In your **GitHub Portfolio Project: Front End Banking Application**you’ll refactor the Bad Bank exercise and add additional functionality. You’ll deploy your app to Amazon Web Services (AWS) so it can be viewed by others. Finally, you’ll create a video walkthrough of your application that highlights key features and includes a discussion of the challenges you encountered during this project.

You will also have the opportunity to revisit this project as part of your capstone project at the end of the course. As one of your options for capstone, you can refactor the Banking Application project to build a back end.

To get started, walk through the exercises with Dr. Sanchez in Week 18: Bad Bank and review the requirements on the Banking Application Assignment Submission page.

As you refactor this project, you will:

* Remove the Balance page, incorporating the balance on to the Withdraw and Deposit Pages
* Add validation to form fields
* Deploy it to AWS or a cloud service provider of your choosing. You will learn how to do this in **Week 20: Testing.** Note that Learning Facilitators will only be able to support you in using an AWS S3 bucket and cannot guarantee support for other providers.

You have the option to:

* Refactor the files to use create-react-app instead of React standalone. Dr. Sanchez demonstrates the React standalone in the videos.
* Include or remove the **login page.** It is not required.
* Add additional features, as long as your submission meets the minimum requirements of layout, functionality, and styles.

Below are some additional resources to help guide you. These are resources that you would often use as a front end developer, working with a UX designer and Product Owner to design and develop a site.

* [**Banking Application Wireframes:**](https://student.emeritus.org/courses/2102/files/1627592/download?wrap=1)[**download**](https://student.emeritus.org/courses/2102/files/1627592/download?download_frd=1) Here you can view mockups of what each page on the site should look like. Note that wireframes are meant to be simple, unstyled layouts, and these are modeled after the industry standard. Please note, *your site should be styled with Bootstrap.*
* [**Banking Application User Story:**](https://student.emeritus.org/courses/2102/files/1627593/download?wrap=1)[**download**](https://student.emeritus.org/courses/2102/files/1627593/download?download_frd=1)A user story details functionality from the perspective of a user. You’ll see that this document has the same requirements as outlined in the assignment rubric, but they are expressed in a way that helps the developer understand why the functionality is necessary and how it will be used.
* **Banking Application Code Starter Files:**It is expected that you refactor your code from the Bad Bank exercises you completed along with Dr. Sanchez’s videos. As a reminder, you can download this [starter code](https://student.emeritus.org/courses/2102/files/1627591/download?wrap=1)[download](https://student.emeritus.org/courses/2102/files/1627591/download?download_frd=1). that went with the Create Account video at the end of the week.

Regardless of the method you've chosen to host your website, **please use the following naming convention as part of your URL:**

Firstname-LastnameBankingApplication

If you have elected to use Amazon S3, the name you choose for the Amazon S3 bucket will be part of your website's URL. So, when you create and name your Amazon S3 bucket, use the previously mentioned naming convention.

**Deployment**

**Deployment For React Standalone**

If you’re using the standalone version of the app (not create-react-app), you can still deploy your application to an S3 bucket. The same rules still apply, you have to upload all your application files to the bucket and make it publicly accessible.

You can make your S3 bucket publicly accessible by following these steps:

* Go to your S3 bucket and locate the “Permissions” tab
* Once on that tab, scroll down to the “Bucket Policy” section and click “edit”
* Paste the following JSON into the box, making sure you change FIRSTNAME\_LASTNAMEBANKINGAPP to your actual bucket name:

{  
  
    "Version": "2008-10-17",  
  
    "Statement": [  
  
        {  
  
            "Sid": "AllowPublicRead",  
  
            "Effect": "Allow",  
  
            "Principal": {  
  
                "AWS": "\*"  
  
            },  
  
            "Action": "s3:GetObject",  
  
            "Resource": "arn:aws:s3:::**FIRSTNAME\_LASTNAMEBANKINGAPP**/\*"  
  
        }  
  
    ]  
  
}

* Click save.

Your bucket should now be publicly accessible.

**Create-React-App**

In the Deploy Static Website On Amazon S3 in Week 20, Dr. Williams will show you how to deploy a create-react-app on S3. You can follow those instructions if you decide to refactor your banking application with create-react-app.

* You can learn how to configure an Amazon S3 bucket for static website hosting [here (Links to an external site.)](https://docs.aws.amazon.com/AmazonS3/latest/userguide/s3-userguide.pdf#EnableWebsiteHosting).

**Website Functionality**

Your website should include the following functionality. Please review the rubric below for specific information on how each element will be graded.

1. **Navigation Bar**

* Includes Create Account, Deposit, Withdraw, All Data, and Home pages
* Routing: Each navigation bar item routes the user to the relevant page. For example, by selecting Home the user should be directed to the Home page.
* Styled with Bootstrap
* Highlighting: Each navigation bar item is highlight when you are on that page. For example, Home is highlighted when you are on the home page.
* Hover effect: When your mouse hovers over a navigation bar item, you see a pop up with a description of that page.

**2. Home Page**

* Includes bank title, image, and a welcome message.
* Styled as a Bootstrap card.

**3. Create Account Page**

* Includes a Bootstrap card with a form that has:
  + Name input field
  + Email address input field
  + Password input field
  + Create account button

**4. Create Account Page Functionality**

* Create account page should include the following functionality:
  + **Success message:**Upon selecting the create account button the user should see a success message.
  + **Add Another Account Button**: Upon selecting the create account button, the user should see an add another account button.
  + **Cleared Create Account Form:** Upon selecting the create account button, t will open a cleared create account form.
  + **Name validation**: The user receives an alert if the name field is left blank.
  + **Email validation**: The user receives an alert if this field is blank
  + **Password validation**: The user receives an alert if the password is less than 8 characters long.
  + **Disable**submit button if nothing is inputted

**5. Deposit Page**

* Includes a Bootstrap card with a form that has:
  + Deposit input field
  + Deposit button
  + Balance information displays above deposit form on the card

**6. Deposit Page Functionality**

* Deposit page should include the following functionality:
  + **Updated Balance:**When a user deposits, the balance updates.
  + **Success Message:**When a user completes the deposit form, they receive a success message confirming their deposit was received.
  + **Not A Number Alert:**User receives an alert if they add something that is not a number.
  + **Negative Deposit Alert:**User receives an alert if they try to deposit a negative number.
  + **Disable**deposit button if nothing is input

**7. Withdraw Page**

* Includes a Bootstrap card with a form that has:
  + Withdraw input field
  + Withdraw button
  + Balance information displays above deposit form on the card

**8. Withdraw Page Functionality**

* Withdraw page should include the following functionality:
  + **Updated Balance:**When a user completes the withdrawal form, the number submitted is subtracted from the total balance.
  + **Success Message:**When a user completes the withdrawal form, they receive a success message confirming that their withdraw was processed.
  + **Account Overdraft Feature:** When a user withdraws a number higher than the account balance, the user receives an alert message on the withdraw page.
  + **Not A Number Alert:**User receives an alert if they add something that is not a number.
  + **Disable**deposit button if nothing is input

**9. All Data Page Functionality**

* All Data page includes the following functionality:
  + **Track User Submissions:** All user submissions appear on All Data page.
  + **All Data Displayed On Bootstrap Card:** All Data is styled and displayed on a Bootstrap card instead of JSON.

**Submission Instructions:**

* Upload the link to your front end banking application website
* The url should include your first name and last name
* Note that the site should be deployed on a cloud service, like AWS3

Rubric can be downloaded [here](https://student.emeritus.org/courses/2102/files/1627595/download?wrap=1)[download](https://student.emeritus.org/courses/2102/files/1627595/download?download_frd=1).

Rubric

**Front End Banking Application Assignment Submission**

| Front End Banking Application Assignment Submission | | |
| --- | --- | --- |
| **Criteria** | **Ratings** | **Pts** |
| This criterion is linked to a learning outcomeNavigation | |  |  |  | | --- | --- | --- | | **15 Pts**  **Excellent**  The navigation bar meets expectations and includes at least one creative styling, such as: Hover effect: When a user hovers their cursor over a navigation bar element, they see a few words describing that page Highlighting: The navigation bar highlights the element of the current page the user is on | **10 Pts**  **Meets Expectations**  1.The navigation bar includes the following pages: Create Account Deposit Withdraw All Data Home Page 2. When selected, each element on the navigation bar displays the correct page 3. Navigation bar is styled with Bootstrap | **0 Pts**  **No Marks**  There is no navigation bar The navigation bar contains default styling The navigation bar does not include all the required pages | | 15 pts |
| This criterion is linked to a learning outcomeHome Page | |  |  | | --- | --- | | **5 Pts**  **Meets Expectations**  The Home Page includes a Bootstrap card with the following: Bank title Image Welcome message | **0 Pts**  **Does Not Meet Expectations**  There is no Home Page.  The Home Page does not include the following minimum requirements: Bank title Image Welcome message Home Page styled as a Bootstrap card | | 5 pts |
| This criterion is linked to a learning outcomeCreate Account Page | |  |  | | --- | --- | | **5 Pts**  **Meets Expectations**  The Create Account page includes a Bootstrap card with a form that includes the following elements: Name input field Email address input field Password input field Create Account button | **0 Pts**  **Does Not Meet Expectations**  There is no Create Account page.  The Create Account page does not include the following minimum requirements:  Styled Bootstrap card with a form Form fields of name, email address, password Create Account button | | 5 pts |
| This criterion is linked to a learning outcomeCreate Account Page Functionality | |  |  |  | | --- | --- | --- | | **10 Pts**  **Excellent**  The Create Account page meets expectations and includes input validation. You will receive 1 point for each piece of input validation functionality implemented:  Name validation: The user receives an alert if the name field is left blank  Email validation: The user receives an alert if this field is blank  Password validation: The user receives an alert if the password is less than 8 characters long  Disabled Submit button: The Submit button should appear disabled in case of no input | **6 Pts**  **Meets Expectations**  The Create Account page includes the following functionality (you will receive 1 point for each piece of functionality implemented):  Success message: Upon selecting the Create Account button, the user should see a success message  Add Another Account button: Upon selecting the Create Account button, the user should see an Add Another Account button Cleared Create Account form: Upon selecting the Create Account button, the use will be able to open a Cleared Create Account form | **0 Pts**  **No Marks**  There is no Create Account page.  The Create Account page does not include at least one feature mentioned below:  Success message: Upon selecting the Create Account button, the user should see a success message  Add Another Account button: Upon selecting the Create Account button, the user should see an Add Another Account button  Cleared Create Account form: Upon selecting the Create Account button, the use will be able to open a Cleared Create Account form Submitted Create Account data: Submitted Create Account data appears on the All Data page. | | 10 pts |
| This criterion is linked to a learning outcomeDeposit Page | |  |  | | --- | --- | | **5 Pts**  **Meets Expectations**  The Deposit page contains a Bootstrap card form with the following: Deposit input field  Deposit button  Balance information displayed above the Deposit form on the card | **0 Pts**  **Does Not Meet Expectations**  There is no Deposit page.  The Deposit page does not meet the following minimum requirements:  Deposit input field on a Bootstrap card Deposit button on a Bootstrap card Balance information displayed above the Deposit form on a Bootstrap card | | 5 pts |
| This criterion is linked to a learning outcomeDeposit Page Functionality | |  |  |  | | --- | --- | --- | | **10 Pts**  **Excellent**  The Deposit page meets expectations and includes input validation You will receive 1 point for each input validation functionality implemented:  Not A Number alert: The user receives an alert if they add something that is not a number  Negative Deposit alert: The user receives an alert if they try to deposit a negative number Disabled Deposit button: The Deposit button should appear disabled in case of no input | **7 Pts**  **Meets Expectations**  The Deposit page includes the following functionality: (You will receive 1 point for each piece of functionality implemented): Updated balance: When a user deposits, their balance gets updated  Success message: When a user completes the Deposit form, they receive a success message confirming their deposit was received | **0 Pts**  **No Marks**  There is no Deposit page. The Deposit page does not include at least one functionality described below:  Updated balance: When a user deposits, their balance gets updated  Success message: When a user completes the Deposit form, they receive a success message confirming their deposit was received | | 10 pts |
| This criterion is linked to a learning outcomeWithdraw Page | |  |  | | --- | --- | | **5 Pts**  **Meets Expectations**  The Withdraw page includes a Bootstrap card with the following: Withdraw input field on a Bootstrap card Submit button on a Bootstrap card Balance information displayed above the Withdrawal form | **0 Pts**  **Does Not Meet Expectations**  There is no Withdraw page.  The Withdraw page does not meet the following minimum requirements:  Withdraw input field on a Bootstrap card Submit button on a Bootstrap card Balance information displayed above the Withdrawal form | | 5 pts |
| This criterion is linked to a learning outcomeWithdraw Page Functionality | |  |  |  | | --- | --- | --- | | **10 Pts**  **Excellent**  The Withdraw page meets expectations and includes input validation. You will receive 1 point for each input validation functionality implemented: Account overdraft feature: When a user withdraws a number higher than the account balance, the user receives an alert message on the Withdraw page Not a Number alert: User receives an alert if they add something that is not a number  Disabled Deposit button: The Deposit button should appear disabled in case of no input | **7 Pts**  **Meets Expectations**  The Withdraw page includes the following functionality. (you will receive 1 point for each piece of functionality implemented): Updated balance: When a user completes the Withdrawal form, the number submitted is subtracted from the total balance  Success message: When a user completes the Withdrawal form, they receive a success message confirming that their withdrawal was processed | **0 Pts**  **No Marks**  There is no Withdraw page.  The Withdraw page does not include at least one functionality described below: Updated balance: When a user completes the Withdrawal form, the number submitted is subtracted from the total balance  Success message: When a user completes the Withdrawal form, they receive a success message confirming that their withdrawal was processed | | 10 pts |
| This criterion is linked to a learning outcomeAll Data Page Functionality | |  |  |  | | --- | --- | --- | | **10 Pts**  **Excellent**  The All Data page meets expectations and includes the following feature improvement:  All data displayed on a Bootstrap card: All Data is styled and displayed on a Bootstrap card instead of JSON | **8 Pts**  **Meets Expectations**  The All Data page includes the following functionality: Track user submissions: All user submissions appear on the All Data page | **0 Pts**  **No Marks**  There is no All Data page.  No data appears on the All Data page. | | 10 pts |
| Total points: 75 | | |

[Previous](https://student.emeritus.org/courses/2102/modules/items/589834)[Next](https://student.emeritus.org/courses/2102/modules/items/589836)

After completing your front end banking application, record a screen capture video that highlights the functionality you’ve developed and describes what you’ve done. Your video should be between 3 - 5 minutes and should answer the following questions:

* Demonstrate the following functionality:
  + Create an account
  + Deposit money
  + Withdraw money
  + Display data on All Data Page
  + Demonstrates input validation where applicable
* Discuss the following questions:
  + What challenges did you encounter as you created this project? How did you overcome them?
  + Discuss your design decisions. What changes did you make from the starter code?
  + Did you create a form component? Why or why not?
  + What extra features or enhancements did you build?

Rubric can be downloaded [here](https://student.emeritus.org/courses/2102/files/1627596/download?wrap=1)[download](https://student.emeritus.org/courses/2102/files/1627596/download?download_frd=1).

* [File upload](https://student.emeritus.org/courses/2102/assignments/108777?module_item_id=589836#submit_online_upload_form)
* [Google Doc](https://student.emeritus.org/courses/2102/assignments/108777?module_item_id=589836#submit_google_doc_form)
* [Media](https://student.emeritus.org/courses/2102/assignments/108777?module_item_id=589836#submit_media_recording_form)

Top of Form

|  |  |
| --- | --- |
| Upload a file, or choose a file you've already uploaded. | |
| Upload fileUse Webcam   Add another file | |
|  | Click here to find a file you've already uploaded |
|  | |
| Cancel Submit assignment | |

Bottom of Form

Rubric

**Front End Banking Application Video Walkthrough Submission**

| Front End Banking Application Video Walkthrough Submission | | |
| --- | --- | --- |
| **Criteria** | **Ratings** | **Pts** |
| This criterion is linked to a learning outcomeDemonstrate Functionality | |  |  |  | | --- | --- | --- | | **5 Pts**  **Excellent**  Video demonstrates the functionality of all five of the following:  Create an account Deposit money Withdraw money Display data on the All Data page Demonstrate input validation where applicable | **3 Pts**  **Meet Expectations**  Video demonstrates the functionality of at least 3 of the following:  Create an account Deposit money Withdraw money Display data on the All Data page Demonstrate input validation where applicable | **0 Pts**  **Does Not Meet Expectations**  Video does not demonstrate the functionality of at least 3 of the following:  Create an account Deposit money Withdraw money Display data on the All Data page Demonstrate input validation where applicable | | 5 pts |
| This criterion is linked to a learning outcomeDiscussion Part 1 Points | |  |  |  | | --- | --- | --- | | **10 Pts**  **Excellent**  Video addresses both of the following discussion points: As you were creating this project, what challenges did you encounter? How did you overcome them? Discuss your design decisions. What changes did you make from the starter code | **5 Pts**  **Meets Expectations**  Video addresses one of the following discussion points: As you were creating this project, what challenges did you encounter? How did you overcome them? Discuss your design decisions. What changes did you make from the starter code? | **0 Pts**  **No Marks**  Video does not meet the minimum requirements of discussing at least one of the following discussion points: As you were creating this project, what challenges did you encounter? How did you overcome them? Discuss your design decisions. What changes did you make from the starter code? | | 10 pts |
| This criterion is linked to a learning outcomeDiscussion Part 2 | |  |  |  | | --- | --- | --- | | **10 Pts**  **Excellent**  Video addresses both of the following discussion points: Did you create a form component? Why or why not? What extra features or enhancements did you build? | **5 Pts**  **Meets Expectations**  Video addresses one of the following discussion points: Did you create a form component? Why or why not? What extra features or enhancements did you build? | **0 Pts**  **Does Not Meet Expectations**  Video does not meet the minimum requirements of discussing at least one of the following discussion points: Did you create a form component? Why or why not? What extra features or enhancements did you build? | | 10 pts |
| Total points: 25 | | |

[Previous](https://student.emeritus.org/courses/2102/modules/items/589835)[Next](https://student.emeritus.org/courses/2102/modules/items/589837)