PROJECT REPORT
CSE4500 Platform Computing
Instructor: Lawrence Orijuela

Name:_Nathan Bush_________________________ SCORE: ___/30

Student ID: _007463099_____________________ DUE: _2/28/23_

LAB:_Project 2 – Physics Project App2___________

Report

1. There are three total ways to add JavaScript to your HTML code. Describe all three with 2 or more sentences each.
   1.1. **Embedding the JavaScript directly in the HTML document.** The <script></script> tag is a way to code the JavaScript directly into the HTML document, by setting the opening tag to type='text/javascript'. Raw JavaScript can then be inserted within that block of code to manipulate the HTML document directly via the DOM.
   1.2. **Inserting the JavaScript into the HTML document by linking an external file.** Again, the <script></script> tags are used, but this time there is no content between the tags. Instead, in the header where the type="text/javascript" is declared, another attribute "src" is declared with a value for the relative path of the JavaScript file to include in the document.
   1.3. **Inserting an external file containing a JavaScript function, and calling that function within the HTML document.** As in the second option, the <script> tag is used to reference an external JavaScript file. However, rather than just containing a block of JavaScript, the code is written as a function. The function can then be called within the <body> tag using the "onload" attribute and passing it the name of the function along with any parameters it requires.

2. Explain how you add multiple pages to an HTML web app.

   Other pages can be added by simply creating new .html files to define those pages. The other .html files can be placed in the same directory as index.html, and there are multiple ways to route between web pages. Since we are only building a simple app, we will be using the jQuery Mobile navbar widget to link between pages. Within the navbar widget, we define an unordered list containing references to the various HTML pages we will write.

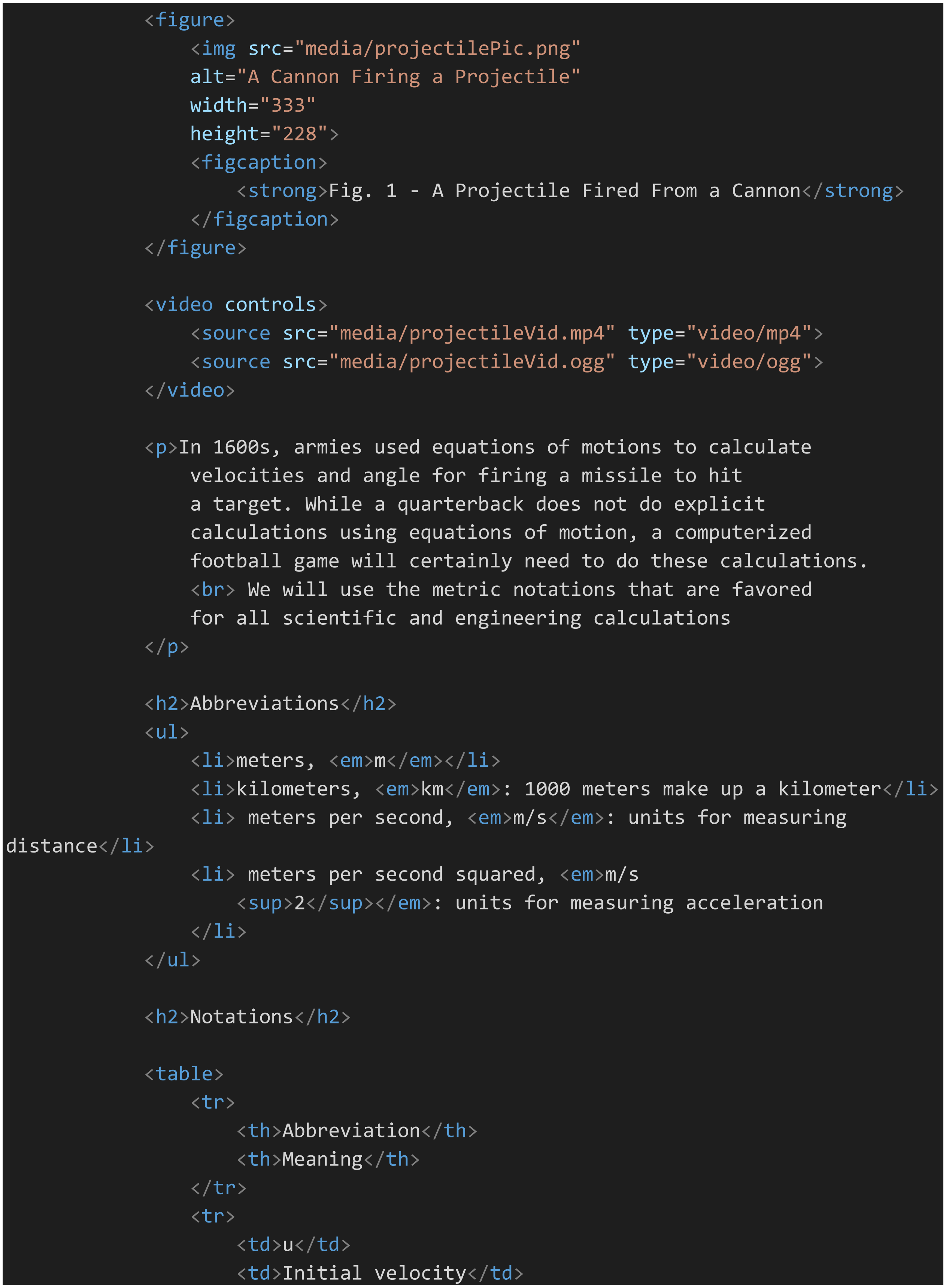
3. Explain how you add a widget (like a button) to an HTML web app.

   jQuery makes it easy by providing us with extensive files that have widgets implemented already. When using jQuery Mobile, we can use the API to use preset tags. For example, the <button></button> tag with text in between adds and names a button. There are many other widgets that can be used, for example the <label></label> tag can be used to label a text-input box by calling out the name of the text box to label with the "for" attribute. The text box itself uses an <input></input> tag along with a variety of attributes to describe behavior of the text box.

Source Code

```
<!-- CSE-4500 Platform Computing SPR23
     Nathan Bush - 007463099
     Project 2 - index.html -->

<!DOCTYPE html>
<html>

<head>
    <title>Physics Simulator App</title>
    <link rel="stylesheet" href="css/jquery.mobile-1.3.1.min.css">
    <script type="text/javascript"
src="scripts/physicsProjectileApp1.js"></script>
    <script src="scripts/jquery-1.8.3.min.js"></script>
    <script src="scripts/chromeFileProtocolFix.js"></script>
    <script src="scripts/jquery.mobile-1.3.1.min.js"></script>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>

<body>
    <div data-role="page">
        <div data-role="navbar">
            <ul>
                <li><a href="information.html">Information</a></li>
                <li><a href="index.html" class="ui-btn-active ui-state-
persist">Interface</a></li>
            </ul>
        </div>
        <div data-role="header">Projectile Calculator</div>
        <div data-role="content">
            <label for="angle">Angle:</label>
            <input type="range" name="angle" id="angle" min="0" max="90">
            <label for="velocity">Velocity:</label>
            <input type="range" name="velocity" id="velocity" min="0" max="999">
            <br>
            <button onclick="display();">Display</button>
            <br>
            <table id="data">
                <tr>
                    <td>Angle Entered:</td>
                    <td id="angleDisplay">0</td>
                    <td>degress</td>
                </tr>
                <tr>
                    <td>Velocity Entered:</td>
                    <td id="velocityDisplay">0</td>
                    <td>meters/second</td>
                </tr>
            </table>
        </div>
        <div data-role="footer">
            <small>
                Cross-platform mobile app demo
            </small>
        </div>
    </div>
</body>

</html>
```

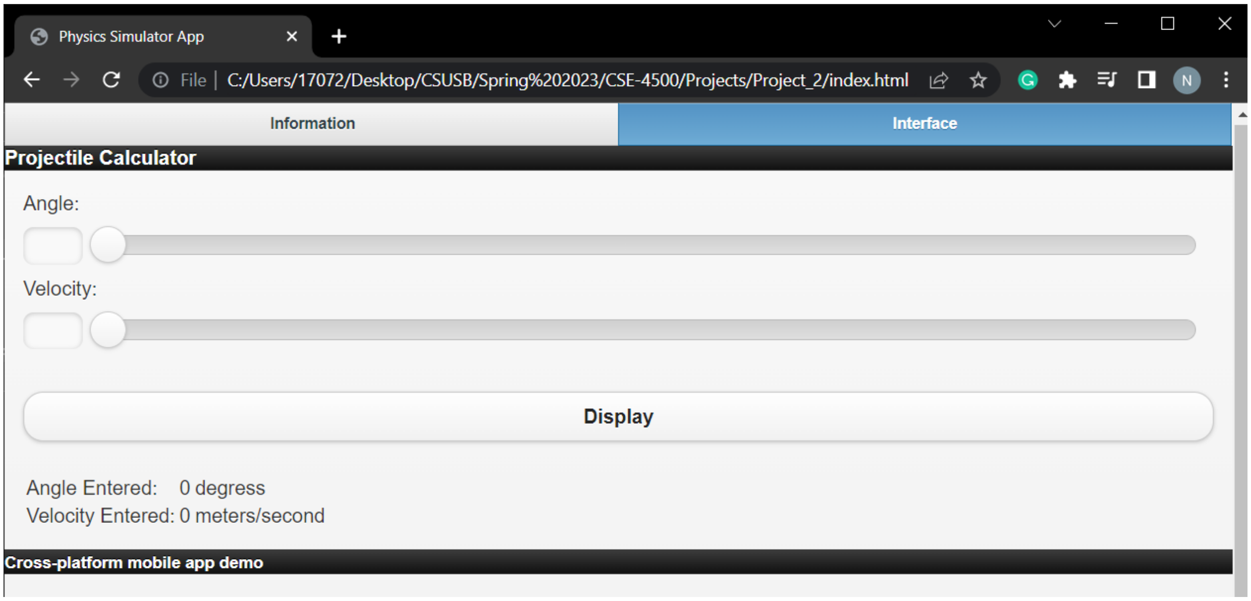
```
<!-- CSE-4500 Platform Computing SPR23
     Nathan Bush - 007463099
     Project 2 - information.html -->

<!DOCTYPE html>
<html>

<head>
    <title>Physics Simulator App</title>
    <link rel="stylesheet" href="css/jquery.mobile-1.3.1.min.css">
    <script type='text/javascript'
src='scripts/physicsProjectileApp1.js'></script>
    <script src="scripts/jquery-1.8.3.min.js"></script>
    <script src="scripts/chromeFileProtocolFix.js"></script>
    <script src="scripts/jquery.mobile-1.3.1.min.js"></script>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>

<body>
    <div data-role="page">
        <div data-role="navbar">
            <ul>
                <li><a href="information.html" class="ui-btn-active ui-state-
persist">Information</a></li>
                <li><a href="index.html">Interface</a></li>
            </ul>
        </div>

        <h1>Projectiles</h1>
        <div data-role="content">
```



```
            <figure>
                <img src="media/projectilePic.png"
                alt="A Cannon Firing a Projectile"
                width="333"
                height="228">
                <figcaption>
                    <strong>Fig. 1 - A Projectile Fired From a Cannon</strong>
                </figcaption>
            </figure>

            <video controls>
                <source src="media/projectileVid.mp4" type="video/mp4">
                <source src="media/projectileVid.ogg" type="video/ogg">
            </video>

            <p>In 1600s, armies used equations of motions to calculate
                velocities and angle for firing a missile to hit
                a target. While a quarterback does not do explicit
                calculations using equations of motion, a computerized
                football game will certainly need to do these calculations.
                <br> We will use the metric notations that are favored
                for all scientific and engineering calculations
            </p>

            <h2>Abbreviations</h2>
            <ul>
                <li>meters, <em>m</em></li>
                <li>kilometers, <em>km</em>: 1000 meters make up a kilometer</li>
                <li> meters per second, <em>m/s</em>: units for measuring
distance</li>
                <li> meters per second squared, <em>m/s
                    <sup>2</sup></em>: units for measuring acceleration
                </li>
            </ul>

            <h2>Notations</h2>

            <table>
                <tr>
                    <th>Abbreviation</th>
                    <th>Meaning</th>
                </tr>
                <tr>
                    <td>u</td>
                    <td>Initial velocity</td>
                </tr>
                <tr>
                    <td>v</td>
                    <td>Final velocity</td>
                </tr>
                <tr>
                    <td>a</td>
                    <td>Acceleration</td>
                </tr>
                <tr>
                    <td>t</td>
                    <td>Time</td>
                </tr>
                <tr>
                    <td>s</td>
                    <td>Distance</td>
                </tr>
            </table>

            <h2>Equations</h2>
            <ol>
                <li><mark>a = (v - u) / t</mark>, which can be arranged to get
the following
                    equation</li>
                <li><mark>t = (v - u) / a</mark>, which can be further arranged
as</li>
                <li><mark>v = u + a * t</mark></li>
                <li><mark>s = u * t + 0.5 * a * t<sup>2</sup></mark>, another
useful
                    equation of motion</li>
            </ol>
        </div>
        <div data-role="footer">
            <small>
                Cross-platform mobile app demo
            </small>
        </div>
    </div>
</body>

</html>
```

Screenshots

Browser render: index.html


Physics Simulator App
File | C:/Users/17072/Desktop/CSUSB/Spring%202023/CSE-4500/Projects/Project_2/index.html
Information
Interface
Display

Projectile Calculator

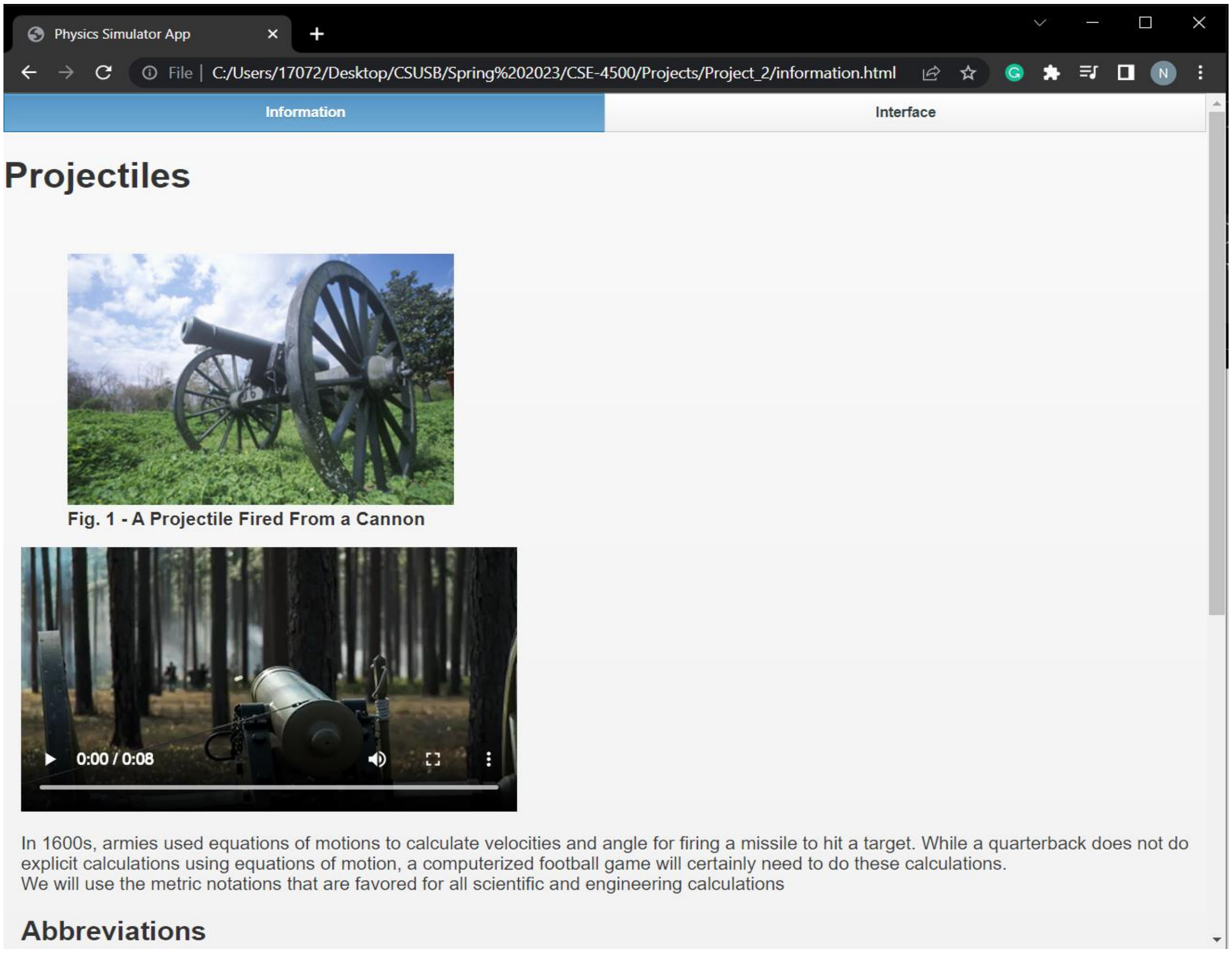
Angle:

Velocity:

Angle Entered: 0 degress
Velocity Entered: 0 meters/second

Cross-platform mobile app demo

Broswer Render: information.html


Physics Simulator App
File | C:/Users/17072/Desktop/CSUSB/Spring%202023/CSE-4500/Projects/Project_2/information.html
Information
Interface
0:00 / 0:08

Projectiles

Fig. 1 - A Projectile Fired From a Cannon

In 1600s, armies used equations of motions to calculate velocities and angle for firing a missile to hit a target. While a quarterback does not do explicit calculations using equations of motion, a computerized football game will certainly need to do these calculations.
We will use the metric notations that are favored for all scientific and engineering calculations

Abbreviations