



Project Title	CoderDojo Zen Projects
Author	Graham Edmond Bartley
Supervisor	Dr. Geoff Hamilton
Document Type	User Manual
Student Number	14541017
Completion Date	12/05/2018

Abstract

This project aims to build a project creation, management and interaction solution for the CoderDojo Foundation (CDF) and their community platform Zen which is currently used to manage over 1,000 Dojos (coding clubs) and their events across the world. This solution allows youths to share software projects they have worked on by uploading their code which is automatically stored and versioned using GitHub. Projects can then be run and interacted with directly in the browser including Python 3, Node.js, HTML5 and Java projects.

Table of Contents

All Users	3
Searching all projects	3
Searching projects in a Dojo	3
Viewing & running projects	3
Youths	4
Checking if your Dojo allows projects	4
Creating a project	4
Sharing your project	5
Updating your project	5
Deleting your project	6
Champions	6
Accessing the administrator panel for your Dojo	6
Integrating a GitHub account with your Dojo	6
Searching projects in your Dojo	7
Editing & deleting projects in your Dojo	7
Removing a GitHub integration	7
Developers & Contributors	8
Setting up the development environment	8
Prerequisites	8
Steps	8
Loading test data	9
Running tests	10
Backend	10
Frontend	10
CoderDojo Foundation (CDF) Administrators	11
Editing & deleting projects	11
Adding support for different runtimes	11
Screenshots	13

All Users

Searching all projects

1. Navigate to the project list page. (*Fig. 1.0.*)
2. All public projects can be found here.
3. Most played, recently updated and recently created projects are displayed at the top.
4. The search bar can be used to search all projects by name or description.
5. Clicking on the name of a project will take you to the page for that project.

Searching projects in a Dojo

1. Navigate to the details page for the Dojo you want. (*Fig. 3.0.*)
2. If this Dojo has a GitHub integration, all projects created in that Dojo will be displayed here.
3. The search bar can be used to search projects by name or description.
4. Clicking on the name of a project will take you to the page for that project.

Viewing & running projects

1. Navigate to the project page for the project you want to view or run. (*Fig. 2.0.*)
2. Information about the project can be found on this page such as:
 - a. Name
 - b. Description
 - c. Technology
 - d. Author name & link
 - e. Dojo name & link
 - f. Source code link
 - g. Resource link
 - h. Last updated time
 - i. Number of plays
3. If the project is a HTML5 project then the running website can be viewed on this page under the “Try it out!” heading. Clicking the “Visit the website” link will allow you to view the website in a new window. (*Fig. 2.3.*)
4. If the project uses a different technology to HTML5 then you will see a button labelled “Play” under the “Try it out!” heading. Clicking this button will take you to the project runtime page. (*Fig. 2.0.*)
5. You will see a loading message for some time (length of time will vary with internet connection). (*Fig. 5.0.*)
6. Once the loading message has disappeared you will see a box containing the running project which you can interact with. (*Fig. 5.1.*)
7. You can click the “Back to Project” link to return to the project page when you are finished.

Youths

Checking if your Dojo allows projects

1. Navigate to the details page for your Dojo. *(Fig. 3.0.)*
2. On the sidebar look under the heading “GitHub Integrated”
 - a. If a green tick is displayed, the Dojo allows projects *(Fig. 3.0.)*
 - b. If a red cross is displayed, the Dojo does not allow projects *(Fig. 3.1.)*
3. If your Dojo does not allow projects you can contact a champion of your Dojo and make a request to allow this.

Creating a project

1. You will need some code written using one of the supported technologies before you can create a project. The supported technologies are:
 - a. Python 3
 - b. NodeJS
 - c. HTML5
 - d. Java

You will also need to be a member of a Dojo on Zen which allows project creation. For more information on this go to the “Checking if your Dojo allows projects” section.
2. Once you have some code to upload, create a .zip file containing all of the code and make sure that the main file of your code is directly inside the .zip file (not inside a sub folder).
3. Log in to Zen using your Zen login details.
4. You will be taken to your profile page. *(Fig. 6.0.)*
5. There are two ways to access the project creation form
 - a. Click the grey bubble with the plus symbol inside of it labelled “Create a Project” under the “Projects” header. *(Fig. 6.0.)*
 - b. Navigate to the project list page. In the orange box which says “Got an idea to share?”, click the “Create a Project” button. *(Fig. 1.1.)*
6. You will be taken to the project creation form. *(Fig. 7.0.)*
7. Fill in the following information:
 - a. Name - give your project a memorable name
 - b. Type - select the technology your project uses
 - c. Description - give your project a useful description
 - d. Dojo - select the Dojo you want to associate this project with
 - e. Resource - optionally give a link to a resource you used when creating your project
 - f. Main filename - give the name of the main file of your project. You do not need to give a file extension as it will automatically be added based on your project technology. If your technology is HTML5 then you do not need to fill this field in.

- g. Files - click the upload button and select the .zip file you created containing your project files. Upload the .zip file
8. Once you have filled the form in and uploaded your project .zip file, click the “Create Project” button.
9. If you see red text appear on the page that means there was an error with one or more of your inputs. Resolve the errors and try clicking the “Create Project” button again.
10. If there are no errors, the button will shrink and a spinner will appear.
11. After some time (length of time may vary with internet connection and project size) you will be taken to the project page for the project you created. (*Fig. 2.2.*)

Sharing your project

1. Navigate to the project page for your project. (*Fig. 2.0.*)
2. In the top right-hand corner under the purple banner you will see a share button. Click the share button.
3. A link will appear with a button to the right of it allowing you to copy the link. (*Fig. 2.1.*)
4. Give this link to others who will be able to navigate to it using a web browser to view and interact with your project.

Updating your project

1. Log in to Zen using your Zen login details.
2. Navigate to the project page for your project. (*Fig. 2.2.*)
3. In the top right-hand corner under the purple banner you will see an edit button. Click the edit button.
4. You will be taken to the edit project page for your project. (*Fig. 8.0.*)
5. You will see the following input fields containing your current project information:
 - a. Name
 - b. Description
 - c. Resource
 - d. Main file
6. Edit any of the fields to change the information.
7. To update project files, click the upload button and upload a .zip file containing the updated files as you did when creating the project.
8. Click the “Update Project” button to update the project with the new values.
9. If you see red text appear on the page that means there was an error with one or more of your inputs. Resolve the errors and try clicking the “Update Project” button again.
10. If there are no errors, the button will shrink and a spinner will appear.
11. After some time (length of time may vary with internet connection and project size) you will be taken back to the project page for your updated project. (*Fig. 2.2.*)

Deleting your project

1. Log in to Zen using your Zen login details.
2. Navigate to the project page for your project. *(Fig. 2.2.)*
3. In the top right-hand corner under the purple banner you will see an edit button. Click the edit button.
4. You will be taken to the edit project page for your project. *(Fig. 8.0.)*
5. Click the “Delete Project” button and a “Confirm Delete” button will appear.
6. Click the “Confirm Delete” button to delete the project.
7. The button will shrink and a spinner will appear.
8. You will be taken back to the project list page once the project has been deleted.
(Fig. 1.1.)

Champions

Accessing the administrator panel for your Dojo

1. Log in to Zen using your Zen login details.
2. Navigate to the Dojo page for your Dojo. *(Fig. 3.2.)*
3. Click the button with the pie chart symbol in the top right of the page.
4. You will be taken to the administrator panel for your Dojo. *(Fig. 4.0.)*

Integrating a GitHub account with your Dojo

1. You will need to create a GitHub account specifically for your Dojo which will be used to store the code for projects created by Ninjas in your Dojo.
2. Ensure you are not logged in to GitHub before starting.
3. Log in to Zen using your Zen login details.
4. Navigate to the administrator panel for your Dojo. *(Fig. 4.0.)*
5. Click the link with the text “Integrate GitHub Account” to begin the integration.
6. You will be brought to a GitHub login page.
7. Login using the details of the GitHub account you created specifically for your Dojo. Do not use your own personal GitHub login or any other one!
8. You will be brought to an authorization page asking for the following permissions:
 - a. Read and write access for all public repository data including:
 - i. Code
 - ii. Issues
 - iii. Pull requests
 - iv. Wikis
 - v. Settings
 - vi. Webhooks and services
 - vii. Deploy keys
9. Click the green authorization button to allow Zen these permissions for the GitHub account you logged in to.

10. You will be brought back to the administrator panel for your Dojo. (*Fig. 4.1.*)
11. You will now see a “Manage Projects” section where projects will show up once Ninjas in your Dojo create them.

Searching projects in your Dojo

1. Log in to Zen using your Zen login details.
2. Navigate to the administrator panel for your Dojo. (*Fig. 4.1.*)
3. If your Dojo has projects you will see them listed under the “Manage Projects” heading with a search bar above them.
4. Type a query in the search bar to search projects based on name and description.
5. The list of projects will update automatically based on your search query.

Editing & deleting projects in your Dojo

1. Log in to Zen using your Zen login details.
2. Navigate to the administrator panel for your Dojo. (*Fig. 4.1.*)
3. If your Dojo has projects you will see them listed under the “Manage Projects” heading.
4. Click the edit button with the pencil symbol next to a project to edit that project.
5. You will be taken to the edit project page for that project. (*Fig. 8.0.*)
6. You can edit the following information here:
 - a. Name
 - b. Description
 - c. Resource
 - d. Main filename
7. Click the “Update Project” button to update the project with the new values.
8. If you see red text appear on the page that means there was an error with one or more of your inputs. Resolve the errors and try clicking the “Update Project” button again.
9. If there are no errors, the button will shrink and a spinner will appear.
10. After some time (length of time may vary with internet connection and project size) you will be taken to the project page for the updated project. (*Fig. 2.2.*)
11. You can also delete the project by returning to step 5 and doing the following:
 - a. Click the “Delete Project” button. A “Confirm Delete” button will appear.
 - b. Click the “Confirm Delete” button to delete the project.
 - c. The button will shrink and a spinner will appear.
 - d. You will be taken to the project list page once the project has been deleted.(*Fig. 1.0.*)

Removing a GitHub integration

1. Please note that removing the GitHub integration for your Dojo will permanently delete all projects associated with it and Ninjas in your Dojo will not be able to create projects again until a new integration has been made.
2. Log in to Zen using your Zen login details.

3. Navigate to the administrator panel for your Dojo. (Fig. 4.1.)
4. Under the “GitHub Integration” heading click the “Remove GitHub Integration” button.
5. A “Confirm Removal” button will appear, click it to confirm removal of the GitHub integration. (Fig. 4.2.)
6. Once the integration has been removed, the administrator panel will display the link to “Integrate GitHub Account” again. (Fig. 4.0.)

Developers & Contributors

Setting up the development environment

Prerequisites

The following are required to get set up to run my project locally for development. Setting up these prerequisites is not explained in this user manual since they are not part of my project and ample resources are available online for them.

1. Computer with an internet connection running a Linux operating system (Fedora, Ubuntu, OpenSUSE, etc.).
2. Git installed
3. Node v8.9.3 or higher installed.
4. Docker installed
5. PostgreSQL installed and a local database set up.
6. A GitHub OAuth app to use for interactions with GitHub.

Steps

Clone my project repository using the following command (you will need to be authorized to access the repository)

```
git clone https://gitlab.com/bartleg3/2018-ca400-bartleg3.git
```

Install dependencies for the backend microservice

```
cd src/cp-projects-service  
npm install
```

You will need the following environment variables in order for the microservice to operate correctly:

PGUSER: A username of your local PostgreSQL database

PGPASSWORD: The password associated with the above username

PGDATABASE: The name of your local PostgreSQL database

PGPORT: The port your local PostgreSQL database is running on

GITHUB_CLIENT_SECRET: The client secret of your GitHub OAuth app

To start the microservice run the following, replacing <yourValue> with your own value in each case

```
PGUSER=<yourValue> PGPASSWORD=<yourValue> PGDATABASE=<yourValue>  
PGPORT=<yourValue> GITHUB_CLIENT_SECRET=<yourValue> npm start
```

This will run database migrations to set up the tables if not already set up and start the microservice at <http://localhost:3000>

Install dependencies for the frontend

```
cd ../cp-projects-frontend  
npm install
```

You will need the following environment variable in order for the frontend to operate correctly:

GITHUB_CLIENT_ID: The client id of your GitHub OAuth app

To start the frontend development server run the following, replacing <yourValue> with your own value

```
GITHUB_CLIENT_ID=<yourValue> npm start
```

This will start the frontend development server at <http://localhost:8080>

Loading test data

I have provided test data which can be used to populate your local database for use in testing and development.

You will need the following environment variables in order for the test data to be loaded:

PGUSER: A username of your local PostgreSQL database

PGPASSWORD: The password associated with the above username

PGDATABASE: The name of your local PostgreSQL database

PGPORT: The port your local PostgreSQL database is running on

To load the test data into your local database run the following from the cp-projects-service directory, replacing <yourValue> with your own value in each case

```
PGUSER=<yourValue> PGPASSWORD=<yourValue> PGDATABASE=<yourValue>  
PGPORT=<yourValue> npm run testdata
```

Running tests

I have written unit tests for both the backend and the frontend as well as integration tests for the endpoints on the backend.

Backend

To run integration tests on the backend you will need the following environment variables:

PGUSER: A username of your local PostgreSQL database

PGPASSWORD: The password associated with the above username

PGDATABASE: The name of your local PostgreSQL database

PGPORT: The port your local PostgreSQL database is running on

Please note that test data is loaded into your local database before each integration test is run and then again at the end of each section of tests in order to ensure each test is independent. This does mean running the integration tests will overwrite any existing data in your database!

To run the integration tests run the following from the cp-projects-service directory, replacing <yourValue> with your own value in each case

```
PGUSER=<yourValue> PGPASSWORD=<yourValue> PGDATABASE=<yourValue>  
PGPORT=<yourValue> npm run integration
```

To run the unit tests on the backend run the following from the cp-projects-service directory

```
npm run unit
```

To run both the unit and integration tests on the backend and generate a coverage report for them run the following from the cp-projects-service directory

```
PGUSER=<yourValue> PGPASSWORD=<yourValue> PGDATABASE=<yourValue>  
PGPORT=<yourValue> npm test
```

A full coverage report will be generated in the cp-projects-service/test/coverage/lcov-report directory and can be viewed in a browser.

Frontend

To run the unit tests on the frontend and generate a coverage report for them run the following from the cp-projects-frontend directory

```
npm test
```

A full coverage report will be generated in the cp-projects-frontend/test/unit/coverage/lcov-report directory and can be viewed in a browser.

CoderDojo Foundation (CDF) Administrators

Editing & deleting projects

1. Log in to Zen using your Zen login details.
2. Navigate to the edit project page for the project you want to edit/delete through either:
 - a. The project page for that project. (*Fig. 2.2.*)
 - b. The administrator panel for the Dojo that project belongs to. (*Fig. 4.1.*)
3. You can edit the following information here:
 - a. Name
 - b. Description
 - c. Resource
 - d. Main filename
4. Click the “Update Project” button to update the project with the new values.
5. If you see red text appear on the page that means there was an error with one or more of your inputs. Resolve the errors and try clicking the “Update Project” button again.
6. If there are no errors, the button will shrink and a spinner will appear.
7. After some time (length of time may vary with internet connection and project size) you will be taken to the project page for the updated project. (*Fig. 2.2.*)
8. You can also delete the project by returning to step 2 and doing the following:
 - a. Click the “Delete Project” button. A “Confirm Delete” button will appear.
 - b. Click the “Confirm Delete” button to delete the project.
 - c. The button will shrink and a spinner will appear.
 - d. You will be taken to the project list page once the project has been deleted. (*Fig. 1.0.*)

Adding support for different runtimes

Assuming you have the local development environment set up as outlined in the “Setting up the local development environment” section above and have access to the “coderdojo” organisation on DockerHub you will be able to add runtime support for different technologies using Docker.

Runtime containers pull their images from this repository

<https://hub.docker.com/r/coderdojo/project-runtime/> which I created on DockerHub. Tags are used to identify which image to pull from the repository based on project technology.

Navigate to the cp-projects-service/scripts directory and create two new files named “Dockerfile” and “docker-entrypoint” inside of a new directory named after the technology you want to support as follows

```
mkdir technologyName
cd technologyName
touch Dockerfile
touch docker-entrypoint
```

The Dockerfile will set up the image we are going to use for the runtimes for this new technology and can be based on an existing image or built from scratch. The docker-entrypoint shell script will contain the commands that are run within the container when it is spawned. These commands should clone the code for the project being run from the \$GITHUB_URL and then run it using the \$ENTRYPOINT as the main filename. These environment variables will be made available to the docker-entrypoint script when the process to spawn the container using the run-project script is executed by cp-projects-service. Examples of these files can be found in the cp-projects-service/scripts directory for each technology I am currently supporting.

The docker-entrypoint file must be made executable as follows

```
chmod +x docker-entrypoint
```

Once these files have been written to suit the technology you want to support you will need to build the Docker image and push it to DockerHub to be used by cp-projects-service. Again, replace technologyName with the name of the technology you are adding support for which will act as a tag to identify this image and run the following

```
docker login
docker build -t coderdojo/project-runtime:technologyName .
docker push coderdojo/project-runtime:technologyName
```

Once this is complete the image will be available to be used by cp-projects-service! The code then needs to be updated to support a new project type on the frontend and backend and also to recognise a new tag when spawning a container to run a project so that it can pull the new image. That's it!

Screenshots

These screenshots of my system are provided as reference to help in following the instructions in this manual.

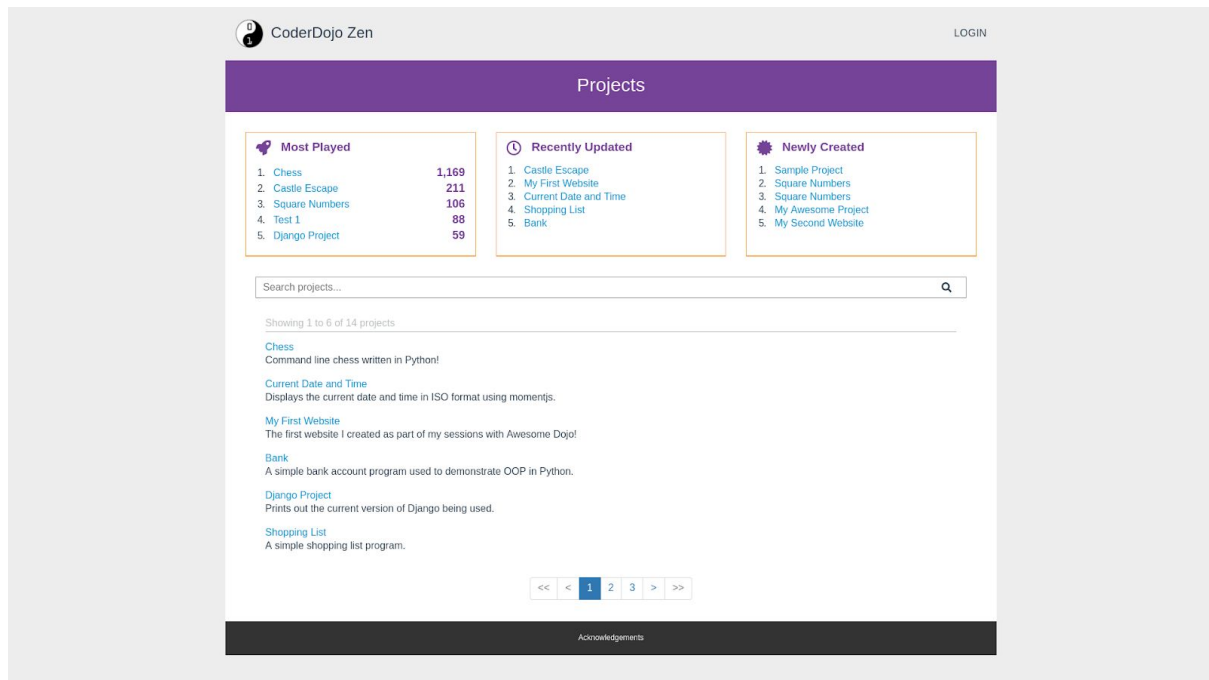


Fig. 1.0. (Project List)

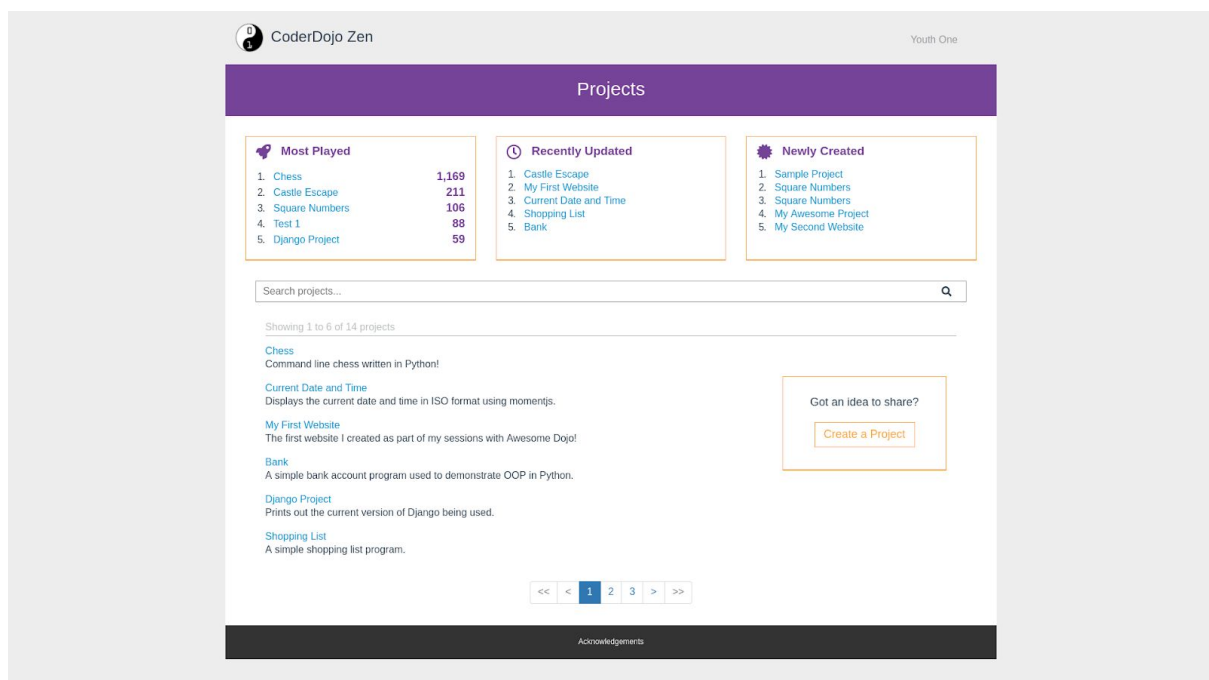


Fig. 1.1. (Project List, logged in as a youth)

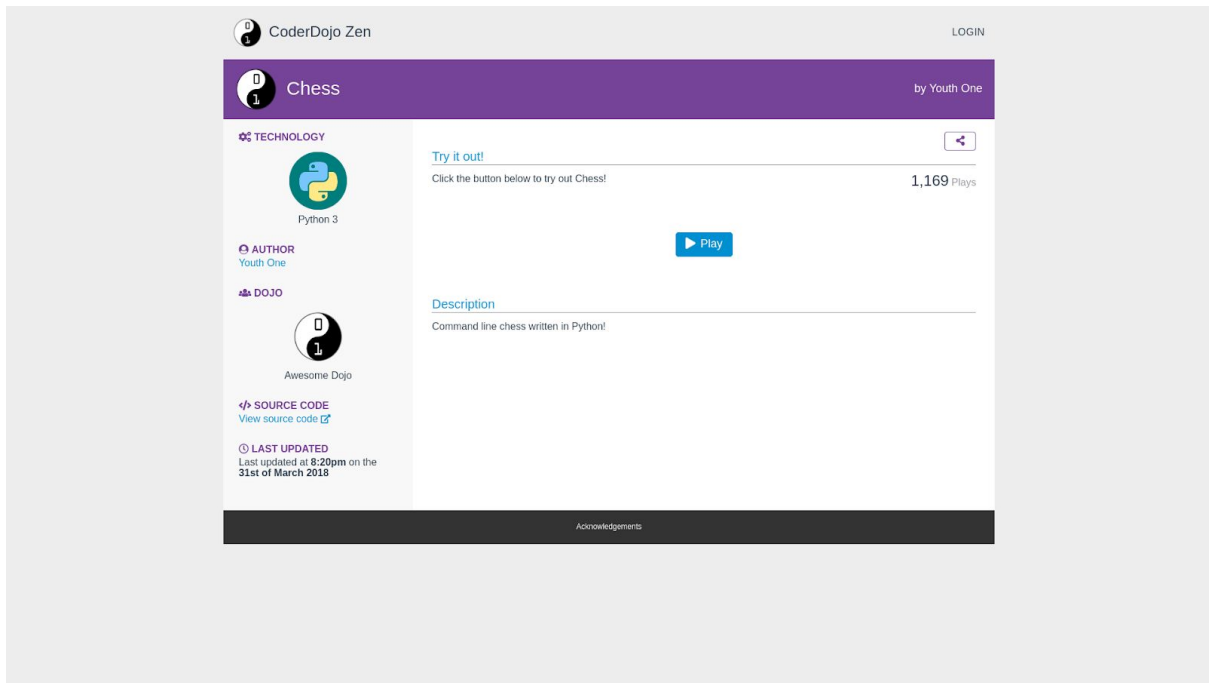


Fig. 2.0. (Project Page)

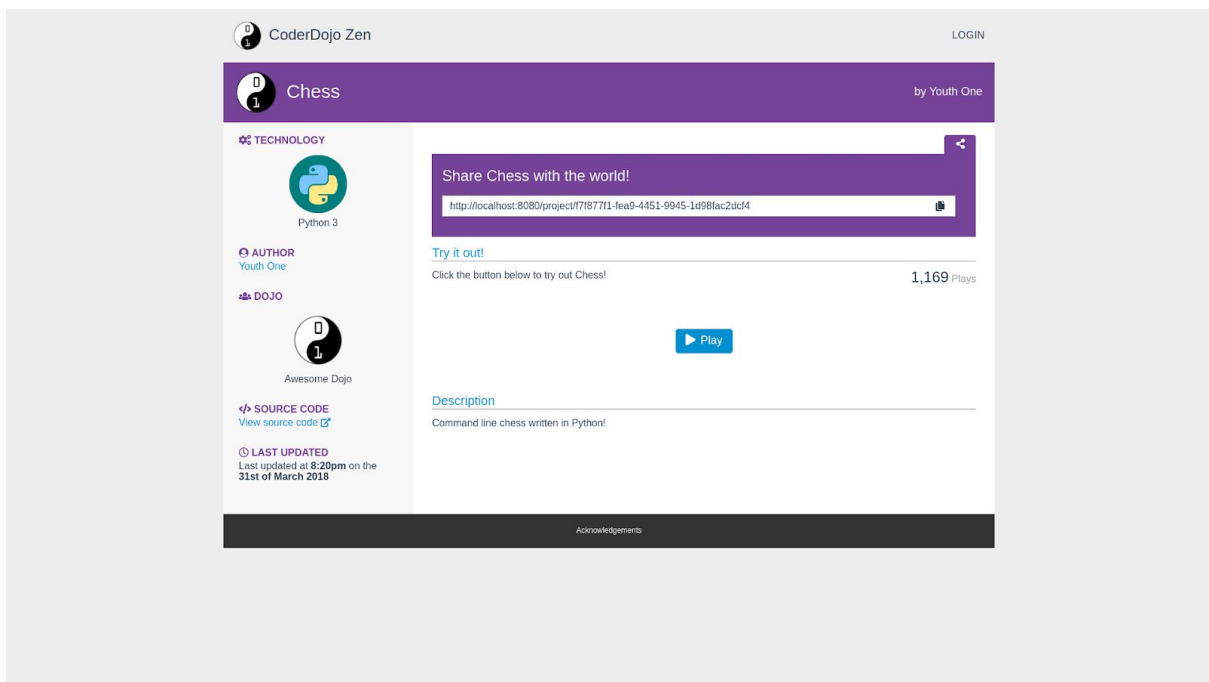


Fig. 2.1. (Project Page, share interface open)

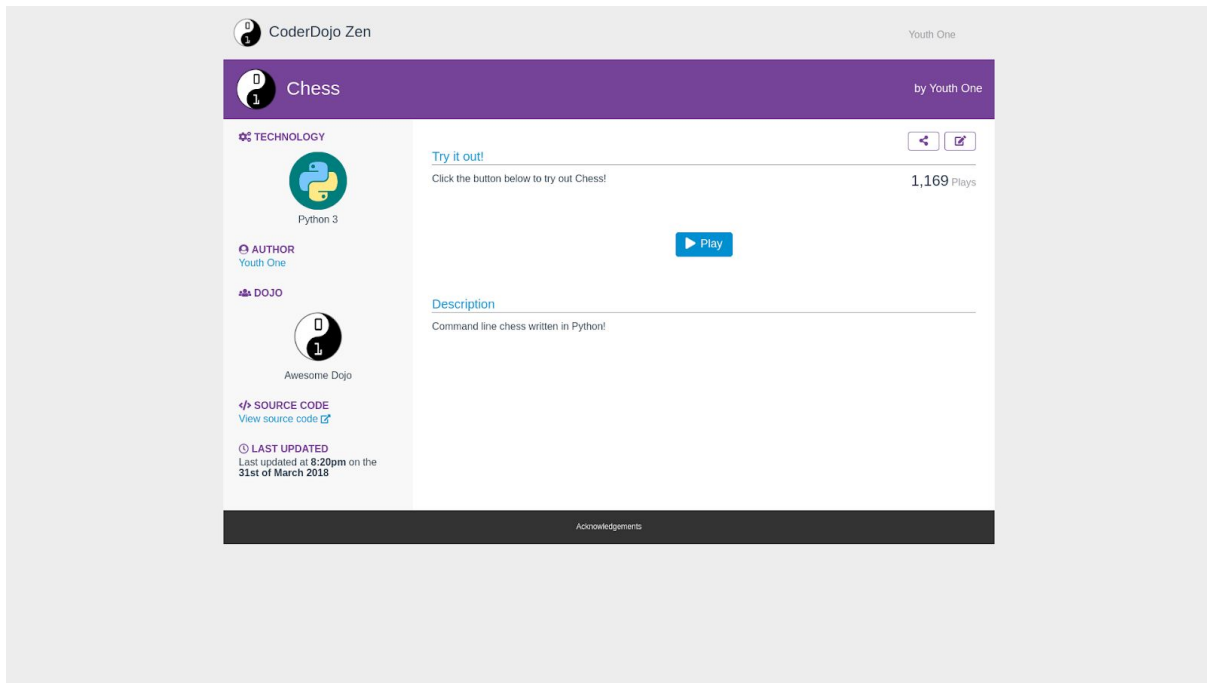


Fig. 2.2. (Project Page, logged in as project owner)

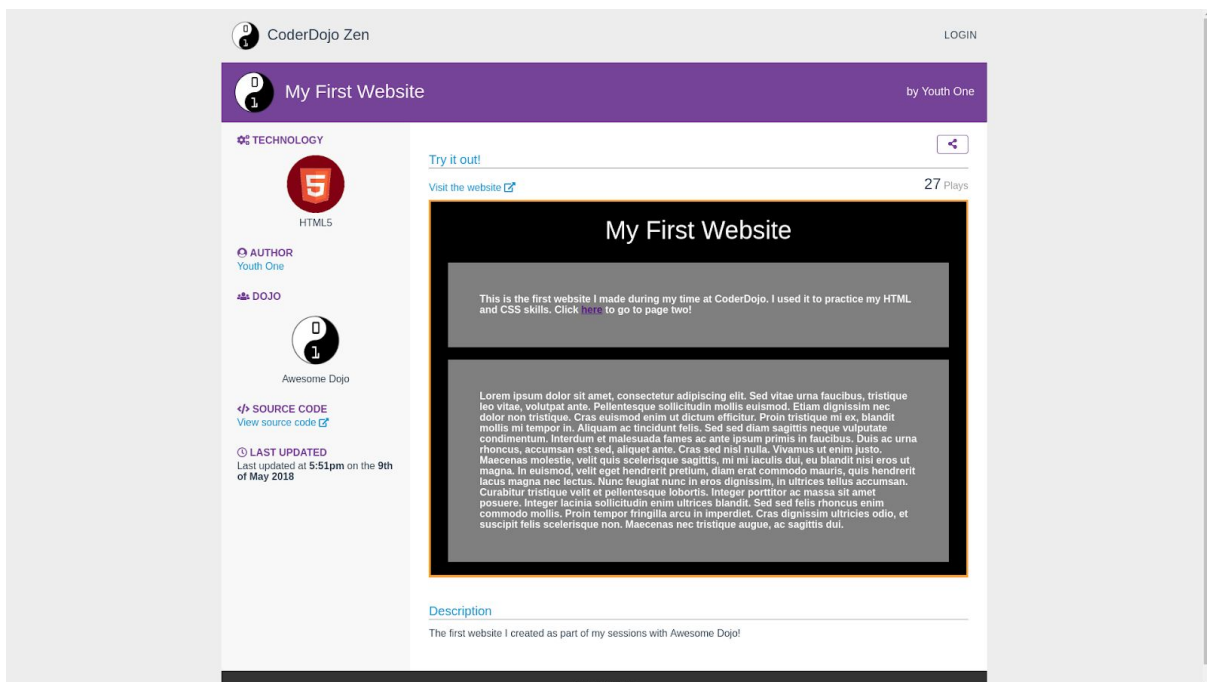


Fig. 2.3. (Project Page, HTML5 project)

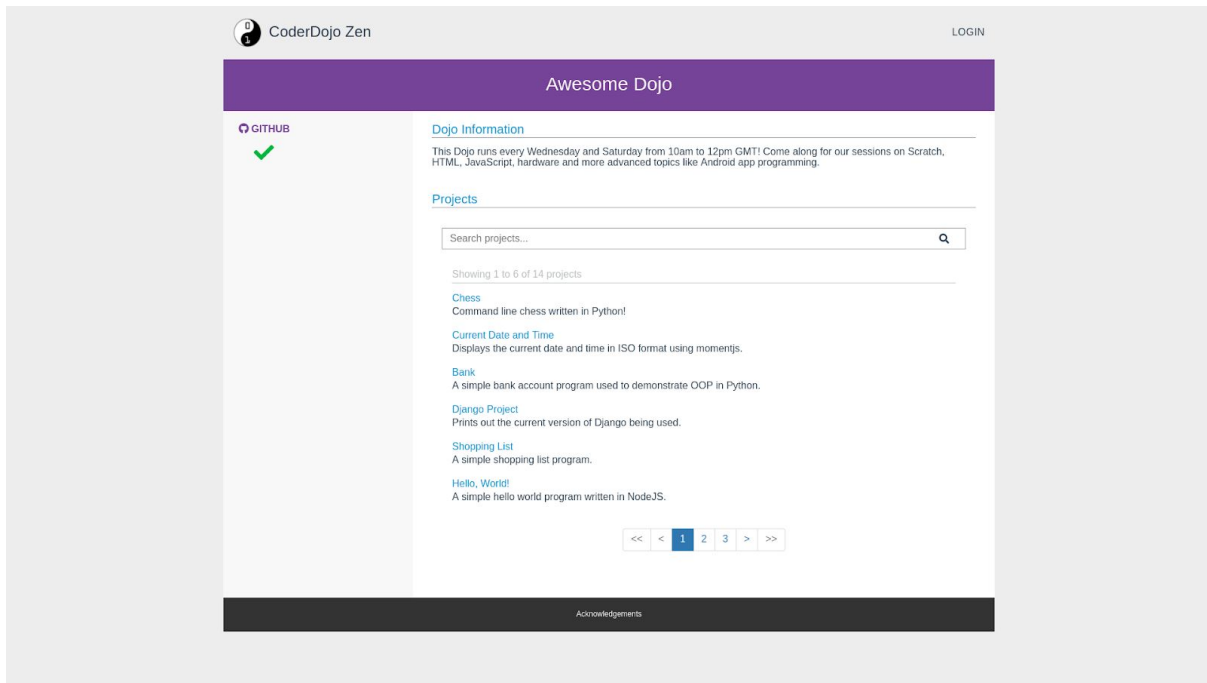


Fig. 3.0. (Dojo Details, GitHub integrated)

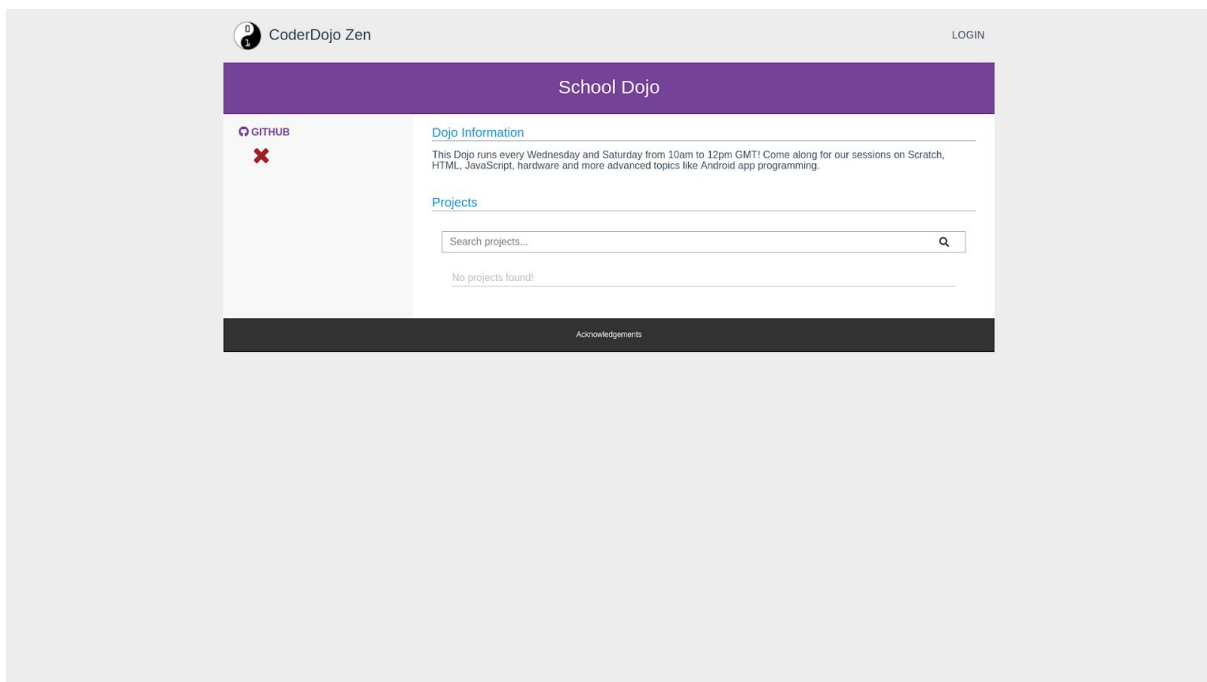


Fig. 3.1. (Dojo Details, GitHub not integrated)

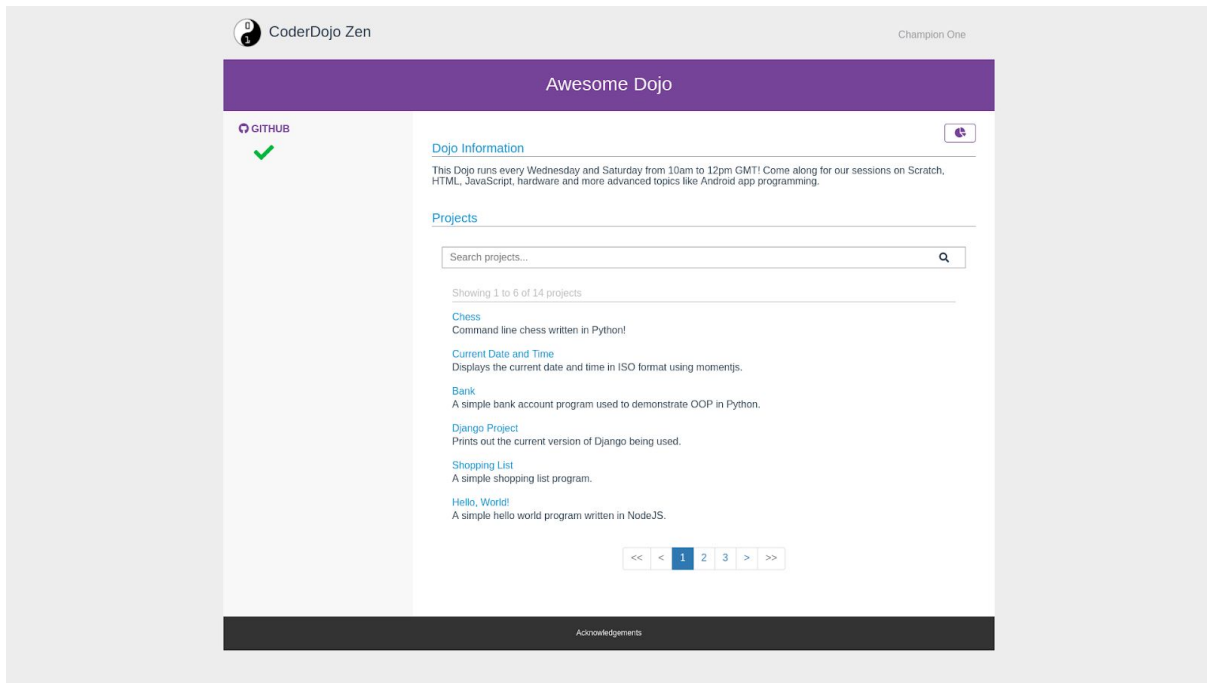


Fig. 3.2. (Dojo Details, GitHub integrated and logged in as Champion)

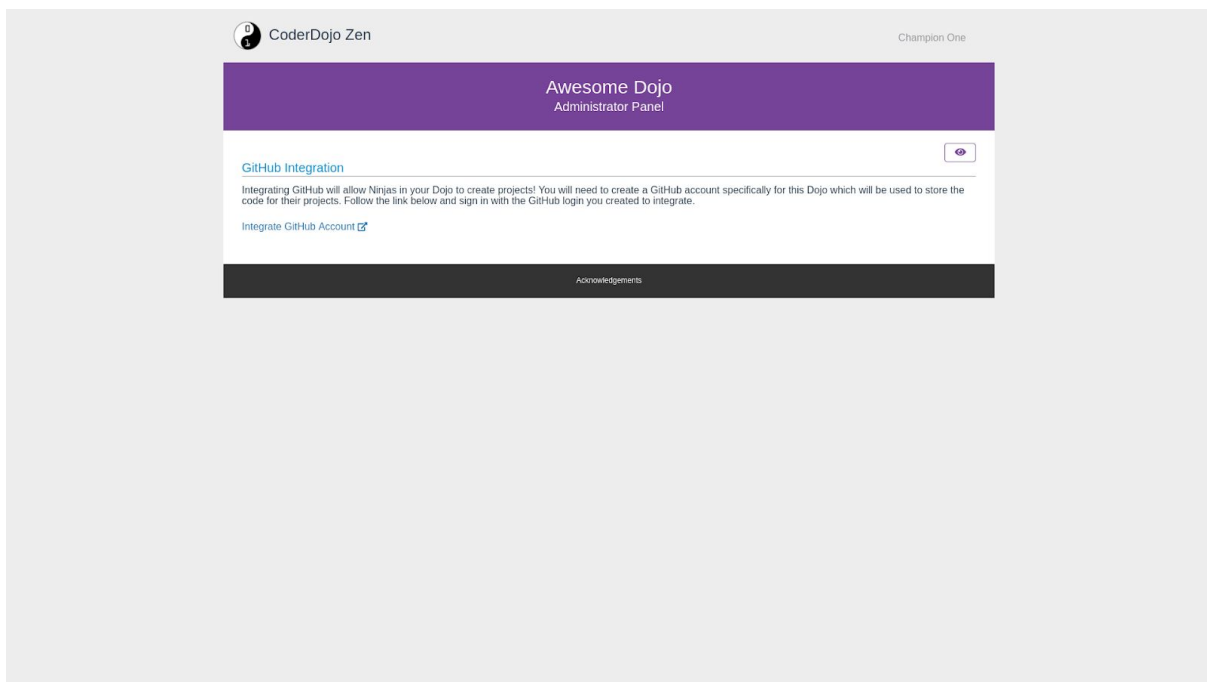


Fig. 4.0. (Administrator Panel, GitHub not integrated)

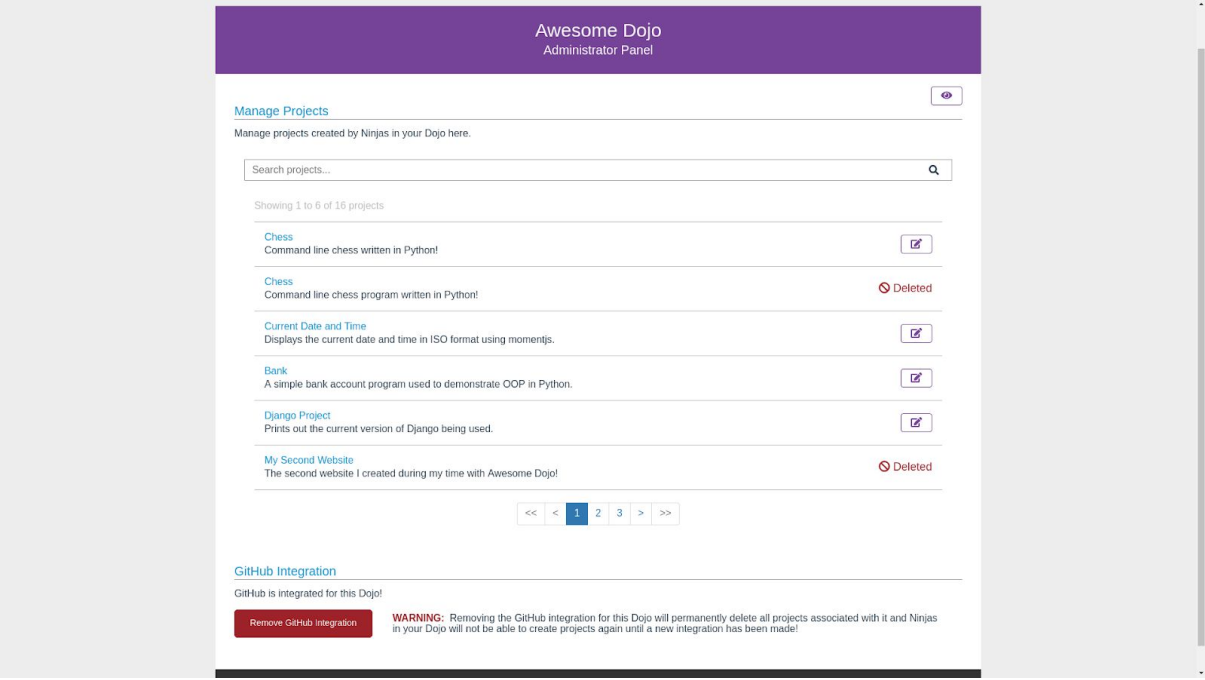


Fig. 4.1. (Administrator Panel, GitHub integrated)

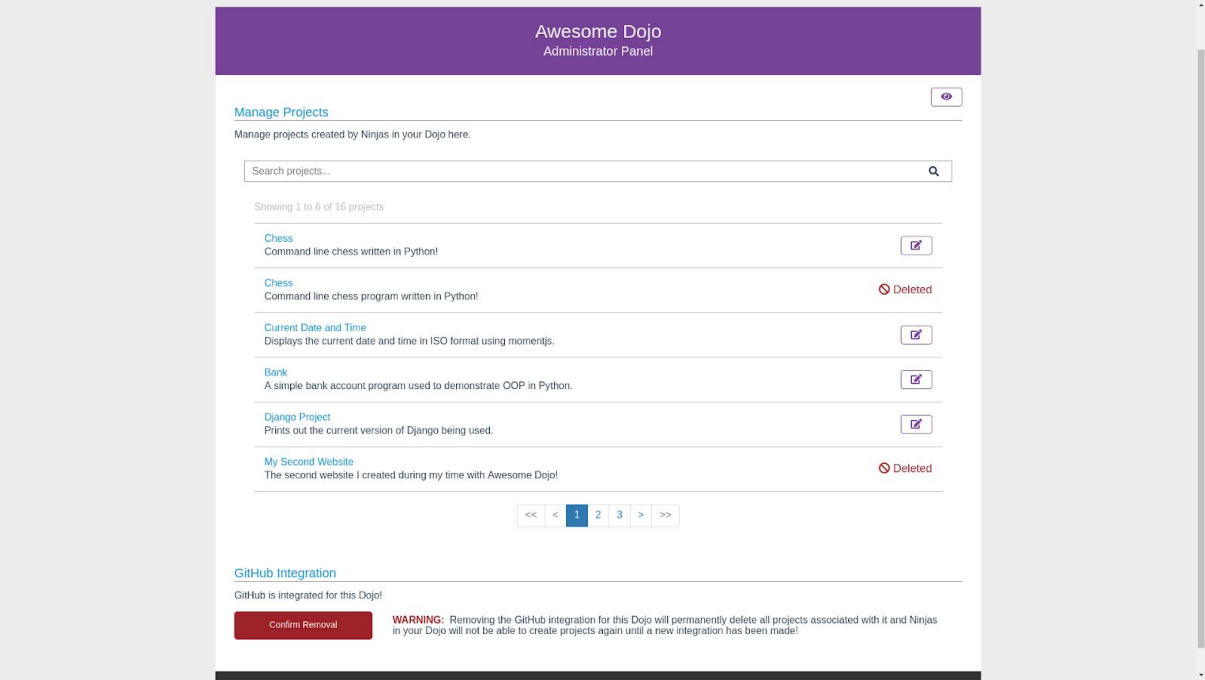


Fig. 4.1. (Administrator Panel, GitHub integrated and confirm button displayed)

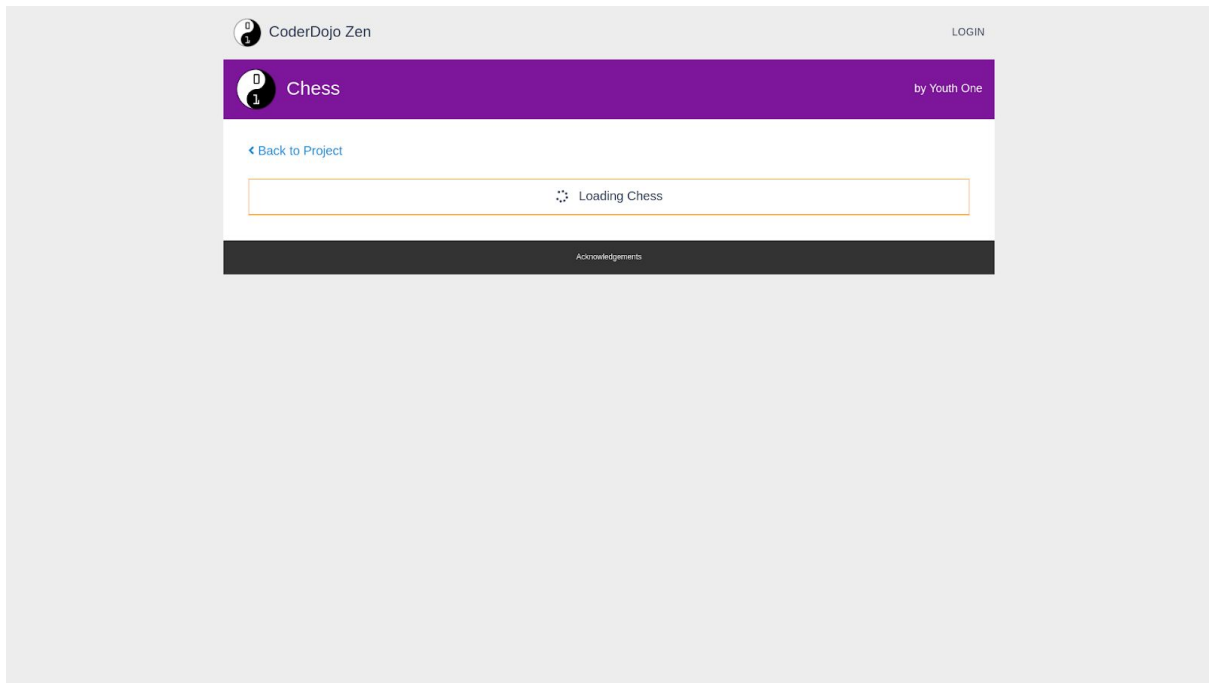


Fig. 5.0. (Project Runtime, loading message)

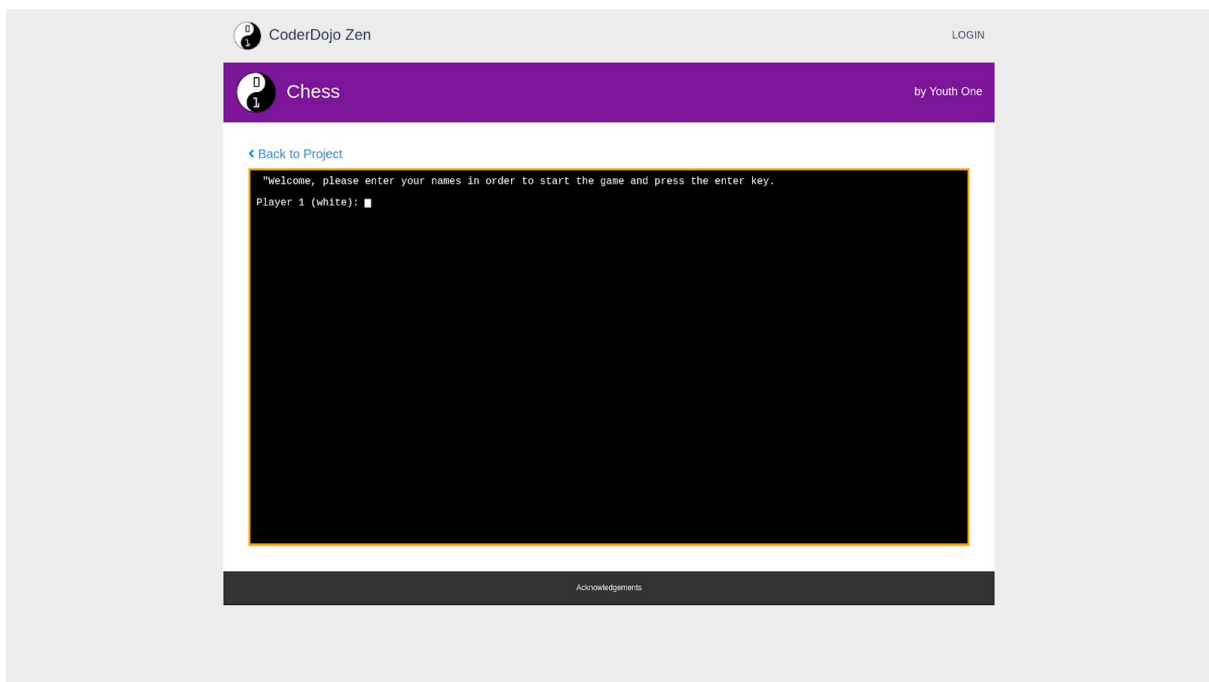


Fig. 5.1. (Project Runtime, terminal)

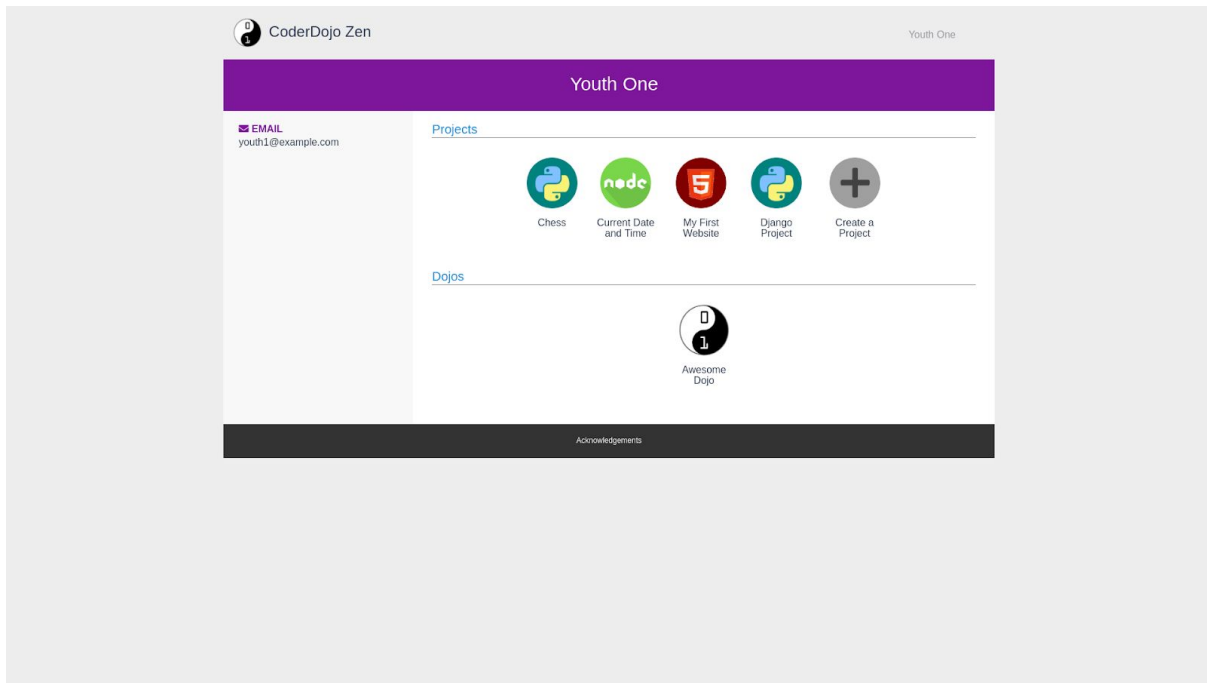


Fig. 6.0. (Profile Page, logged in as a youth)

The screenshot shows the 'New Project' form in CoderDojo Zen. The form is titled 'New Project' and has a purple header. Below the header, there is a section for 'Project Information' with the following fields: 'Project Name' (e.g. My First Project), 'Project Type' (Python 3, NodeJS, HTML5, Java), 'Project Description' (e.g. This is the first project I created on Zen), 'Dojo' (Awesome Dojo), and 'Project Resource' (e.g. kaita.coderdojo.com). Below this is a section for 'Project Files' with the instruction 'Upload the code for your project here!'. It includes a field for 'Project main filename' (Main filename (without extension)) and a button to upload a zip file. At the bottom, there are two informational icons and a 'Create Project' button. The footer of the page is a dark grey bar labeled 'Acknowledgements'.

Fig. 7.0. (Project Creation Form, logged in as a youth)