Capstone Option 1: Full-Stack Banking Application Introduction And Instructions (5:00)

(https://student.emeritus.org/courses/3289/modules/items/1112541)

○ (https://student.emeritus.org/courses/3289/modules/items/1112542)

○ (https://student.emeritus.org/courses/3289/modules/items/1112543)

○ (https://student.emeritus.org/courses/3289/modules/items/1112594)

○ (https://student.emeritus.org/courses/3289/modules/items/1112596)

Capstone Option 1: Full-Stack Banking Application

Choosing A Capstone Project

You can choose between two options for your capstone project:

- Full-stack banking application, which you were introduced to in Week 27. This project is recommended if you have found the course challenging or just at the right level.
- Full-stack restaurant application, which you will learn about in Weeks 28 and 29. This project is recommended if you want an additional challenge, as it will involve new tools and frameworks.
- Both capstone options require you to submit a link to your working application and three, 5-minute video presentations discussing your project.

Full-Stack Banking Application Project Introduction

For the full-stack banking application capstone project, you will submit four deliverables:

- Three presentation videos discussing different aspects of your project
- A link to your full-stack banking application website

Deliverables	Total points
Presentation Video 1: Front-End Architecture, Authentication, And App Diagram	12
Presentation Video 2: Database And API	12

Presentation Video 3: Deployment, Additional Features, App Demonstration, And Reflection	16
Full-Stack Banking Application Link	60
Total	100 points

Application Requirements

You can refactor your front-end banking application GitHub portfolio project into the full-stack banking application. Follow along with Dr. Sanchez's videos in the Back End section of Week 27. While you are encouraged to refactor your front-end banking application project, you may also use the **starter files.** (https://mo-pcco.s3.us-east-1.amazonaws.com/mo-baen/week7/full-stack-banking-application_starter.zip)

Your refactor should include:

- Front end modifications
- A Node.js server
- A MongoDB database and data abstraction layer
- API integration
- Connection of the front end and back end
- Authentication
- The application deployed to a cloud service of your choice
- Development of additional features, which will be evaluated in your presentation

You are encouraged to develop additional features for your app. You may develop any of the features recommended below or develop new features that are not listed here:

- Roles for different users, such as a bank employee vs customer (authorization)
- Money transfer between users
- Account types, such as checking vs savings accounts
- Assignment of random account numbers to new accounts
- User profile updates that are persistent
- Check deposit by taking a picture of the check

Grading Your Application

Your application will be evaluated in a manner similar to user acceptance testing. The grading team will access your site as users and confirm that the following functionalities are working:

- Create an account with an email address and password
    - A success message is received upon account creation

    - The email address or username appears at the top right corner when logged in
- Log in to the account
    - The user can log in with an email address, password, or OAuth2 authentication
    - The user can log out of the account
- Deposit
    - The user can deposit money
    - The balance updates according to the deposit
    - Information persists across logins
- Withdraw
    - The user can withdraw money
    - The balance updates according to the withdrawal
    - Information persists across logins
- Database
    - You will submit a screenshot of user information in your database as additional support that your database is working

Presentation Requirements

You will complete the banking app presentation in three parts.

Presentation Video	Presentation length	Total points
Video 1: Front-End Architecture, Authentication, And App Diagram	5 minutes	12
Video 2: Database And API	5 minutes	12
Video 3: Deployment, Additional Features, App Demonstration, And Reflection		
Total	15 minutes	40 points

Presentation Video 1: Front-End Architecture, Authentication, and App Diagram

Here, you'll outline a high-level roadmap of the front end, middle tier, and back end of your application in a diagram. You'll also share the front-end architecture and the methods of authentication that you've developed.

Presentation Video 2: Database And API

In this video, you'll describe your database, data structure, and data abstraction layer (DAL). You'll also discuss your API endpoints, showcase API documentation, and share what API tools and protocol you used.

Presentation Video 3: Deployment, Additional Features, App Demonstration, And Reflection

In this final presentation, you'll discuss the deployment process, share any roadblocks you encountered, and how you overcame them. You'll also highlight any additional features you've developed, demo the working application, and reflect on what you might have done differently if you were starting the project today.

Grading Your Presentation

Please review the rubric and the assignment submission page for a detailed outline of how your submissions will be graded.

Each presentation should follow this format:

- Three to five minutes in duration.
- A link to your screencast. You can use the free accounts on tools such as **Zoom (https://www.zoom.us/)** or **Screencast-O-Matic (https://screencast-o-matic.com/screen-recorder)**.
- A slide template: You must use this **slide deck (https://student.emeritus.org/courses/3289/files/3139912/download?wrap=1)** to structure your presentation around the key points of the presentation. You are also welcome to include screenshots of code within the deck or navigate from the deck to show your application or code files. You must submit your slides along with your screencast links.
- The following naming convention: please name your video titles FirstName_LastName_PresentationVideo#

Learning Outcomes

Course Learning Outcomes Addressed

- Build a full-stack web application using the MERN Stack
- Explain the interactions between the different components of the MERN Stack
- Create a back-end API with Express
- Create and interact with a database
- Integrate React with a back-end Express API

Activities

Key Activities

- Full-Stack Banking Application link submission
- Full-Stack Banking Application presentation submission
- Share your Full-Stack Banking Application discussion

Support

Q&A Discussion Board

If you have questions about this week's assignments or content, you can post them on the Q&A discussion board, which will be monitored by your learning facilitators. In this forum, you can also interact with fellow classmates and help each other. If you do not get sufficient clarification here, please bring the question to your learning facilitators' office hours.

Need help? Contact Program Support

For assistance, select **Support** in the course menu. The Help Center has answers to the most common queries we receive. For further assistance, select **Submit a Reques**t at the top of the Help Center page and complete the form. You will receive a prompt response.