CSC330 Homework 2 Fall 2016

Subject: Homework 2
Date assigned: Friday, October 14, 2016
Date due: At start of class on Wednesday, October 26, 2016

**General**: This assignment gives you a chance to apply the skills learned in chapters 1 - 6 to create a responsive **Future Value Calculator Application**. This application runs on a **OpenShift** Tomcat 7 web server and calculates the total amount accrued for the input parameters entered by the user for an investment amount (e.g. $1,000), a yearly interest rate (e.g. 5%) and the number of integer years to get compounded interest (e.g. 10 years). The application must use the latest html 5 form and input attributes to insure only valid input are provided. The application must support responsive web design for iPhones 5 and later.

**Objectives:** The objectives of this homework are:

1. To demonstrate your mastery of html and css by giving you the design of a web page desired by the customer and have you implement it using **validated HTML 5** and **validated CSS3**.

2. To demonstrate your mastery of **Model View Controller** by implementing control in servlets, business model logic in Java classes and views in Html 5 and JavaServer pages.

3. To demonstrate your ability to provide a **custom error page** if a wrong URL is entered by the user.

4. To demonstrate your ability to use **includes** to have a common header and common footer on each view.

5. To demonstrate your mastery of **responsive web design** to dynamically modify display based on width of device.

**Collaboration Requirements**: This is a pairwise assignment. You are only allowed to discuss the coding of the assignment with your assigned partner and Dave Powell. I will treat any violations of this requirement as an honor code violation. **You must place a copyright statement** as a comment near the top of each html, css, jsp and java file that indicates the work is completely and originally your own.

**Requirements:** The specific requirements are listed below. If you have any questions on the requirements then please see me for interpretation.

1. Create a Java Web Application Project with a project name of **homework2** with a context path of **/homework2**. You should use an OpenShift Tomcat 7 application for your application deployment.

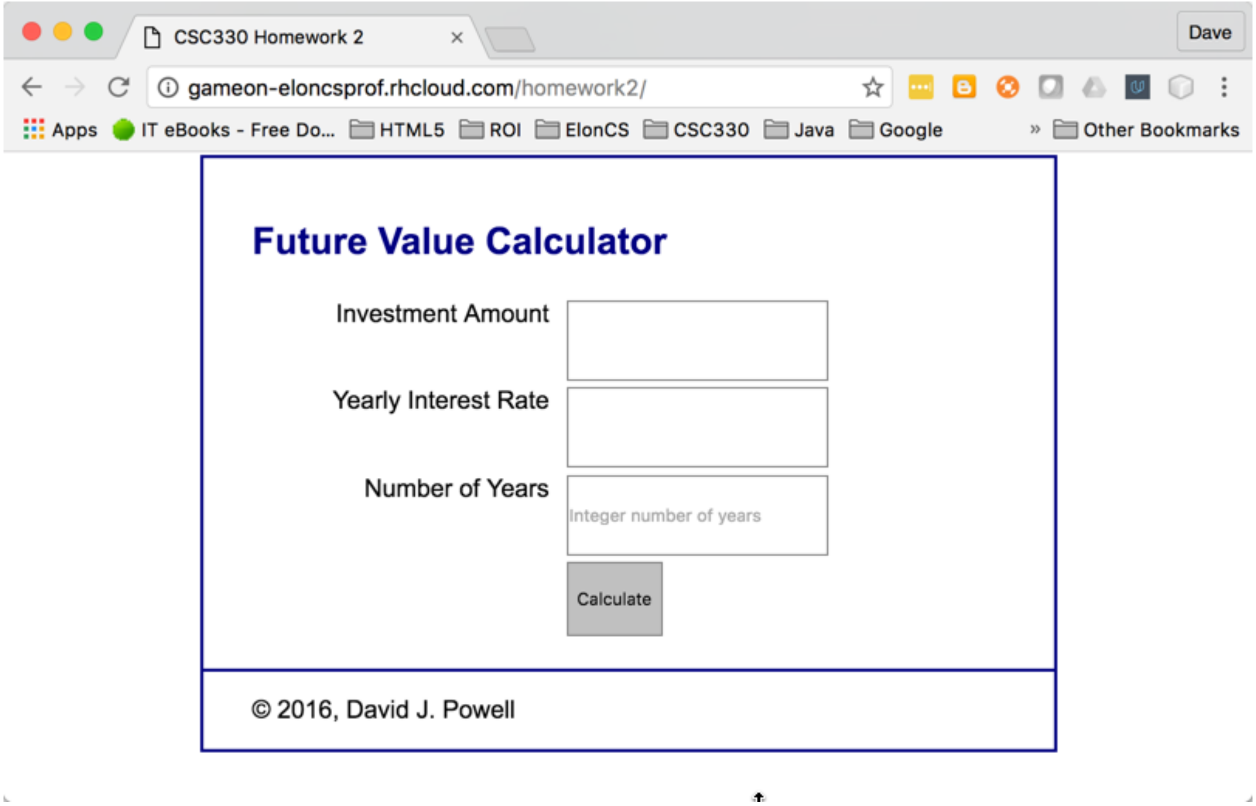
2. Your html, css and jsp files must follow the ***Google HTML/CSS Style*** guide. In addition to the style guide, you must place a Copyright with your name and your team mates name on each html, css and jsp file indicating that the work was originally and completely done by you and your team mate.

3. All css styles should come only from external files. The files should be in a **styles** folder.

4. All html and jsp files that display a view should include the same header and footer files.

5. Your servlet code and model code must follow the ***Google Java Style Guide*** coding conventions. You must use a **Model 2 MVC** architecture with all control coming from java servlet controllers, all model calculations coming from java classes and all views coming from html and jsp files.

6. Your application should be started by either typing in the OpenShift Application URL followed by the Context Path or the application URL followed by the ContextPath followed by the name of the controller servlet of **/calculate**. You need to support both ways. My application is called gameon-eloncsprof.rhcloud.com and my ContextPath is homework2. When invoked as **http://gameon-eloncsprof.rhcloud.com/homework2/** or as **http://gameon-eloncsprof.rhcloud.com/homework2/calculate** then a Jsp or html page should be displayed similar to that shown in Figure 1 if the device width is greater than 400 pixels. If the device width is less than 400px then you need to support responsive web design to smoothly accommodate display width changes down to and including iPhone 5 and iPhone 6 with a width of iPhone 5 in portrait mode of 320px. A display for an iPhone 5 is shown in Figure 2.



Future Value Calculator

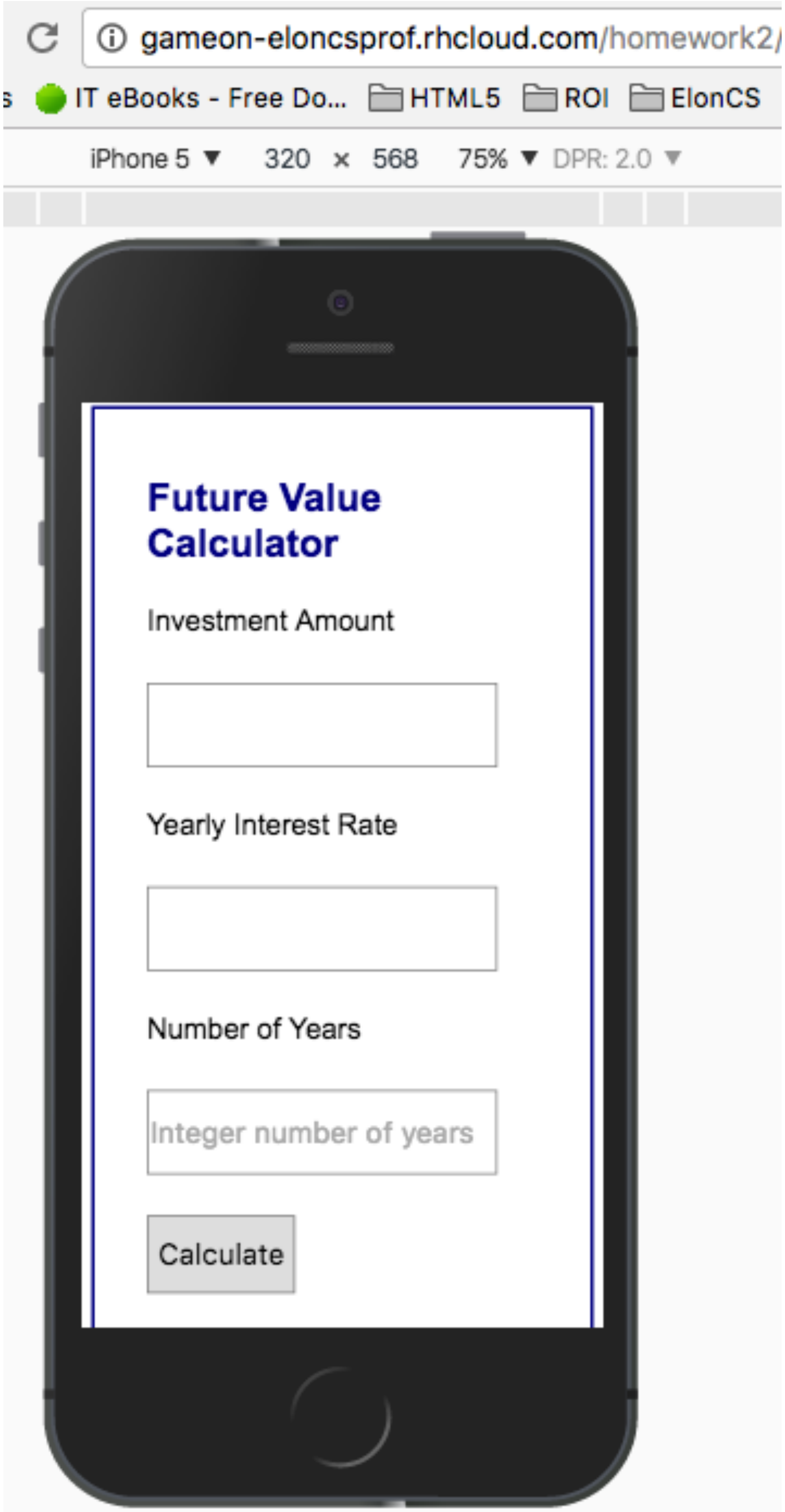
Investment Amount

Yearly Interest Rate

Number of Years

© 2016, David J. Powell

Figure 1: Initial Display of View for input of values for devices with minimum width greater than 400px

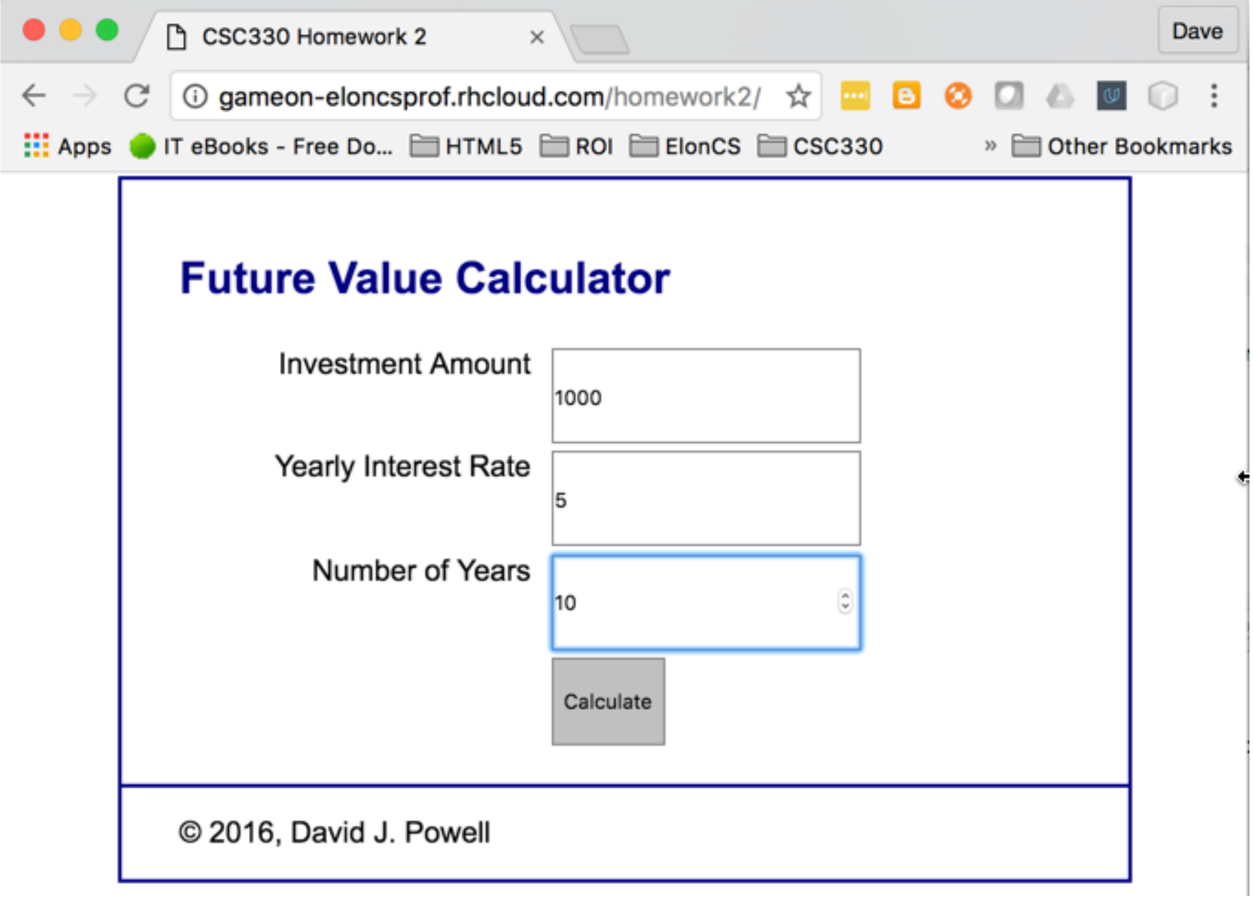


gameon-eloncsprof.rhcloud.com/homework2/

iPhone 5 ▼ 320 × 568 75% ▼ DPR: 2.0 ▼

Figure 2: Initial Display of View for iPhone 5 (widths between 320 and 400 should be similar)

7. Figure 3 shows user inputs of $1,000 dollars, 5% interest rate and 10 years to collect compounded interest on a device with a width greater than 400px. Figure 4 shows the same data entered on a device (iPhone 5) with a width less than or equal to 400px. The user clicks Calculate for the data to be sent to the controller from the page using **post** and not **get** methods. The html 5 form and input attributes should insure that only valid, non-negative numbers are entered for each of the three inputs. Until valid entries are made then the page should not be sent to the controller servlet. If valid values are entered and **Calculate** is clicked then a view similar to that shown in Figure 5 should appear.



Future Value Calculator

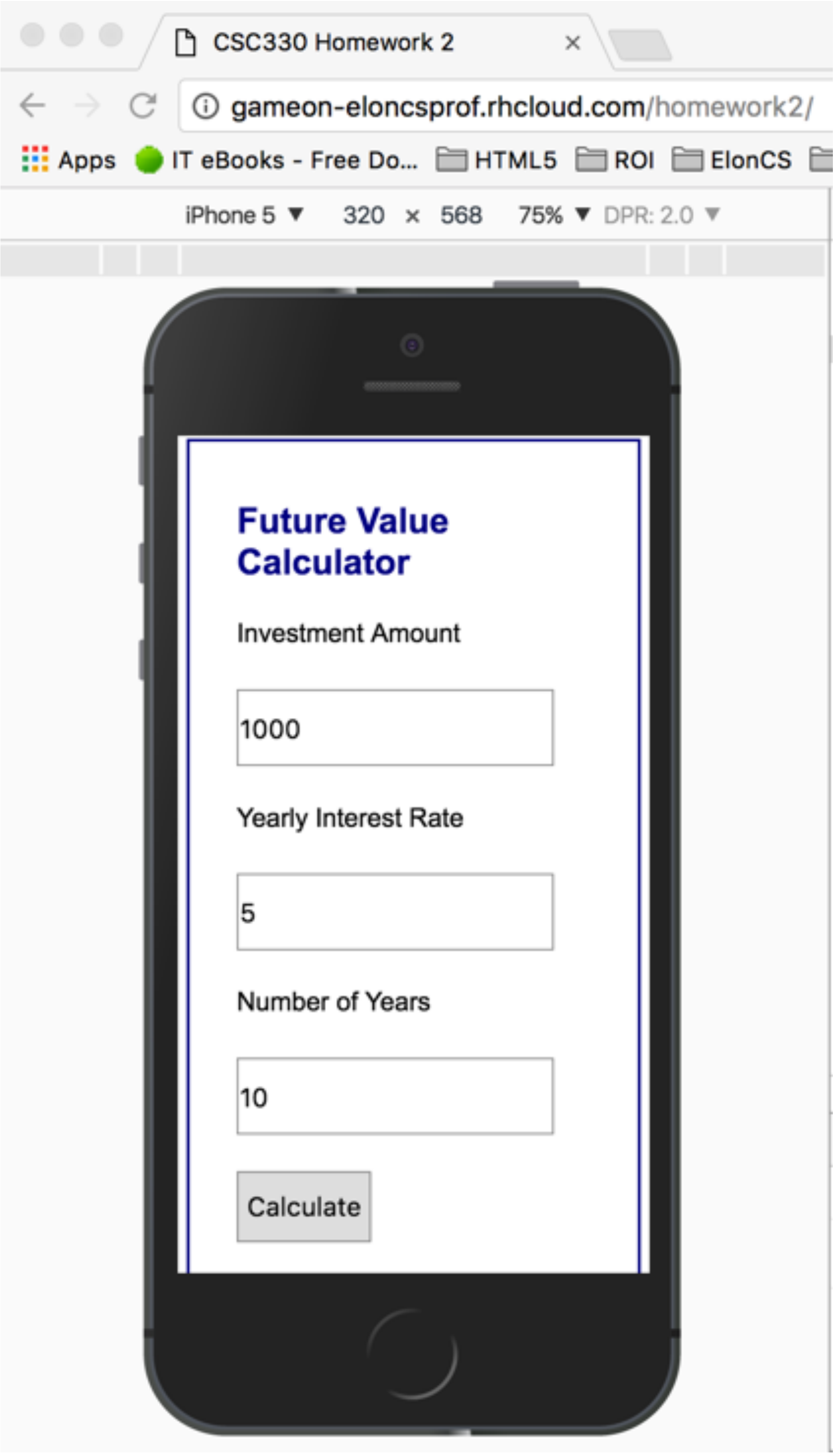
Investment Amount 1000

Yearly Interest Rate 5

Number of Years 10

© 2016, David J. Powell

Figure 3: User has entered valid numeric data for each input field and is about to click Calculate on a device larger than 400 px.



Future Value Calculator

Investment Amount

1000

Yearly Interest Rate

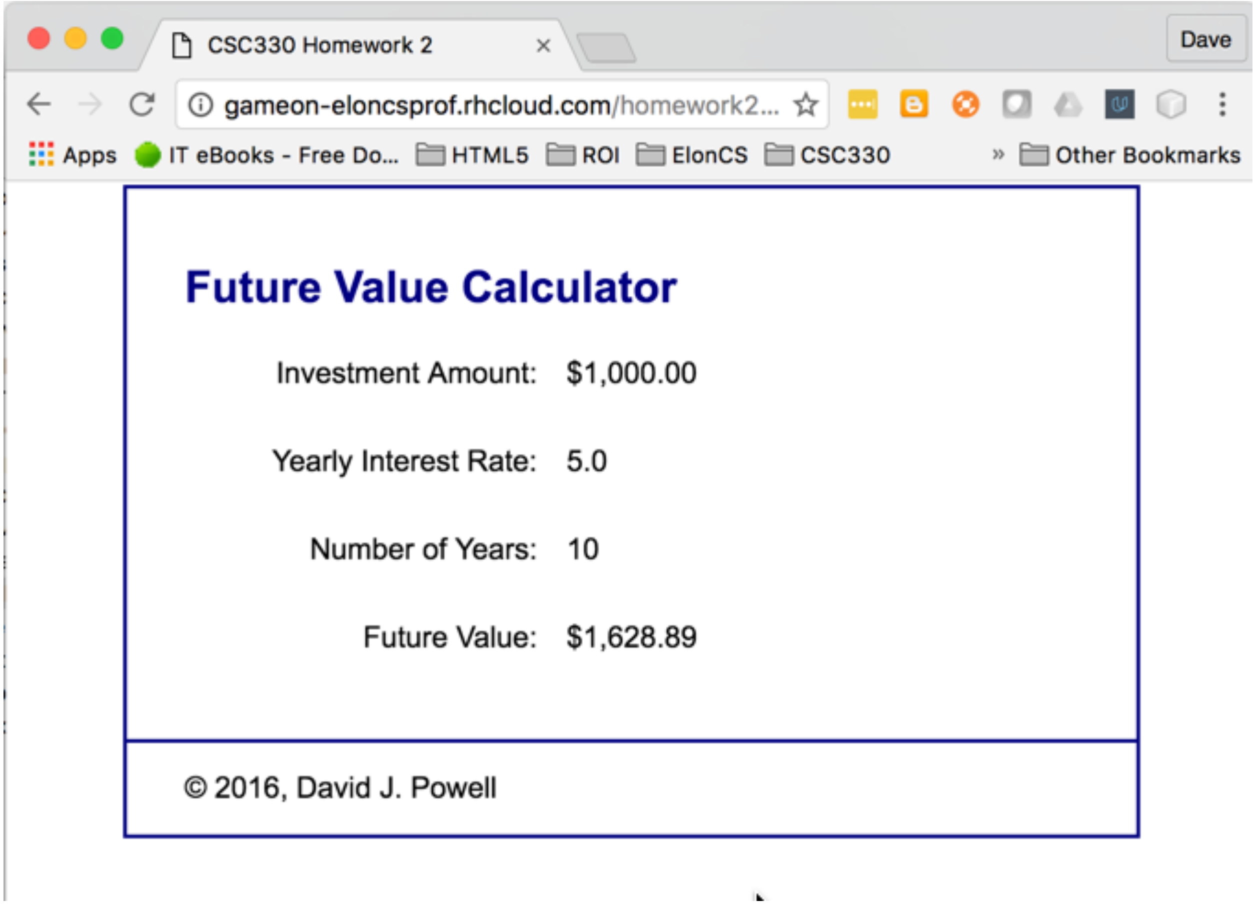
5

Number of Years

10

Figure 4: User has entered valid numeric data for each input field on an iPhone 5 and is about to click Calculate

8. Figure 5 shows the resulting view from clicking Calculate in Figure 3. The view has the same header and footer as Figure 3. The view should use currency formatting to provide dollar signs, commas if needed and 2 decimal place accuracy for the investment amount and future value. Figure 6 shows the resulting view on an iPhone 6. Note for the result views the formats are identical.



## Future Value Calculator

Investment Amount: $1,000.00

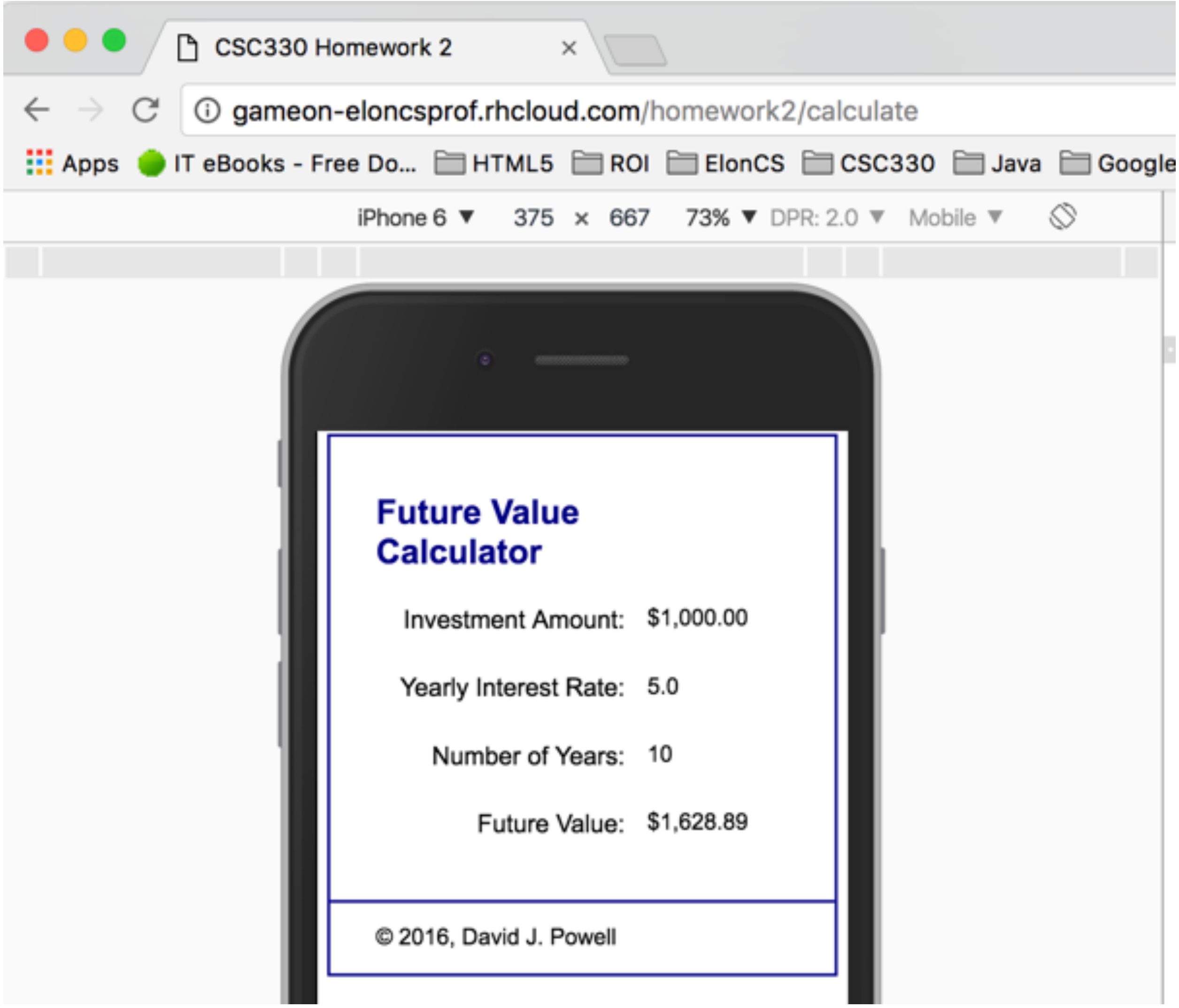
Yearly Interest Rate: 5.0

Number of Years: 10

Future Value: $1,628.89

© 2016, David J. Powell

Figure 5: View displays results of calculating future value



Future Value Calculator

Investment Amount: $1,000.00

Yearly Interest Rate: 5.0

Number of Years: 10

Future Value: $1,628.89

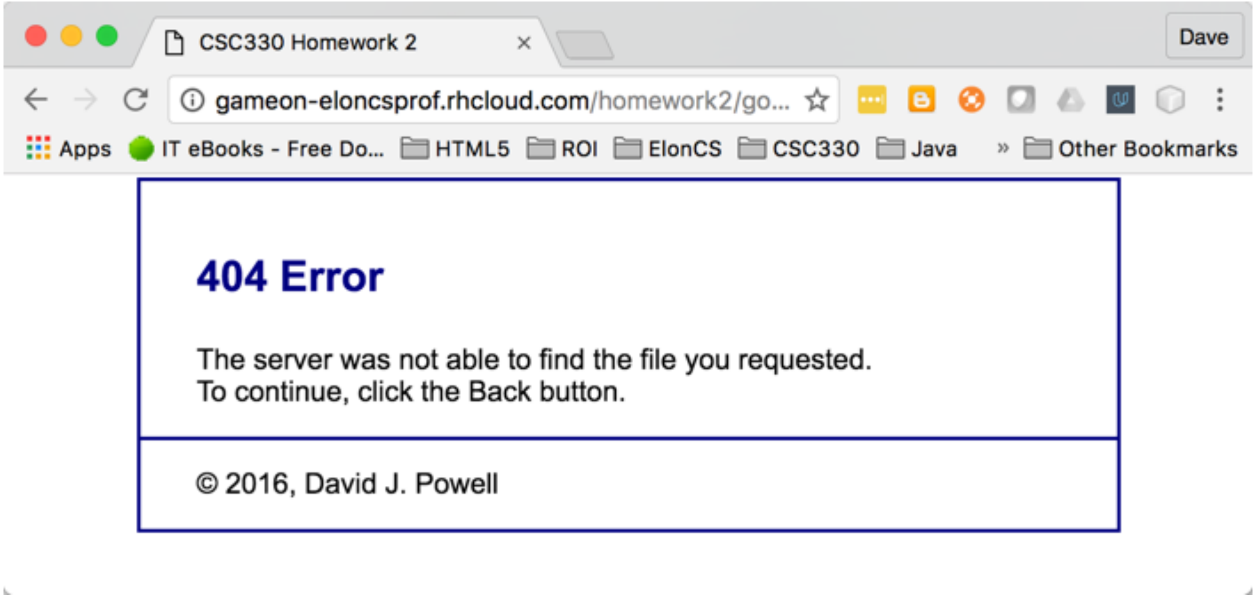
© 2016, David J. Powell

Figure 6: iPhone 6 view displays results of calculating future value. View is similar to that of larger devices.

9. Your servlet should be the controller for the application and all communication goes through the servlet. The servlet should set the data needed for the JSP shown in Figures 5 and 6 to display.

10. All jsp files that need to get data from the request object set by the servlet should use the old style **jsp:useBean** and **jsp:getProperty** tags discussed on pages 184-190.

11. If the user enters an invalid URL for your application then you should display a custom error page. For example, if the user entered **http://gameon-eloncsprof.rhcloud.com/djphomework2/goelon** then a custom error page similar that shown in Figure 7 should be displayed. Notice how the page has the same header and footer as Figures 1 – 6.



404 Error

The server was not able to find the file you requested.
To continue, click the Back button.

© 2016, David J. Powell

Figure 7: Custom error page for giving application an invalid URL

12. Each html and jsp file must use a DOCTYPE of html and validate for Html 5 using the validator at http://validator.w3.org/

13. Each css file must validate for css 3 using the validator at http://jigsaw.w3.org/css-validator.

14. There is a soft copy and hard copy submission requirement.

    a. For the softcopy, you must submit one NetBeans exported project file from your team to Moodle. Export your Project as a zip file called **homework2.zip** and submit it to moodle as an assignment file upload before the start of class on the due date.

    b. For the hard copy that you submit, hand in at the start of class your html files, your css files, your jsp files, your java files and your web.xml. Insure these files are properly formatted so they are easily readable with no wrapping. **At the top of the hardcopy, hand write or underline the typed complete URL on OpenShift that I need to invoke your application. Please double check your link for legibility and readability.**