add: design the background*

***Outcomes***

The intent of this assignment is to design, from scratch, the HTML, CSS and graphics for a simple web based card game. As such, you need to show you can:

* apply responsive design ideas to CSS/HTML pages.
* create a separate design for mobile and large displays using the same HTML.
* use media queries to determine which styles to apply based on browser size.
* create appropriate graphics from scratch for mobile & desktop versions of the site.
* create and style several HTML forms.
* create a visually appealing and usable website using valid HTML5.

***Submitting***

Submit all your work in a single folder that includes your username as the folder name, for instance nkhemka\_assignment3. Inside this folder, you will have sub-folders for the html, graphics, and style files. You may choose to submit a zip archive of your assignment.

When confident it is ready, drag the entire folder to the Submit Drive (I:). If submitting from an off-campus computer, use https://secure.mtroyal.ca to submit your files.

**You will lose marks if you do not follow these submission instructions.**

***Grading***

Your assignment will be graded based on the overall design and usability of the webpage, the quality and appropriateness of the graphics, the correct validation of HTML, HTML forms and CSS. Above all else, your site must apply responsive design techniques so mobile and desktop versions of your page look different, and demonstrate that your design decisions address the unique considerations of large and small screens/high and low bandwidth.

***Overview of Requirements:***

Your HTML webpage must:

1. Validate as HTML5 via <http://validator.w3.org/>.
2. Contain all of the forms and game areas as illustrated in the wireframe in figure 1 below. These areas include:
   * header (consists of the title) and footer (consists of your name and assignment number),
   * well-styled areas with a form to enter names (there are only two players, player 1 and player 2) and request a new game,
   * an area for the player’s scores,
   * an area indicating which player’s turn it is and the “value to reach” field,
   * a "game board" where the ten cards are laid out,
     + (*About cards*: you will have five cards that you will design in Photoshop or GIMP. These cards are only allowed values between 0 and 4 (therefore, each card is displayed twice in the wireframe). You will have pseudo-classes for hover and active.)
   * an area with “end turn” button and “end game” button.

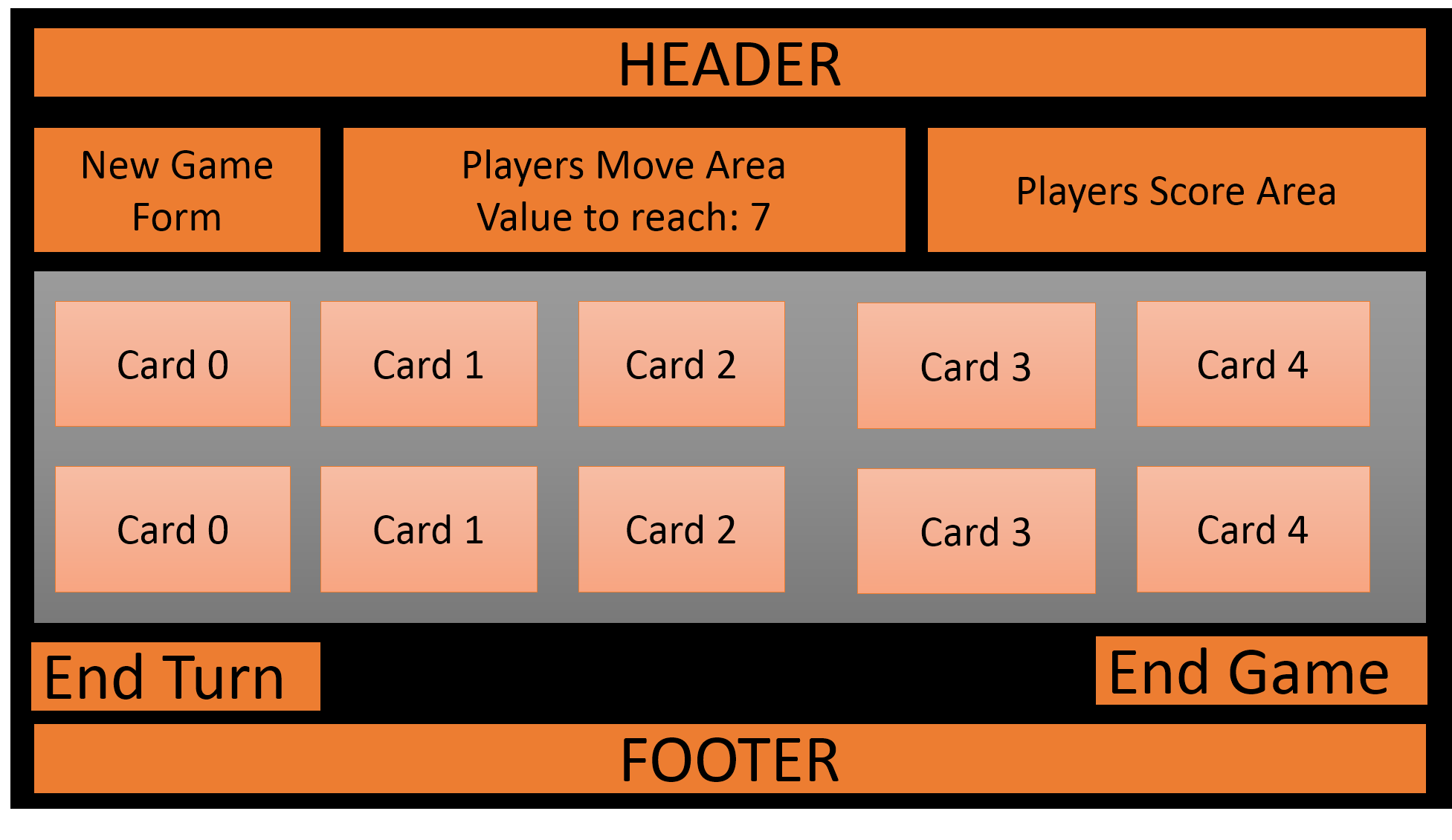


Figure Wireframe example for one way a desktop game board could be laid out. Mobile must be different...

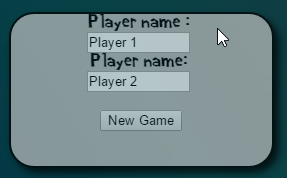
1. Have a significantly different look for mobile and desktop applications. Each layout and design will be given distinct marks for how well the needs of that screen size are addressed.

***More Detailed Requirements***

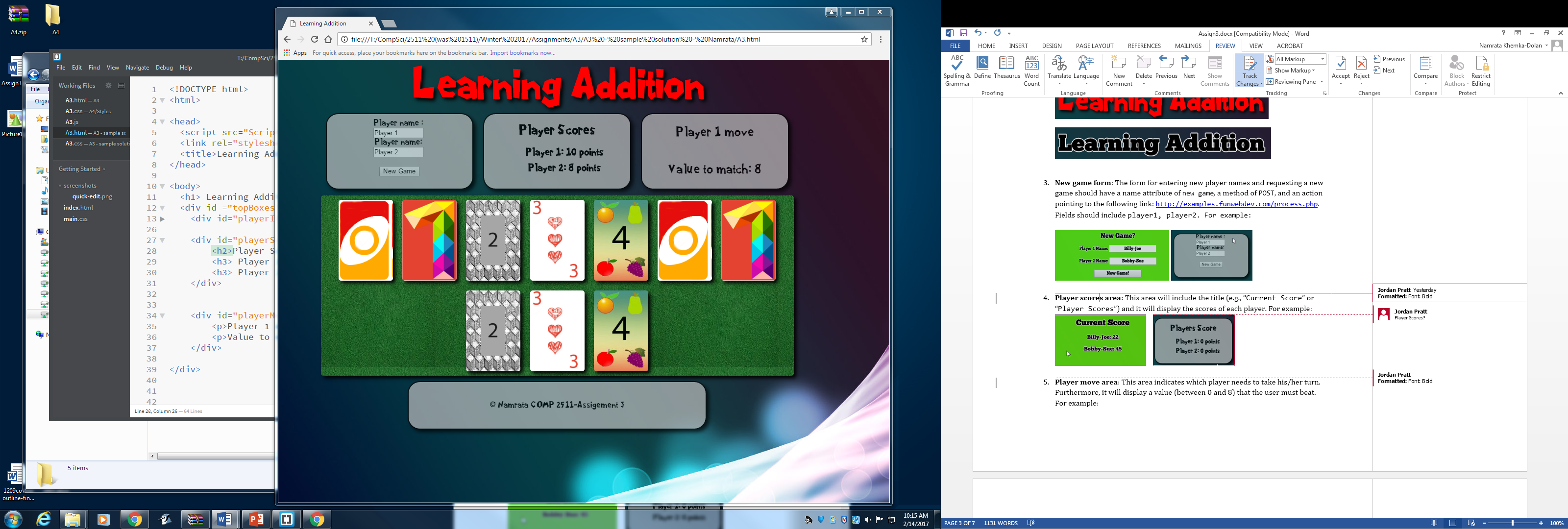
1. Use a media breakpoint to style for mobile devices: assume sizes under 480px are mobile.
2. **Background**: A background of any color of your choice or an image that you have created in Photoshop/GIMP.
3. **Header**: The header should display the game’s name. You can choose your name for the game. For example, if you chose “Learning Addition”, your header might look like this:  
   



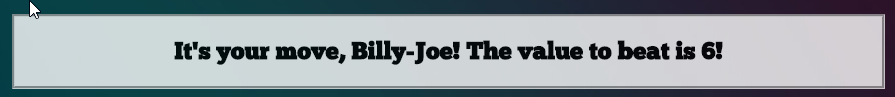
1. **New Game Form**: The form for entering new player names and requesting a new game should have a name attribute of new game*,* amethod of POST, and an action pointing to the following link: <http://examples.funwebdev.com/process.php>*.* Fields should includeplayer1, player2. For example:

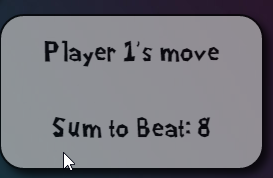
 

1. **Players Scores Area**: This area will include the title (e.g., “Current Score” or “Player Scores”) and it will display the scores of each player. For example:

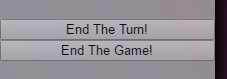
 

1. **Player Move Area**: This area indicates which player needs to take his/her turn. Furthermore, it will display a value (between 0 and 8) that the user must beat. For example:





1. **End Area**: This area will have “End Turn” and “End Game” buttons. For example:

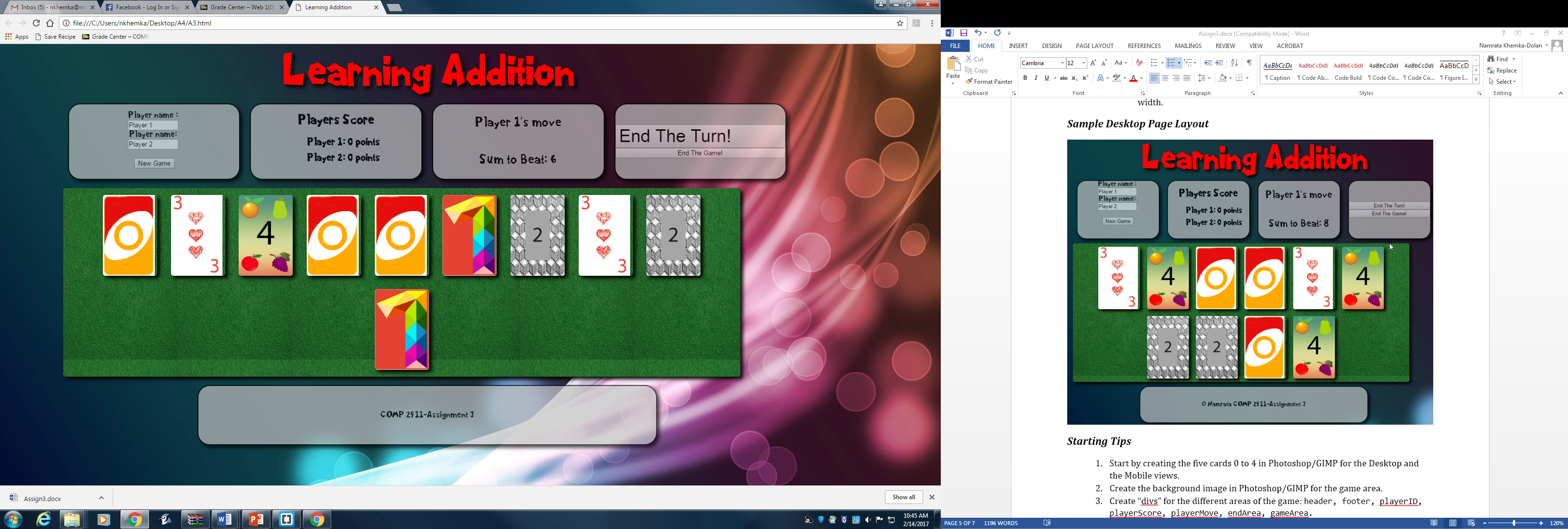
 

1. **Footer**: The footer will include your name and the assignment number.

1. **The Game Area**:
   1. Must be distinct from the rest of the page. In the example below, I have created an image in Photoshop that looks like green felt. You will be designing yours in either Photoshop or GIMP.
   2. You will design 5 cards in Photoshop or GIMP for the *Desktop* view. These cards will have 5 different designs that depict values 0 to 4. You will also design 5 cards in Photoshop or GIMP for the *Mobile* view. The Mobile view cards will have less details than the Desktop view cards. Note that you must create and design your own cards. Do not use the card designs in the example below.
   3. A pseudo-class must be used for when the player hovers (i.e., pseudo-class hover) over the cards. This is to make the card look a little distinct from the others. This can be as simple as applying a box-shadow to the card.
   4. A pseudo-class must be used for when the card is clicked (i.e., pseudo-class active). This can be as simple as changing the border style and width.

***Sample Desktop Page Layout***



***Starting Tips***

1. Start by creating the five cards 0 to 4 in Photoshop/GIMP for the Desktop view and five cards for the Mobile view. Therefore, you will create 10 cards in total.
2. Create the background image in Photoshop/GIMP for the game area.
3. Create “divs” for the different areas of the game: header, footer, playerID, playerScore, playerMove, endArea, gameArea.
4. Design the CSS for each of the components, the page layouts and use external fonts.
5. Once you are satisfied with the desktop view, make changes to the CSS file to incorporate the differences for the mobile view (assume sizes under 480px are mobile).

***Programming Tips***

1. Use double class attribute values to make styling easier. Here’s an example:

HTML:

<h1 class=“fruit apple”> Apple - Double class selector </h1>  
<h1 class=“fruit orange”> Orange - Double class selector </h1>

CSS:

.fruit {border-radius: 5px; border-width:5px;}

.apple {color: red};

.orange {color: orange};

1. The two pseudo-classes that you should include are :hover and :active.
2. There are multiple ways of working on the playArea. One of the ways is to consider using display: flex together with flex-wrap: wrap.

***Marking guide***

|  |  |  |
| --- | --- | --- |
| Area | Max | Grade |
| Desktop Site design   * Layout of game / dice / forms * Make use of larger screen with more details (such as bigger fonts, bigger images of the dice, detailed footer, etc) | 15 |  |
| Mobile site Design   * Layout of game / dice / forms * Smaller screen concerns addressed | 15 |  |
| CSS   * A new font using @font-face * Media breakpoints used * Background images loaded through styles * Use the cascade properly, limit duplication of attributes | 20 |  |
| Graphics   * Background image for the play area * Appropriate detail for large and small images. Remember to submit the photoshop/GIMP files * Bandwidth lower for mobile site * Graphics must be original / high quality / correct formats | 30 |  |
| Forms   * Well-designed visually (on both mobile and desktop) * Proper fields, method and action | 5 |  |
| Gameplay Styling   * Ten cards are displayed on play surface * When hovering over cards, the look changes * When the cards are clicked, the look changes | 15 |  |
| TOTAL | 100 |  |