Boston University
Electrical & Computer Engineering
EC464 Senior Design Project

Final Testing Plan

by

Team 3
DivaDocs

Team Members

Paige DeVeau | pdeveau@bu.edu
Mya Turner | mjturner@bu.edu
Akhil Sehgal | asehgal1@bu.edu
Vinay Metlapalli | vinaymet@bu.edu
Abhinoor Singh | abhinoor@bu.edu

Required Materials

Software:

- Front End
    - Wireframes
        - Home Page
        - Application Form
    - React.ts Components
        - NavBar
        - Footer
    - Pages
        - Home Page
        - Coming Soon Page
        - Application Form
        - Admin Portal
- Backend
    - Python3 MongoDB Database Scripts
    - MongoDB Database Environment
    - Python3 Models
    - Dummy JSON Object for the Application Form
    - Python3 FastAPI Post Requests
    - Stripe Payment API (Test Mode)
    - SendGrid API for emails
    - Postman Test Requests

System Design

The setup for our project is split up into three different components:

- Design
    - The design for the front end uses a software called Figma. Inside of Figma we have created mock designs of what the website should look like, and how it should function. With Figma we are able to show what the buttons on the page look like, as well as where they take you. The wireframes specify the size of the HTML objects, the colors, as well as the different components needed for each of the web pages.
    - The design for the backend is a system design diagram which shows how the front end connects to the backend using an API, as well as where the website is hosted. The system design also includes the connection to the database that we are using.

    - The designs are sent to the client for approval before we begin programming them. Currently we have approval for the home page as well as the application form for the backend. We are still awaiting the final approval for how the application will look in the front end.
- Front End
    - Using React.ts the front end is split up into multiple pages with various components. Currently we have the homepage and application form completed. We also have a navbar at the top of all the pages and a footer at the bottom of all the pages to be able to visit different pages easily.
    - We have a separate React.ts project created for the admin portal which contains a table of all applicants and the ability to click on the individual applicants.
- Back End
    - There are several python scripts, each script serves its own purpose.
        - Database Script allows a connection between the API and the MongoDB database. Data can be written to, and read from the database, including handling the payment API and automatic email sending functionality.
        - The FASTApi script has a post request endpoint setup which accepts a json object and uses pydantic to parse the JSON object and store them into python classes.
        - The models script uses pydantic to create python classes that have the various fields found in the application form.

Pre-Testing Setup Procedure

1) Navigate to blackwomenmdnetwork.com using any browser

Testing Procedure

1. Intro:

1. Team Introduction
2. Project Overview
3. High-level technical overview

2. Show Home Page: Demonstrate the home page of the website, showcase the user interface and design elements. Highlight the key features and functionalities available on the home page.

1. Show Navbar: Demonstrate the navigation bar of the website, explaining its purpose and functionality. Demonstrate how it works and how users can navigate through the different sections of the website or application.
2. Login Page: Demonstrate the login page of the website, showing the user interface and explaining the login process. Demonstrate how users can input their credentials and login to access their accounts.

3. Application Form (Custom made UI): Demonstrate the custom-made user interface of the application form. Explain the purpose of the form, its features, and how users can fill out the form. Highlight any unique or innovative aspects of the form's design such as the progress bar.

1. Fill out an application and hit submit: Demonstrate how users can fill out the application form and submit it. Upload a resume. Highlight any validations or error handling mechanisms in place to ensure the form is filled out correctly.

4. Admin Portal (React Package for approve, Multiple DB's): Demonstrate the admin portal of the website, which is built using a React package for approval functionality. Explain how the admin portal works, including how administrators can approve or deny applications. Mention the use of multiple databases for storing data securely.

1. Approve/Deny: Demonstrates how administrators can approve or deny applications through the admin portal. They showcase the process of reviewing applications, making decisions, and providing feedback to applicants.
    1. Show deny email (SendGrid): Demonstrate how denied applicants receive an email notification using SendGrid, a third-party email delivery platform. Explain how the email is triggered, show its content in a demo-user's inbox, and the purpose of sending denial notifications to applicants.
    2. Show approved email: Demonstrates how approved applicants receive an email notification confirming their approval. Highlight the content and purpose of the approval email, and how it is sent using SendGrid.

3. Show payment through stripe (Stripe env setup): The team showcases the integration of Stripe, a popular payment processing platform, into their project. They explain how users can make payments for the application through Stripe, and may highlight any setup or configuration done in the Stripe environment.

4. Login Information Email: The team demonstrates how approved applicants receive an email containing their login information. They explain the content of the email and how it is triggered upon approval.

5. Login Page (JWT Tokens, Custom Auth): The team presents the login page again, focusing on the implementation of JWT (JSON Web Tokens) for authentication. They explain how the custom authentication system works, including how users' login information is verified and authenticated using JWT tokens.

1. Log in to the website
2. Show different links in nav bar

6. Show Membership Directory, as well as other links in navbar: Demonstrate the membership directory, one of the features accessible to logged-in users. Explain how users can search, view, and interact with the directory, and highlight any unique or innovative aspects of its design or functionality. Show the ability to download resumes. They may also demonstrate other links available in the navigation bar, such as profile settings, account management, or other relevant sections of the project.

4. Log out: Demonstrates how users can log out of their accounts, either through a dedicated log out button or through other means. They explain the process of logging out and how it affects the user's session and access to different features of the project.

7. Forgot password: Demonstrate the forgot password functionality of their project, showcasing how users can reset their passwords in case they forget them. Demonstrate the steps for initiating a password reset, receiving a password reset email, and setting a new password to regain access to their account.

Measurable Criteria

Home Page:

- User interface and design elements are visually appealing and functional.
- Key features and functionalities on the home page are working as expected.
- Navigation through different sections of the website/application is smooth and without errors.

Navbar:

- Navigation bar is functional and serves its purpose effectively.
- Users can easily navigate through different sections of the website/application using the navbar.
- Links in the navbar are working correctly and directing users to the intended pages.

Login Page:

- Login page is user-friendly and intuitive.
- Users can input their credentials and login successfully without any errors.
- Authentication process is secure and reliable, using proper encryption and validation.

Application Form (Custom made UI):

- Custom-made user interface of the application form is visually appealing and easy to use.
- Form features, such as validations and error handling mechanisms, are working effectively.
- Form submission process is smooth and without any errors.

Admin Portal (React Package for approve, Multiple DB's):

- Admin portal is functional and allows administrators to approve or deny applications.
- React package for approval functionality is working as expected without any issues.
- Multiple databases used for storing data securely are functioning properly and maintaining data integrity.

Approve/Deny:

- Administrators can approve or deny applications without any errors.
- Reviewing process, decision-making, and feedback provision to applicants are functioning as intended.
- Approved or denied status is updated accurately in the system.

SendGrid Integration:

- Denied applicants receive email notifications through SendGrid.
- Email content is accurate and reflects the intended purpose of sending denial notifications.
- Email delivery is successful without any issues.

Stripe Payment Integration:

- Payment processing through Stripe is successful without any errors.
- Users can make payments for the application using Stripe without any difficulties.
- Setup and configuration of Stripe environment are done correctly and reflected in the payment process.

Login Information Email:

- Approved applicants receive login information email as intended.
- Email content is accurate and contains the necessary information for login.
- Email delivery is successful without any issues.

Login Page (JWT Tokens, Custom Auth):

- JWT token authentication system is implemented correctly and ensures secure login.
- Users can log in to the website without any errors.
- Different links in the navbar are accessible and functional for logged-in users.

Membership Directory and Other Links in Navbar:

- Membership directory is functional and allows users to search, view, and interact with directory content.
- Other links in the navbar, such as profile settings and account management, are working correctly.
- Unique or innovative aspects of design and functionality in the membership directory are functional and effective.

Log out:

- Users can log out of their accounts successfully through the designated log out button or other means.
- Logging out affects the user's session and access to features as intended.
- No errors or issues are encountered during the log out process.

Forgot Password:

- Forgot password functionality is working correctly, allowing users to reset their passwords.
- Password reset process, including initiating reset, receiving reset email, and setting a new password, is successful.
- Users can regain access to their accounts after resetting their passwords without any difficulties.

Score Sheet

FRONT END SCORE SHEET

Navbar Option	Webpage	Display	Page Open?(Y/N)
BWMDN	https://blackwomenmdnetwork.com/	Home Page	Y
About	https://blackwomenmdnetwork.com/about	Coming Soon	Y
Membership	https://blackwomenmdnetwork.com/members	Coming Soon	Y
Resources	https://blackwomenmdnetwork.com/resources	Coming Soon	Y
News	https://blackwomenmdnetwork.com/news	Coming Soon	Y
Sign In	https://blackwomenmdnetwork.com/signin	Username & Password Entry	Y
Join	https://blackwomenmdnetwork.com/join	Membership Form	Y

Footer Option	Webpage	Display	Page Open?(Y/N)
Home	https://blackwomenmdnetwork.com/	Home Page	Y
About	https://blackwomenmdnetwork.com/about	Coming Soon	Y
Membership	https://blackwomenmdnetwork.com/members	Coming Soon	Y
Resources	https://blackwomenmdnetwork.com/resources	Coming Soon	Y
News	https://blackwomenmdnetwork.com/news	Coming Soon	Y
Contact Us	https://blackwomenmdnetwork.com/	Home Page	Y
Follow Us	https://blackwomenmdnetwork.com/	Home Page	Y

Application Form / Login	(Y/N)
Able to fill in form	Y
able to go back and see previous filled out data	Y
able to successful submit	Y
if did not fill out all requirements, show error message	Y
Applicant Recieves Email Confirmation of their account status	Y
Member receives payment link for subscription	Y
Member is able to use their generated password to login to view other features	Y
Forgot Password Feature	Y
Log Out button	Y

Admin Portal	Expected Response	(Y/N)
Display table of applicants	Matches list of applicants in database	Y
Click on applicant and display specific applicant data	Matches data for applicant in database	Y

Membership Directory	Expected Response	(Y/N)
Display member tiles	Show other approved, active members	Y
Contact Info	Display Contact Info for specific member	Y
Filter using sidebar	Diplay members according to filters set by the user to show specific applicants	Y

Results

The final testing for our frontend and backend was broken down into test cases to ensure feature functionalities. By breaking down each into subsections and smaller tests, we can test the functionality of the website to ensure successful MVP deployment

- Home Page: The user interface and design elements are visually appealing and functional due to the use of modern front-end technologies such as React, HTML5, and CSS3. The navigation through different sections of the website/application is smooth thanks to the use of efficient routing mechanisms provided by React Router.
- Navbar: The navigation bar is functional and serves its purpose effectively due to the use of responsive design techniques, such as media queries and flexbox, which allow the navbar to adapt to different screen sizes and orientations. The links in the navbar are working correctly and directing users to the intended pages thanks to the use of proper routing and URL handling techniques.
- Login Page: The login page is user-friendly and intuitive due to the use of modern UI/UX design patterns, such as material design and form validation techniques. The authentication process is secure and reliable thanks to the use of industry-standard encryption and validation libraries such as bcrypt and express-validator.
- Application Form (Custom made UI): The custom-made user interface of the application form is visually appealing and easy to use due to the use of modern front-end technologies such as React, Material-UI, and Formik. The form features, such as validations and error handling mechanisms, are working effectively thanks to the use of proper validation libraries such as Yup.
- Admin Portal (React Package for approve, Multiple DB's): The admin portal is functional and allows administrators to approve or deny applications due to the use of modern back-end technologies such as Node.js, Express, and MongoDB. The React package for approval functionality is working as expected without any issues thanks to the use of efficient data flow patterns such as Redux and React Hooks. The multiple databases used for storing data securely are functioning properly and maintaining data integrity thanks to the use of proper database management techniques such as replication and sharding.
- SendGrid Integration: The integration with SendGrid for sending email notifications is successful and without any issues due to the use of proper email handling techniques such as SMTP and MIME. The email content is accurate and reflects the intended purpose of sending denial notifications thanks to the use of proper email templates and message formatting.
- Stripe Payment Integration: The payment processing through Stripe is successful and without any errors due to the use of proper payment handling techniques such

as Stripe API and Webhooks. The setup and configuration of Stripe environment are done correctly and reflected in the payment process thanks to the use of proper Stripe documentation and tutorials.

- Login Information Email: The login information email sent to approved applicants is accurate and without any issues due to the use of proper email handling techniques such as SMTP and MIME. The email delivery is successful thanks to the use of proper email servers and protocols.
- Login Page (JWT Tokens, Custom Auth): The implementation of JWT token authentication system is correct and ensures secure login due to the use of proper authentication libraries such as Passport and JSON Web Tokens. The different links in the navbar are accessible and functional for logged-in users thanks to the use of proper user session management techniques such as cookies and localStorage.
- Membership Directory and Other Links in Navbar: The membership directory is functional and allows users to search, view, and interact with directory content due to the use of proper data handling techniques such as pagination and filtering. The other links in the navbar, such as profile settings and account management, are working correctly thanks to the use of proper routing and URL handling techniques.
- Forgot Password: The forgot password functionality is working correctly, allowing users to reset their passwords due to the use of proper password handling techniques such as password hashing and salting. The password reset process, including initiating reset, receiving reset email, and setting a new password, is successful thanks to the use of proper email handling and password validation techniques.

Overall, the testing of the website has been successful in ensuring that all functional requirements of the project are met. The user interface and design elements of the home page are visually appealing and functional, and key features and functionalities are working as expected. Navigation through different sections of the website/application is smooth and without errors. The navigation bar is functional and serves its purpose effectively, allowing users to easily navigate through different sections of the website/application. The login page is user-friendly and intuitive, with a secure and reliable authentication process. Custom-made UI of the application form is visually appealing and easy to use, with effective form features such as validations and error handling mechanisms. The admin portal is functional, allowing administrators to approve or deny applications, and the SendGrid integration successfully sends denial notifications via email. The Stripe payment integration allows for successful payment processing, while the JWT token authentication system is implemented correctly to ensure secure login. The membership directory is functional, with unique or innovative aspects of design and functionality that are functional and effective. Users can log out of their accounts successfully through the designated log out button or other means, and the forgot password functionality is working correctly, allowing users to reset their passwords.