Rowing Club Web Application

**Author Name: Erin Rutan**

**Professor: Chris Kreider**

**Executive Summary:**

I am going to create a web application for the Rowing Club at Christopher Newport University. This application will be used by all club members to track events (practice, races, and social events), availability of club members, and attendance at practices. Additionally, this application will include a roster where club members share their contact information. This application will ease the administrative burden on the e-board as well as enhance club communication and information sharing with everything being in one place.

Table of Contents

Introduction 3

Project Background 3

Project Description 3

Proposed Solution 3

Development Approach 3

High Level Plan 4

Project Schedule 4

Work Breakdown Structure 4

Project Calendar 4

Intermediate Milestones 4

Intermediate Milestone #1 4

Intermediate Milestone #2 4

Weekly Updates 5

Weekly Update #1 5

Weekly Update #2 5

Weekly Update #3 5

Weekly Update #4 5

Weekly Update #5 5

Weekly Update #6 5

Weekly Update #7 5

Weekly Update #8 5

Weekly Update #9 5

Weekly Update #10 5

Gantt Chart 6

Project Deliverables 6

Conclusion 6

Appendices 7

Appendix A - <How to Build a Web Application> 7

# Introduction

The purpose of this project is to create an application for the rowing club here at CNU to make it easier to plan events, track availability and attendance, and manage the team. Currently everything is tracked manually, and information is shared through email. This means things can get lost or confused. With an application, everything will be in the same place and easy to find.

## Project Background

I have been in the rowing club here at Christopher Newport University since my freshman year, and I have been rowing for 7 years. I joined the e-board of the rowing club my sophomore year and have held various positions in e-board from historian to scholarly chair, and now president. I have seen the way the club has been run in the past, and I think this app could help ease the administrative burden and improve team communication. In addition, I worked on webpage components and some aspects of website development during my summer internship at ADP. This experience, as small as it may be, should help me in the development of this web application.

## Project Description

When completed, this project will be a functional web application that will ease the administrative burden on the rowing club. The application will be web based and use responsive web design, so it can be used by all club members on any device. Using a web application as opposed to developing an application specifically for Android or iOS will allow this app to be used by a greater audience and be more useful to full club communication. More specifically, the application will have a feature where e-board members can schedule upcoming practices, races, social events, etc. Club members can indicate availability for each event. Attendance can be tracked by e-board members. Information can be shared among club members in one central platform.

# Proposed Solution

## Development Approach

I intend to take an agile approach. During my summer internship, we worked in a scrum environment with daily stand ups and one-week sprints. I have a little bit of experience working in an agile environment, so I think it will help me focus and continue to get things done on this project throughout the semester. I like the idea of the Kanban system, so I intent to try this approach for this project. With my existing agile experience, I think it will benefit me to continue to work in such an environment. I believe that using Kanban boards will help me track progress as well as what needs to get done. Hopefully, this will help me to not bounce between tasks too much, which is something I have found myself doing recently. Plus, there is always the rewarding feeling of finally moving something into the done category to motivate me to get things done.

## High Level Plan

To develop this rowing club web application, I will be utilizing Amazon Web Services. Hosting will be done through Amazon Amplify. The database will be constructed using Amazon Dynamo DB. An API will be created with Amazon’s REST API. User authentication will be managed through Amazon Cognito. The front end of the application will be developed using HTML, CSS, and JavaScript, potentially using a framework such as React. Back end will be written using Python and potentially the Django framework. GitHub will be used for version control.

Seeing as this is not something I have done before; I did a lot of Googling to find resources that would help me figure out everything I would need. Some of these sources relate specifically to using Amazon Web Services to host the web application. All such sources can be found in Appendix A.

# Project Schedule

## Work Breakdown Structure

*Major Tasks to be completed:*

* *Define functionality – 3 hours*
* *GUI mockup/ web application sketch – 2 hours*
* *Wireframe/prototype web application – 2 hours*
* *Setup of AWS hosting service – several days*
* *Database design – 5 hours*
* *Database Implementation – several days (?)*
* *API development – weeks?*
* *Front end code development – several weeks*
* *Back end code development – several weeks*
* *Testing – at least one week*
* *Deployment/ Application goes live – several days (?)*

## Project Calendar

### Intermediate Milestones

#### Intermediate Milestone #1 (3/8)

* Database connected to front end of web application, Log in page constructed, home page set up and visually available for user.

#### Intermediate Milestone #2 (4/5)

* Web application has all features visually present for users to interact with. Users can log in and move around the app from feature to feature. Features available include schedule, roster, and attendance.

### Weekly Updates

#### Weekly Update #1 (2/17)

* Researching/studying Frameworks (5 hours)
* Database building (3 hours)
* Additional research and learning (2 hours)

#### Weekly Update #2 (2/24)

* Database Set up (5 hours)
* Research (3 hours)
* Debugging (2 hours)

#### Weekly Update #3 (3/3)

* Connect front end to database (4 hours)
* Start setting up log in page (2 hours)
* Create roles with different levels of access (2 hours)
* Debugging (2 hours)

#### Weekly Update #4 (3/10)

* Refine login page (3 hours)
* Set up home page/ menu to access each feature (4 hours)
* Debugging (2 hours)

#### Weekly Update #5 (3/17)

* Set up calendar/scheduling page front end (3 hours)
* Set up calendar functionality (5 hours)
* Debugging (2 hours)

#### Weekly Update #6 (3/24)

* Set up Roster front end (3 hours)
* Configure Roster functionality (4 hours)
* Debugging (2 hours)

#### Weekly Update #7 (3/31)

* Set up Attendance page (3 hours)
* Configure attendance functionality (4 hours)
* Debugging (2 hours)

#### Weekly Update #8 (4/7)

* Finish configuring application functionality (6 hours)
* Debugging (3 hours)

#### Weekly Update #9 (4/14)

* UI improvement (5 hours)
* Debugging (2 hours)

#### Weekly Update #10 (4/21)

* Testing/fine tuning (5 hours)
* Debugging (3 hours)
* Creating final presentation (2 hours)

## Gantt Chart

Attached separately as a PDF document.

# Project Deliverables

Documentation:

* Database diagrams (ER Diagram)
* Stakeholder Identification: Discussion of target users and how they will be able to use the application.
* User stories: as tracked using Kanban agile system
* GUI sketch: preliminary sketch(es) of GUI interface of web application
* GUI Wireframe: generated using an online resource, following GUI sketch
* System architecture diagram: graphical representation of the architecture for the entire application system

Technical Deliverables:

* Link to working web application
* Link to Git repository with complete project code
* Instructions document explaining how the instructor can test the system demonstrated, both from the URL of the active web application, and if the instructor wants to set it up themselves.

Final presentation Deliverables:

* PowerPoint presentation

# Conclusion

The project I intend to develop is a web application for the rowing club at CNU. This application will be available to all members of the club. It is intended to enhance communication between club members and ease the job of the e-board. This application will allow for creation of upcoming events including practice, races, and social events. The e-board will be able to track attendance at practice as well. Everything club members need to know what is going on in the rowing club will be in one central location.

This application will require database development, front end, and back end development as well as a hosting service. Most of these things will be handled using Amazon Web Services. Front end development will use HTML, CSS, and JavaScript, while back end development will feature Python. Frameworks may be used to aid in the creation of the application.

This project will take approximately 10 weeks and will be worked on throughout the remainder of the spring semester.

I am somewhat concerned about all the technologies I have listed here that I have little to no prior experience with. Most of it seems straightforward, but it is certainly scary to look at the whole project ahead of me.

# Appendices

## Appendix A - <How to Build a Web Application>

“How to Build a Public Facing API - Amazon Web Services (AWS).” *Amazon Web Services,*

*Inc.*, 2021, aws.amazon.com/startups/start-building/how-to-build-a-public-facing-API/.

“How to Build a Simple Web App Using AWS.” Performance by Mat Werber, *Youtube*, Amazon

Web Services, 8 Jan. 2021, www.youtube.com/watch?v=voa0btsVSfk.

“How to Build a Simple Web Application - Amazon Web Services (AWS).” *Amazon Web*

*Services, Inc.*, 2021, aws.amazon.com/startups/start-building/how-to-build-a-web-app/.

Johnston, Joe. “How to Build a Web App: A Beginner’s Guide (2020).” *Www.budibase.com*, 23

Apr. 2019, www.budibase.com/blog/how-to-make-a-web-app/.

‌