

WEB701 Assessment Two
Milestone Three

Xander Crocker

Table of Contents

1. Part One: Implementation Critique.....	3
1.1 Requirement One Critique:	3
1.2 Requirement Two Critique:	4
1.3 Requirement Three Critique:	5
1.4 Requirement Four Critique:	5
1.5 Requirement Five Critique:.....	6
2. Part Two: Researching and Implementing Emerging Technology	7
3. References.....	9

1. Part One: Implementation Critique

1.1 Requirement One Critique:

Charity members and beneficiaries can register, login and administer their own accounts.

Description of Implementation:

When users reach the home page of the Gaming4Good website, they will be able to navigate to the registration page. On the registration page, a user can register an account. When they do the user will have three options to choose from that will determine what role/permissions they will have as a user. These options include user, charity member, and beneficiary. A user with just user permissions is able to book events that are listed on the events page. A charity member has the same functionality as a user except they are able to create events that will be seen by all users on the events page. Lastly, a beneficiary user has the same functionality as both other roles, but they will receive a 50% discount when booking an event.

Once a user has registered their account, they can navigate to the login page to log in using their registered credentials. If the user logs in with the correct details, they will have access to two new pages that will show up in the navigation bar to the top right of the website. These will be the dashboard page and the profile page. The dashboard is where a user will be able to view any booked events they have purchased tickets to. The profile page will allow users to edit their details such as username, email, and password. Note that users cannot change their roles/permissions using the profile page.

Comparing Implementation with other approaches:

Looking back, I would have changed the way I implemented the roles/permissions. A better way of configuring these would be to remove the user role/permission as a choice when registering for an account. Once the user role/permission is removed both the charity members and beneficiary's functionality would need to be changed. Charity members would then not be able to create events and beneficiaries would be the only users able to create events. To stop users from abusing the choice between charity members and beneficiaries there would need to be a clause that if a user wishes to become a beneficiary, they will have to pay a fee to the charity when they create an account.

In addition to removing the user option from the registration system, I would also add a way for charity members to apply to be beneficiaries after their accounts were created. This would also need a clause where if a charity member wishes to become a beneficiary they would be required to pay a fee to the charity.

1.2 Requirement Two Critique:

Members use the website to register their products and services, and beneficiaries use the system to acquire tokens.

Description of Implementation:

In the current state of the website, both charity members and beneficiaries can create an event for other users to book tickets to. A charity member or beneficiary can do this by logging into the website, then navigating to the dashboard page and clicking the 'create event' button. The user will then be redirected to the Create Event page. This page contains a form with all the necessary inputs such as title, description, date, location, ticket price, total number of tickets available, and checkboxes for the related accessibility features that the event will contain. Once all relevant information is inputted into the form the user can click the 'create event' button at the bottom of the form to save the event to the database. The event will then show up for all users to see on the events page for users to book tickets.

Users who created an event can view the status of their event by navigating to the dashboard located in the navigation bar. The events that the logged-in user has created will be listed and displayed to the user with the following information. The title of the event, the date of the event, and how many tickets are still available. When a ticket is booked the number of tickets will be decreased by one each time. When an event has no tickets left the event will still be listed on the events page but the 'book now' button will be unable to be clicked.

When any user books a ticket to the event, the ticket will be displayed to the user by navigating to the My Tickets page located in the navigation bar for the tickets to be viewed at the user's leisure. Each ticket will show the user the title of the event, the date of the event, the location of the event, and the token issued to the ticket that is unique to the user who booked the event.

Comparing Implementation with other approaches:

There are many ways this system could be improved. Regarding a booked ticket, there is little functionality on the My Tickets page for the user to interact with. There is no way to remove the ticket from the user's account and no way to view more details about the event or ticket from the My Tickets page. The same goes for a created event viewable on the dashboard page. There is no way for the user who created the event to take it down, nor is there a way for the event to be automatically removed once the date of the event has passed.

1.3 Requirement Three Critique:

Interactive elements that engage the website user.

Description of Implementation:

There are two notable interactions that users are able to take advantage of. The first is the booking of a ticket to an event. Users can view all the available events on the events page and book a ticket from there. The second, however, is on the accessibility page. Users can scroll down to the bottom section of the accessibility page. Here the user is greeted with a form that allows them to select from multiple checkboxes to find events that are relevant to their needs. Once the relevant checkboxes are selected and the 'find matching event' button is clicked a list of the relevant events will be listed below for the user to view.

Comparing Implementation with other approaches:

Additionally, there should be a way for the users to interact with the 'find matching events' form on the events page as well as the accessibility page. This would reduce the need for the user to navigate to the accessibility page and scroll down to the bottom to find the form. Maybe the form could be placed on a separate page and accessed via a button on the events page.

1.4 Requirement Four Critique:

The system provides an interface that the members can use to accept a token in a transaction.

Description of Implementation:

Users accept a token by booking an event. This is achieved by the user being logged in to their account and clicking the 'book now' button on a selected event. Once a user books an event a token is accepted by the user and the ticket is saved in the database with a reference to the user and the event. The ticket is then viewable on the My Tickets page and provides the user with the ticket token given to them when the ticket was booked.

Comparing Implementation with other approaches:

Another way of incorporating a token system into the user interface would be to include more services other than events. Such services could be booking appointments with other community members. These services could include discussing best steps and practices based on the accessibility needs of the charity member booking the appointment. This may require another role/permission to be created that allows specific users to list their services on the website.

1.5 Requirement Five Critique:

Store and retrieve data from a server-side database.

Description of Implementation:

Three sets of data are stored in the database. The first is the user's data. This data contains the user's username, email address, password, and role/permission. The second set of data is the events created by users with the relevant roles/permissions. This data includes the title of the event, a description of the event, the date the event will be hosted, the location of the event, the price of the event, how many tickets are available, the date and time the event was created, and the user's unique ID and a unique ID given to the event by MongoDB. The last type of data contained in the database is ticket information. The data for tickets include a unique ID for the ticket given by MongoDB, the unique ID of the event the ticket was booked for, the unique ID of the user who booked the ticket, a generated token, the status of the ticket, and the time and date the ticket was booked.

The user's data is stored in the database when a user creates an account via the registration page and is retrieved when a user logs into their account via the login page. Event data is stored in the database when a user with the correct roles/permissions creates a new event and is retrieved by the database to be displayed to users on the events page. Tickets are stored in the database when a user books a ticket and is retrieved and shown to the specific user that booked that ticket on the My Tickets page.

Comparing Implementation with other approaches:

Another way of establishing data storage and retrieval would be to use a different type of database. For example, the current version of the database uses MongoDB Atlas. This is connected using a Dotenv file and a database configuration file that takes the Mongo URI connection string to establish communication between the database and the front end of the website. Instead of using MongoDB, another database solution could be used such as MySQL. This would require changing the current Dotenv file and configuration file to suit the changes. However, this would also require you to write the SQL in software such as MySQL Workbench.

2. Part Two: Researching and Implementing Emerging Technology

Description and Explanation of Emerging Technology:

AI code generation tools offer several benefits to developers. AI code generators can quickly and easily produce developers with advanced boilerplates that eliminate repetitive and time-consuming coding tasks. In turn, they are also capable of detecting bugs and are fantastic at suggesting potential fixes (GitLab, 2024). This is achieved by analysing the existing codebase based on a prompt-and-response system and uses machine learning and data collection to determine the best course of action (Shaha, 2024).

Overall, AI code generators are faster and ultimately more efficient at producing quality building blocks for developers to build upon. Because of the way AI generators can produce quality components at the start of a project, it can help reduce the number of bugs in the less attended-to sections of the codebase. This in turn enhances the productivity of developers and reduces the human workload allowing developers to spend more time on more important and complex problems (GitLab, 2024).

However, AI code generators have a downside when they are relied upon too much. This can lead to messy and broken code that has several issues. This has the opposite effect on developers. Instead of AI helping reduce the workload for developers, developers can spend more time attempting to fix the code that has been given to them.

In February of 2024, GitHub surveyed the impact of AI on developer's experiences. Over 500 U.S.-based developers at over 1000 companies were asked how AI was affecting their workloads and how much it was being used. The key findings of this survey were overwhelming. 92% of respondents are currently using a form of AI coding tools both inside and outside of the workplace (Shani, 2024). This leads us to believe that developers are embracing AI code generation tools and are enjoying their experiences using them.

Description of Implementation that Supports Requirements:

The AI tool that was used to help with the development of this website is Bolt. Bolt is developed by Stack Blitz and is integrated into the online Stack Blitz IDE. With the AI tool being partly built into the Stack Blitz IDE is that developers can go from using a powerful AI code generation tool that evaluates the entire codebase they can then open the given solution into the IDE for further customization. In addition to this, users can take that same codebase and transfer it back to the AI tool if necessary. This can be done indefinitely allowing users to get the best of both worlds. Accessing a powerful AI code generation tool and a dedicated IDE.

For this project, I gave Bolt a few prompts and the website requirements for it to produce for me a boilerplate that I could work from without spending several hours creating files and assembling components for later use.

The first prompt I gave Bolt was. “I want to create a website using the MERN stack that uses MongoDB Atlas and incorporates JWT.” I then gave Bolt the five website requirements and my connection string to MongoDB. From there I gave Bolt a basic understanding of the requirements I wanted based on the previous milestones. “I want the users to be able to create events and book tickets to those events.

Bolt then created most of the necessary files and structured the codebase. Using the files it created Bolt then implemented several react components, configured a basic navigation bar, and hooks for the login and registration systems. Following this it produced basic routes for events, tickets, and authentication along with basic middleware and index files. Lastly, Bolt created the schema models for events, tickets, and users.

I then exported the codebase from Bolt to Stack Blitz to review the code and attempt to run the codebase. It didn’t work as expected. I was at a loss as to why it wasn’t working. After some searching around online, I found out that some systems such as MongoDB cloud services were not compatible with the online Stack Blitz IDE. So, to get the codebase running as intended I downloaded the code base as a zip and imported it into Visual Studio Code.

Initially, as I got the front end running, I realised that there were several additional things that I either missed when providing prompts to Bolt or Bolt didn’t pick up on everything I wanted from my prompts.

From here I started to edit the codebase creating the configuration file for MongoDB and adding the necessary pages that were left out by Bolt such as the profile page and my ticket page. I then managed to get the connection to MongoDB working and everything was running as expected.

I wanted to use Bolt again to review my changes to make sure everything was correctly set up before I went any further. So, I attempted to import the updated codebase back into Stack Blitz. When I did, I found out that there was no option for the use of Bolt anymore. Again, after some research, I found out that Bolt can only be used when a codebase is constantly connected to it. So, I went back to Visual Studio Code and carried on the old-fashioned way.

I had to do a fair amount of refactoring a large chunk of the codebase including adding more front-end files to match the given website requirements. Once everything was running as intended, I started to incorporate the features that Bolt missed when initially creating the codebase. These are things such as the profile page for users to update their details, a tickets page for users to view their booked tickets, and numerous other changes to make sure that all the requirements were met, and the website was running as I wanted it to.

3. References

GitLab. (2024, May 22). *AI code Generation explained: A developer's guide* | GitLab. GitLab.

<https://about.gitlab.com/topics/devops/ai-code-generation-guide/>

Shah, T. (2024, July 15). AI Code Generators: An In-Depth Guide to How They Work. *AI Code*

Generators: An In-Depth Guide to How They Work. [https://zencoder.ai/blog/how-ai-](https://zencoder.ai/blog/how-ai-code-generators-work)

[code-generators-work](https://zencoder.ai/blog/how-ai-code-generators-work)

Shani, I. (2024, February 8). *Survey reveals AI's impact on the developer experience* - *The*

GitHub Blog. The GitHub Blog. [https://github.blog/news-insights/research/survey-](https://github.blog/news-insights/research/survey-reveals-ais-impact-on-the-developer-experience/)

[reveals-ais-impact-on-the-developer-experience/](https://github.blog/news-insights/research/survey-reveals-ais-impact-on-the-developer-experience/)