WARE

Web Application for Robotics Education

University of Nevada, Reno
Department of Computer Science and Engineering

Project Prototype

Team 17
Zachery Wiles, Ryan Lunt, Sean Griffith, Herman Hira

Instructors: Sergiu Dascalu and Devrin Lee

External Advisor: Ben Gallagher

December 11, 2020

1. Table of contents

1. Abstract

WARE is a web application that can be utilized by students engaged in robotics classes. The main goal of WARE is to provide students with a platform in which they can explore robotics principles and experiment in various environments to reinforce in-class instruction. At the same time, WARE has minimal performance costs associated with compiling user code. WARE will consist of multiple environments with varying goals and mechanics that students can experiment with to test and expand their knowledge. In a classroom format, instructors will be able to evaluate the performance of a student in a given environment. The purpose of this document is to describe the functionality of the prototype for WARE as well as challenges encountered during development. Additionally, feedback obtained from meetings with project stakeholders will be summarized and changes made to the project's design and requirements according to prototype feedback will be discussed.

2. Introduction

The goals of developing the Web Application for Robotics Education are to give robotics students an interactive experience in learning how to code robots, make it possible for users of smaller devices to utilize the application, and allow robotics instructors to set up virtual classrooms to track student progress. With the rise of modern web technologies and the prevalence of using mobile devices for learning, WARE will help meet the educational demands of today's robotics students and instructors.

Since the submission of the design report, the team has started on a WARE prototype. A basic Flask server instance has been coded to implement essential functionalities such as serving individual web pages to achieve some core functionality. Next, code compilation, a critical component of WARE, was implemented within the Flask framework. Both a visual representation of the robot and a terminal output were developed to show the final results of the code compilation. Finally, Jinja HTML templates and CSS code were developed for some of the core pages of the WARE website.

One significant requirement that has been added to the project is the hosting of the WARE application. Currently, the prototype is being developed and tested as a local instance that is bound to a port on the host machine. After discussing the new requirement with the project sponsor, Dr. Rui Wu, it was determined that the team will set up a basic server hosted on Amazon Web Services or Digital Ocean Droplets to allow users to interact with the application over the internet.

3. Prototype Objectives and Functionality

The main objectives in developing the prototype were to showcase some of the essential functionalities of what makes WARE unique in the software landscape. Specifically, the team decided to dedicate substantial effort towards creating an environment page in which the user can input code of their choosing and see the results generated by the back-end server. Additionally, the team also spent some time creating and designing some of the web pages of WARE for both desktop and mobile platforms.

Four components were successfully implemented by the team for the prototype: homepage, environments list page, a single environment page, and back-end code compilation. The team found it necessary to first develop some of the webpages in order to provide a design framework in which all other pages would be based on. Additionally, the team decided to develop some of the back-end in order to demonstrate how a user would interact with and receive feedback within an environment. For the homepage, the main goal was to create a good first impression to the user about the capabilities of WARE. The environments list page was created to allow a user to view all of the different types of environments to explore. A single environment page was developed to provide a testing ground for the back-end code compilation. Finally, back-end code was developed to retrieve form input for user code submission, compile the code, and display the results.

Some of the components we decided not to implement for the prototype were user account management, web hosting, and user documentation. The team did not have extensive knowledge in these areas of concern at the start of prototype development. Thus, implementing them in the allotted time would have been infeasible. Specifically, creating a user login process requires developing and testing important security requirements. Furthermore, deploying the prototype on a remote server would have involved obtaining a domain name, deploying the server, and checking for security vulnerabilities. Finally, creating user documentation will require thorough reading of the OpenGym AI documentation to fully understand what content will be useful to robotics students.

4. Prototype Development

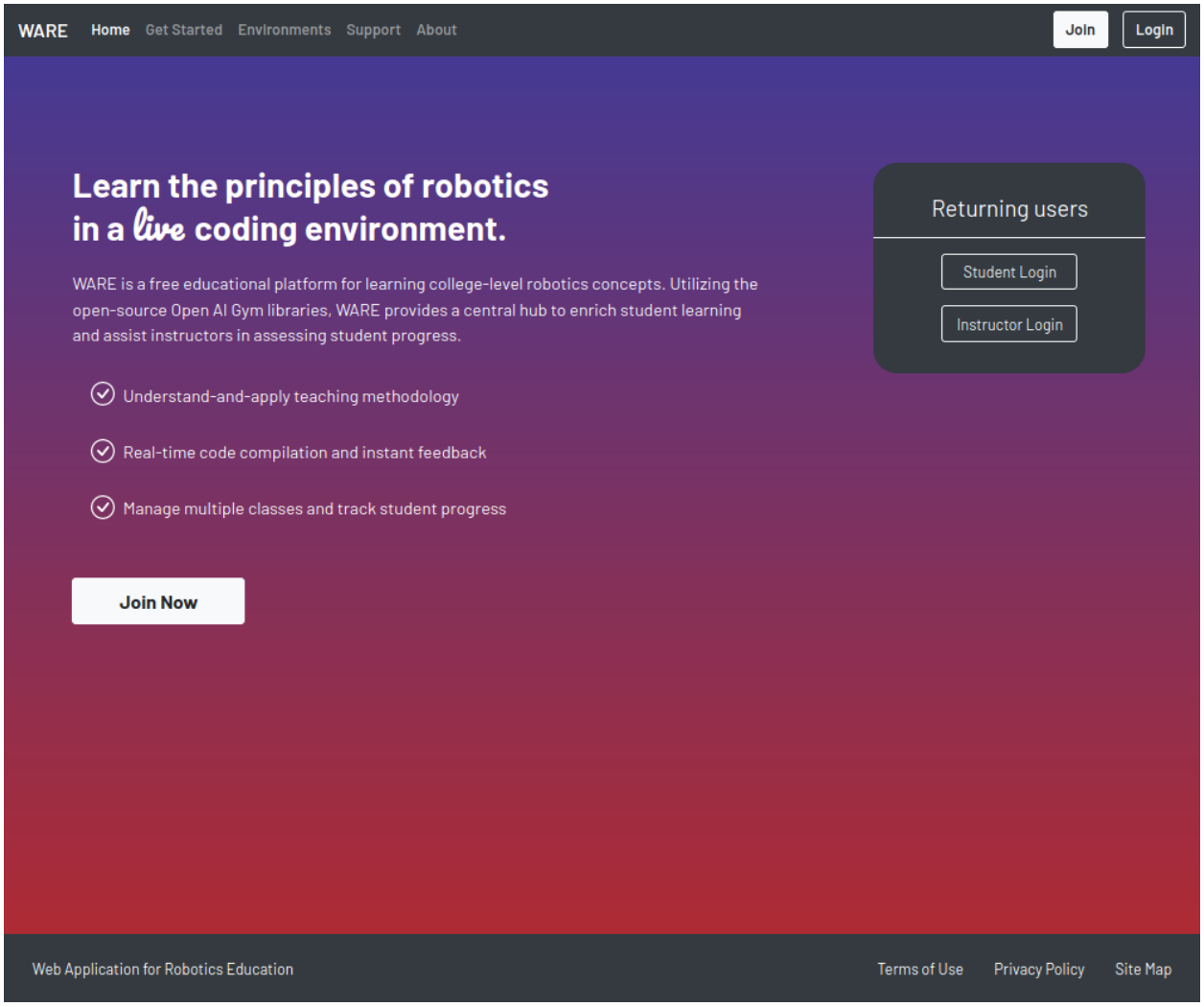

Figure 4.1: The template for the homepage. This is what visitors and potential users will see first upon visiting the website. The buttons displayed will be usable in the future when we begin implementing account creation.

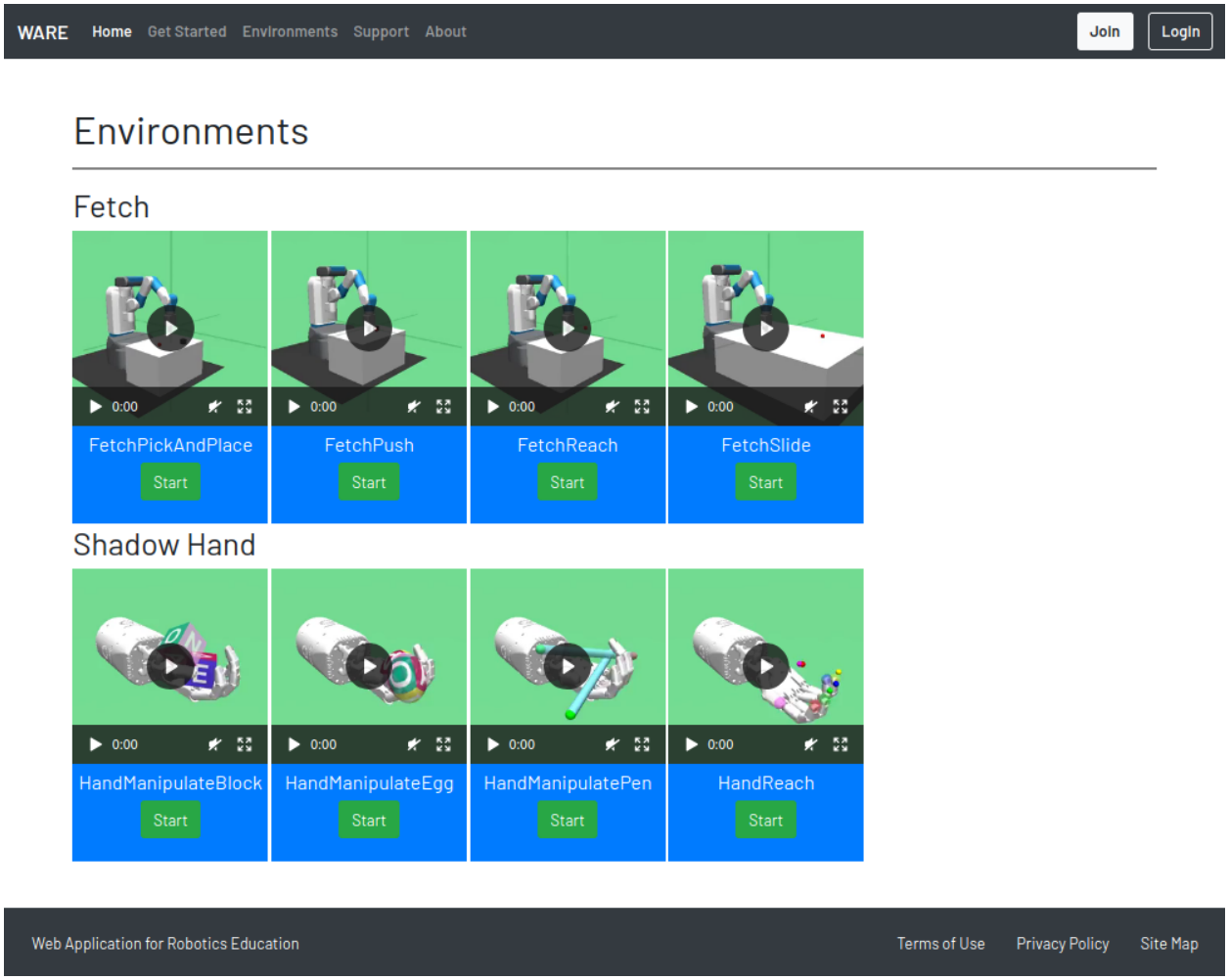

Figure 4.2: Environment selection page with eight different environments. The user will be able to select an environment which will then lead to the singular environment page. Each environment has its own short preview video.

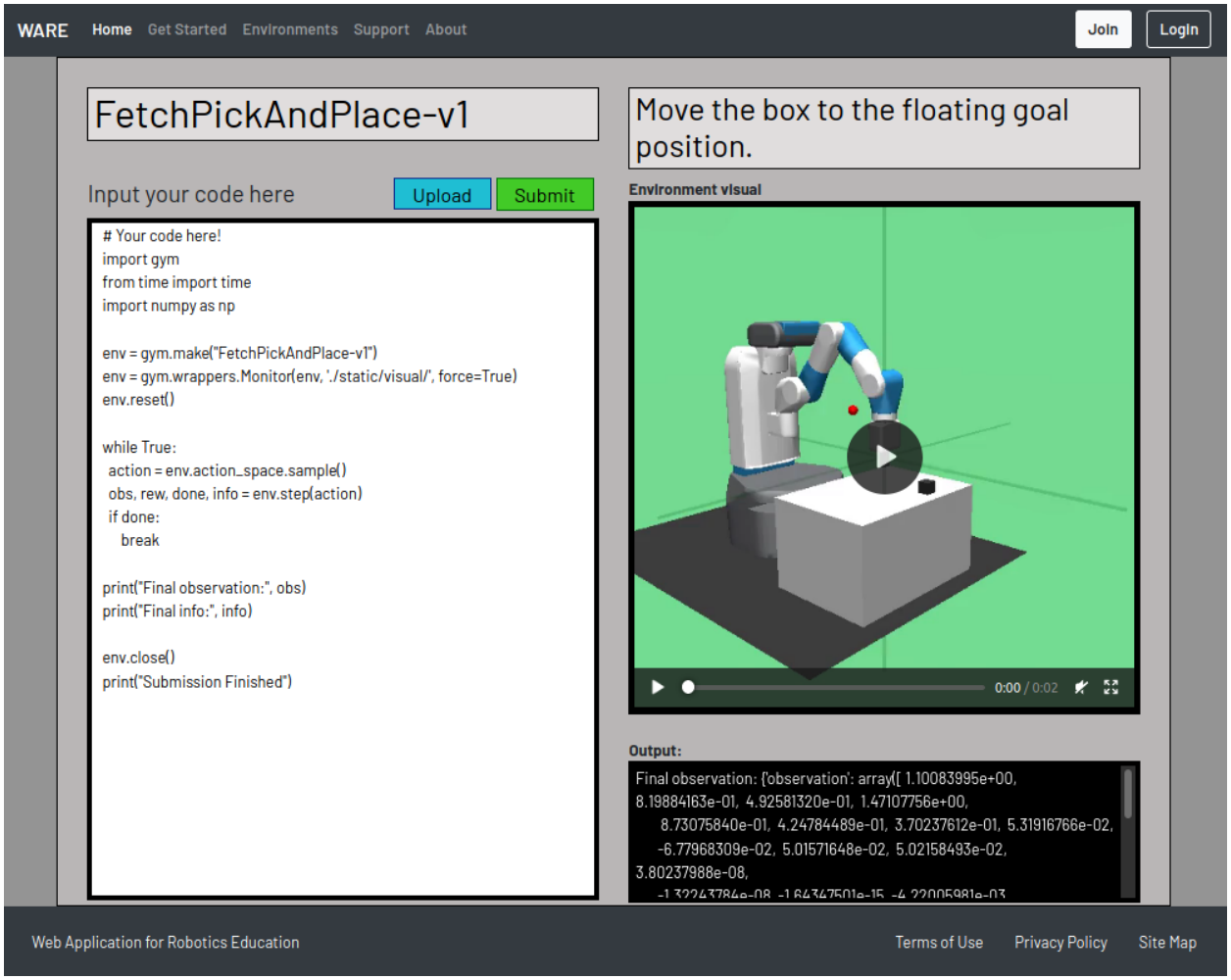

Figure 4.3: This is one of the eight environment pages (as seen in Figure 4.2) where the user will be programming or uploading their Python code to achieve the objectives set for the environment. Users will have a visual showing what their code has done as well as textual results printed at the bottom right upon pressing the green submit button.

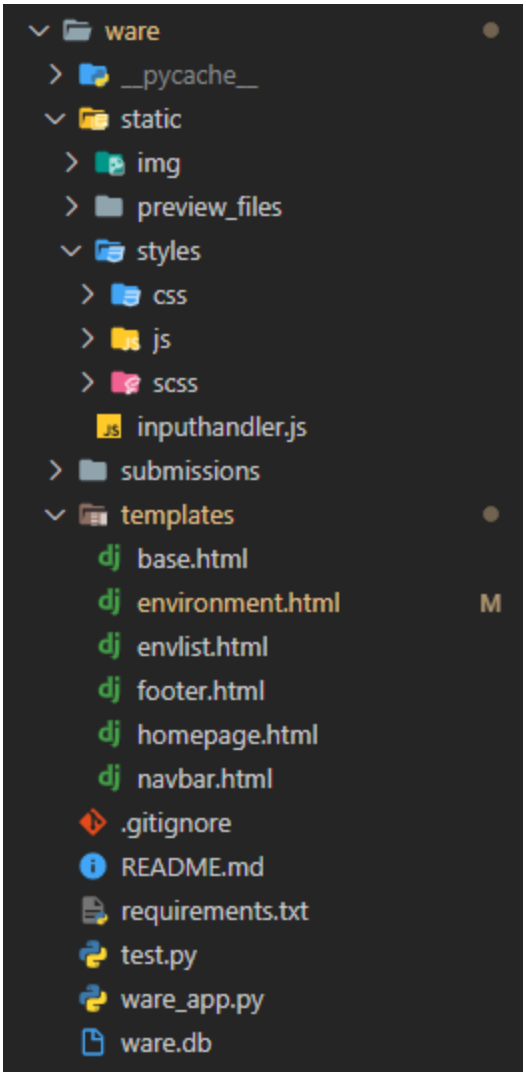

Figure 4.4: This was our code editor's explorer showcasing our various files and folders. Above you can see our templates, three of which we showcased in the demo, as well as our fourth component 'ware_app.py' which is our backend written in Python.

5. Prototype Evaluation

A meeting was set up with Dr. Rui Wu on December 9th to evaluate the prototype. He was very pleased with the fully functioning environments that could run and compile code as well as catch errors. Some suggestions he had to change and improve the web application was to utilize a code mirror in the text box so that key phrases would be in different colors for students to better differentiate and organize their code. Another suggestion Dr. Wu had was to add a help / hint section to the singular environment pages after the user fails to successfully complete the environment. We also discussed how to set up the login using flask-login as well as implementing the professor view to select certain environments for their students to use. A

potential issue that was discussed was how students would be able to press the submit button multiple times which may cause potential issues. The suggestion was to add a notification that would indicate to the user that their code was compiling, and disable the button for further submissions. Another issue discussed was multiple users submitting code at the same time. As of now, the program compiles on a local machine. Dr. Wu suggested that we implement a database to handle this problem. In this way, the users will be able to individually compile their code at the same time.

6. Prototype Demonstration

The demo time was on December 8th at 1:30pm. One major suggestion given was making sure we create a tutorial/documentation for students to utilize, since WARE is a teaching tool, it needs to have a way for students to learn and understand what is being taught, so that they can apply it to their environments they are working on. Other suggestions were a focus on setting up login for accounts and also start the process of thinking of security and how to best protect users as well as the web application itself. For security that means setting up a protected database for user login info as well as making sure bad actors do not have a way to enter malicious code to affect the backend through the online ide/compiler.

7. Software Design Modifications

Throughout our demonstrations of the WARE project prototype to both the project sponsor Dr. Rui Wu as well as the instructors, we have noted several modifications that should be made to our project's design. Although our project prototype was well received by Dr. Wu, there were several suggestions he had to improve our project overall. One suggestion made by Dr. Wu was to add animations - such as a placebo loading bar or animation for whenever the submit button is pressed. This would be to make the application seem more professional while also letting the user know results are being processed. Another suggestion made by Dr. Wu was to save submissions by the user to the database instead of a separate file because this would allow multiple people to submit their code at the same time. Reprogramming sections of the backend to allow for that will not be too substantial of a change.

Additionally, Rui found our implementation of the code input section of the environment page unappealing. In order to combat this, it was suggested that we improve the user's development experience with regards to code input through the use of a javascript component known as CodeMirror in order to provide the user python syntax highlighting that would contribute to a more effective and development process. Finally, it was suggested to implement a hint section for each individual environment where hints would be displayed to the user after a failed attempt at completing the environment. This could likely be added as a tab where the text results box is.

In addition to feedback obtained from the prototype demonstration with our project sponsor, several suggestions were made during the prototype demonstration with the instructors. The main suggestion provided to our team during this demonstration was to provide students with

documentation or tools that would enable students to seek help with their implementation of a solution to the environment. In order to accommodate this need, we plan to implement a help section for each environment page with links to the environment's documentation as well as hints that would provide students with direction on how to proceed with their implementation of a solution.

8. Team Contributions

Sean Griffith
Total time committed: 13 hours
Developed the Flask server back-end using Python code to host the WARE web app and process user code submissions. Assisted team members in resolving technical issues with prototype development. Created the cover page and table of contents for this report, and assisted in writing the Abstract and Software Design Modifications sections of the report.

Ryan Lunt
Total time committed: 9 hours
Designed and developed the single environment page using HTML and CSS code. Wrote the Abstract, Prototype Development and Software Design Modifications sections for this report.

Herman Hira
Total time committed: 7.25 hours
Designed and developed the environments list page using HTML and CSS code. Wrote the Prototype Evaluation and Prototype Demonstration sections for this report.

Zachery Wiles
Total time committed: 9.75 hours
Designed and developed the homepage using HTML and CSS code. Wrote the Introduction, Prototype Objectives and Functionality sections for this report.