

Step 1: Set Up Your Project Directory and Files

1. Create a New Folder

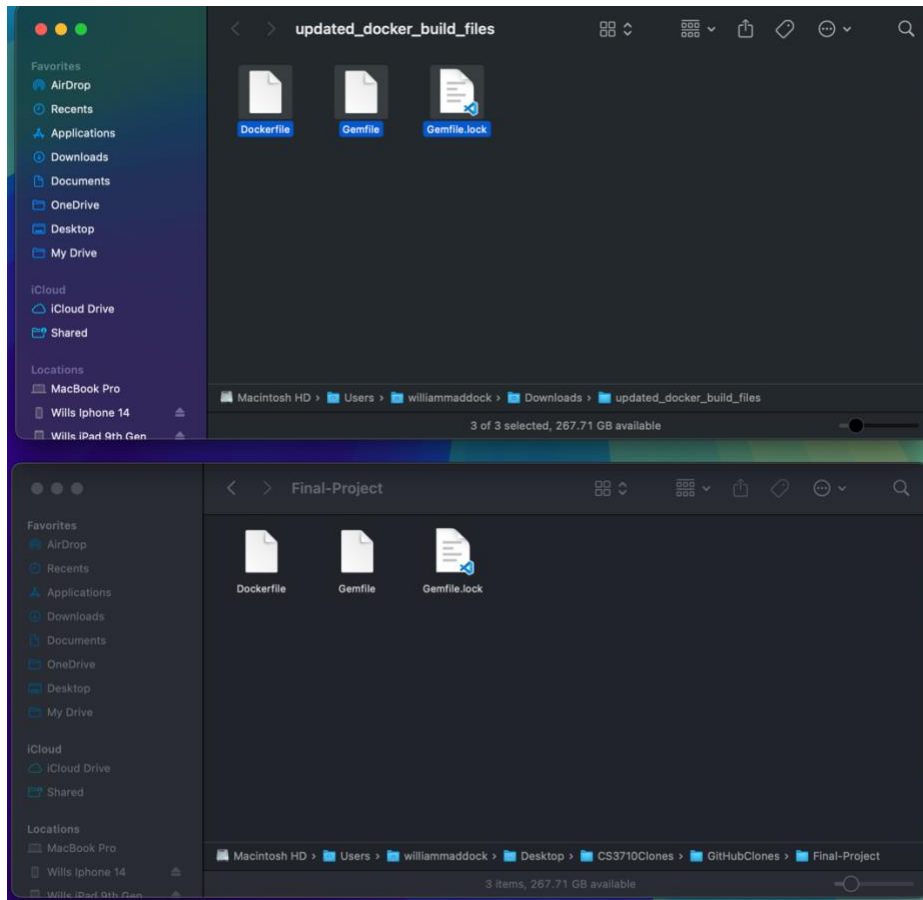
Create a new folder and name it something meaningful (e.g., Final-Project).

2. Download Updated Docker Build Files

If you haven't already downloaded the updated Docker build files, you can do so [here](#).

3. Place Docker Files in Directory

After downloading, move the Updated Docker files into your project directory (e.g., /Users/williammaddock/Desktop/CS3710Clones/GitHubClones/Final-Project).

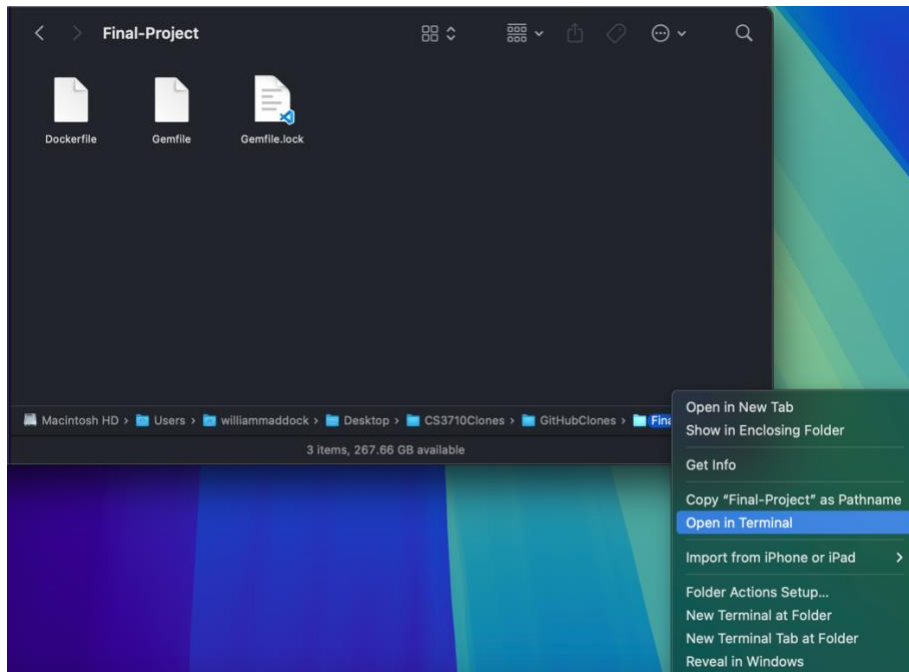


Step 2: Build and Run Your Docker Image

1. Navigate to Project Directory

Open your terminal and navigate to your project directory:

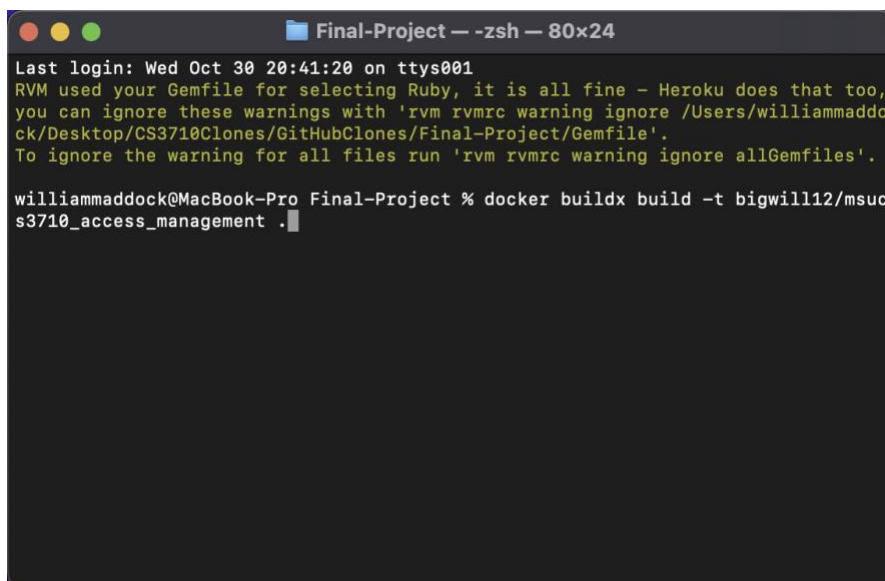
```
cd /Users/williammaddock/Desktop/CS3710Clones/GitHubClones/Final-Project
```



2. Build Docker Image

Run the following command to build your Docker image, replacing your `_github_login` and `name_your_app_here` with your GitHub username and app name:

```
docker buildx build -t bigwill12/msucs3710_name_it_whatever .
```



```
Final-Project - zsh - 80x24
Last login: Wed Oct 30 20:41:20 on ttys001
RVM used your Gemfile for selecting Ruby, it is all fine - Heroku does that too,
you can ignore these warnings with 'rvm rvmrc warning ignore /Users/williammaddock/Desktop/CS3710Clones/GitHubClones/Final-Project/Gemfile'.
To ignore the warning for all files run 'rvm rvmrc warning ignore allGemfiles'.

williammaddock@MacBook-Pro Final-Project % docker buildx build -t bigwill12/msucs3710_access_management .
```

2. Run Docker Container

Start the Docker container:

```
docker run -it -p 3000:3000 -v "$(pwd):/workspace" bigwill12/msucs3710_name_it_whatever
```

```
Final-Project — docker run -it -p 3000:3000 -v ~/Desktop/CS3710Clones...
=> [5/7] COPY Gemfile Gemfile.lock ./ 0.1s
=> [6/7] COPY . /workspace 0.0s
=> [7/7] RUN bundle install 10.0s
=> exporting to image 28.5s
=> => exporting layers 23.7s
=> => exporting manifest sha256:4a27d5e2a5efbf9cf32f5f193bbe05407d0a77fc 0.0s
=> => exporting config sha256:b5844952aa4f0549c88dc7d020238cbeb988a65bad 0.0s
=> => exporting attestation manifest sha256:c9f3b442e90be19268b5e577701c 0.0s
=> => exporting manifest list sha256:6daede59043397bbf6282499a613f170c82 0.0s
=> => naming to docker.io/bigwill12/msucs3710_access_management:latest 0.0s
=> => unpacking to docker.io/bigwill12/msucs3710_access_management:lates 4.8s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux
/wppil9remtzclcm9pfrin09h3

1 warning found (use docker --debug to expand):
- FromAsCasing: 'as' and 'FROM' keywords' casing do not match (line 5)

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout q
uickview
[williammaddock@MacBook-Pro Final-Project % docker run -it -p 3000:3000 -v $(pwd)]
:/workspace bigwill12/msucs3710_rails7
root@f5235246cabd:/workspace#
```

Step 3: Customize App Name in Rails Setup

1. Update App Name in Rails Command

Following M03 L05 Team Roles and Rails Project (Slide 22), change portfolio_app to your chosen app name.

[Presentation Link](#)

Versions and Rails Commands

```
# root@1d0d1e8d6785:/workspace#
# rails new portfolio_app --skip -bundle
# ls
# portfolio_app
# cd portfolio_app
# bundle install
# ls
# ruby --version
# rails --version
```

Workspace # Interactive terminal for linux environment

Create rails app for portfolio

You must cd into rails project directory Install [gems](#).

Notice all the files that were generated for the rails framework

Resources: Make sure you know what versions for your documentation

- <https://guides.rubyonrails.org/v7.1/>
- [Bundler: Getting Started](#)
- [Ruby & Rails Compatibility Table - FastRuby.io | Rails Upgrade Service](#)

Guides

Version	✓ Edge
7.2	
7.1	

Why is version compatibility important?

2. Run Rails Commands

Use these commands in the terminal:

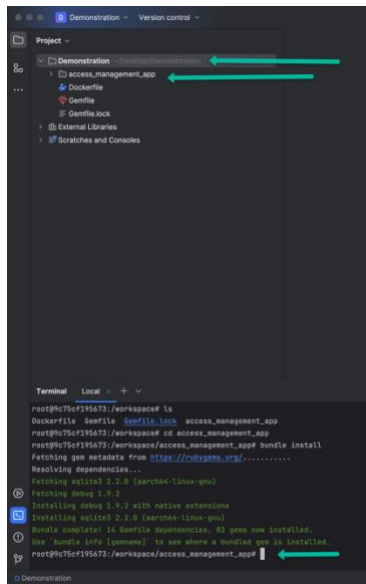
```
rails new access_management_app --skip -bundle
```

 (name it what ever you like)

```
cd access_management_app
```

```
bundle install
```

And now you have a folder in your Final Projects directory.



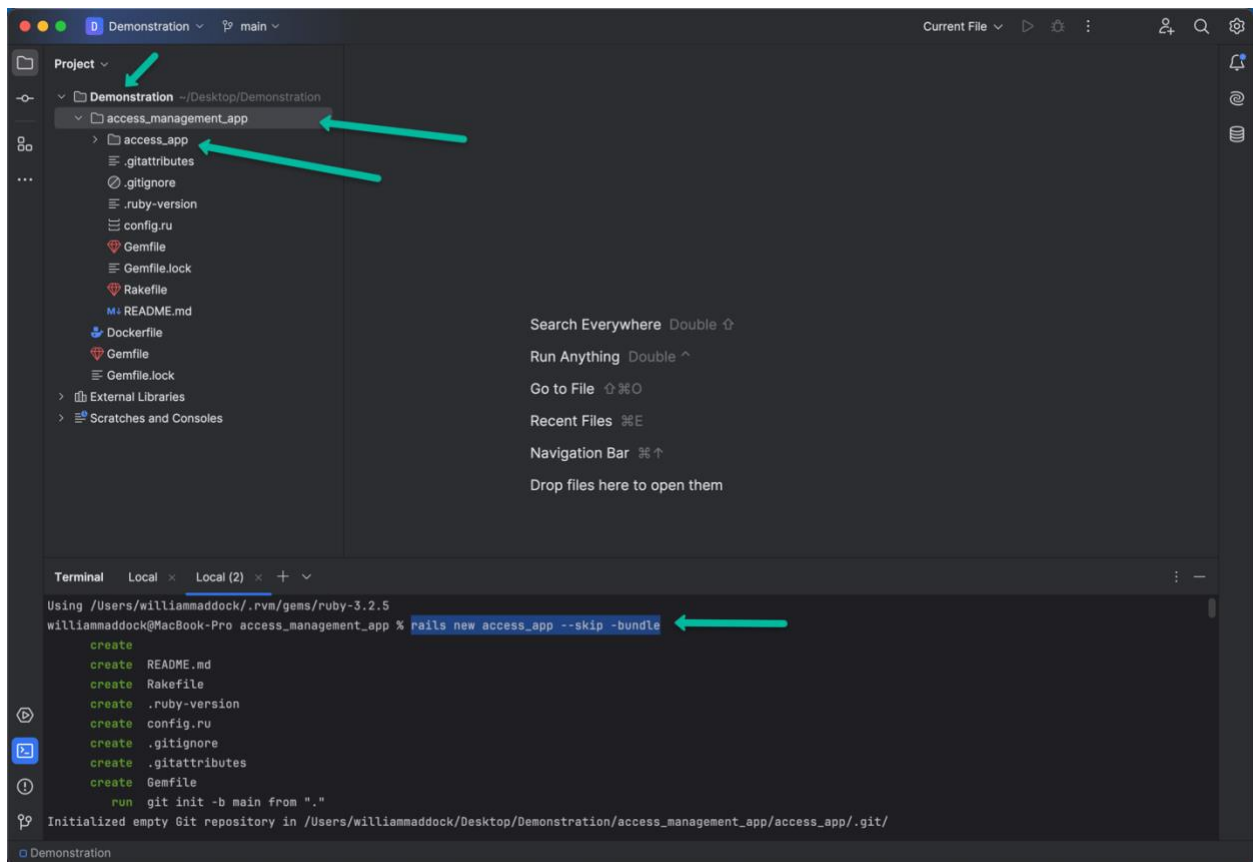
now do it again except name it something else you want to name it:

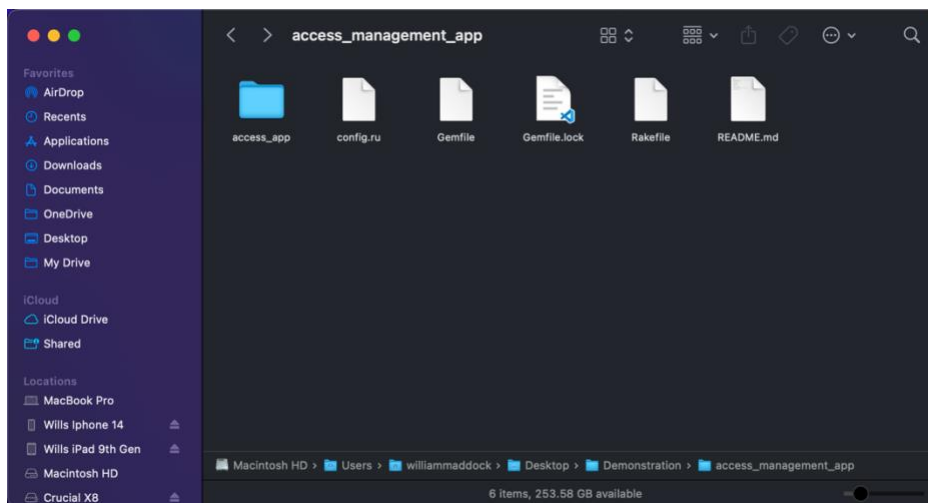
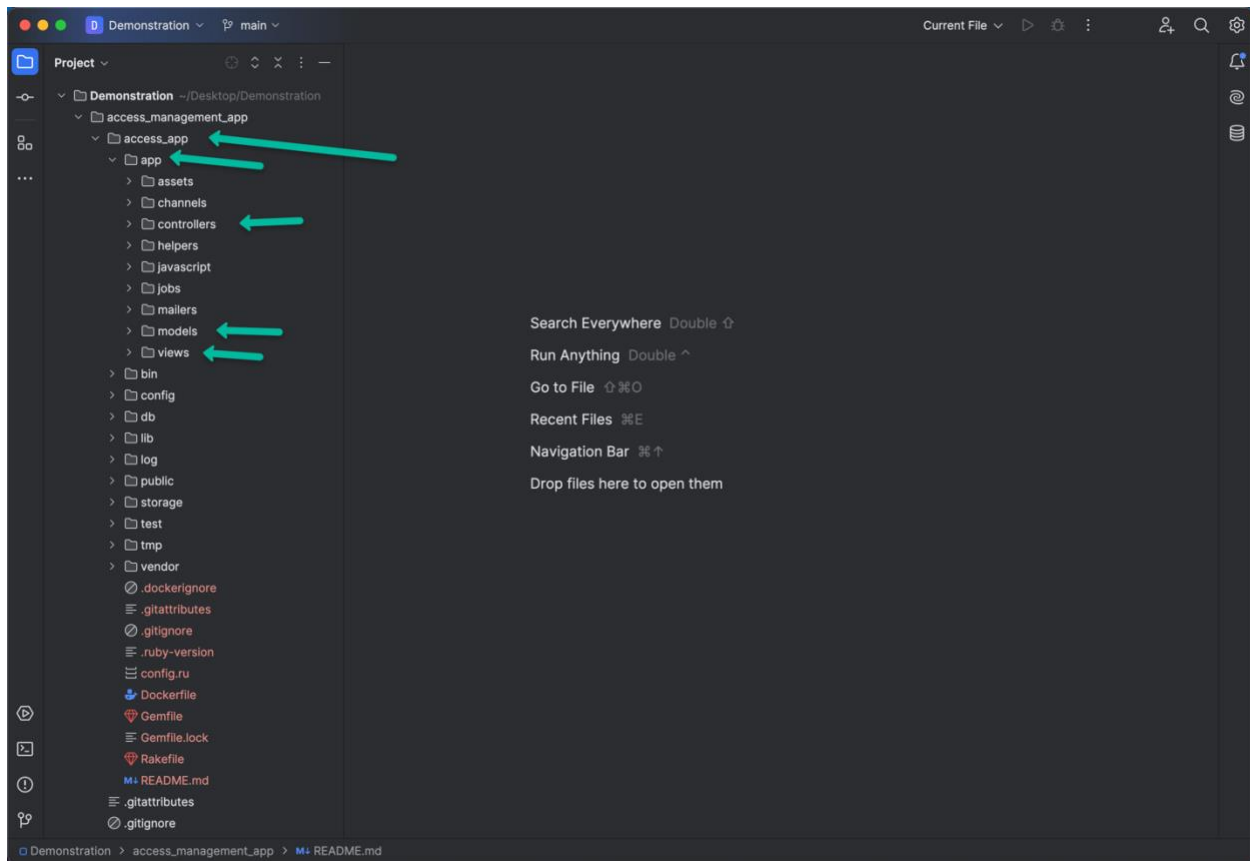
```
rails new access_app --skip -bundle
```

```
cd access_app
```

```
bundle install
```

Now look at the pic below





Step 4: Start Rails Server

1. Install Gems and Start Rails Server

bundle install

rails server -b 0.0.0.0

```
Final-Project — docker run -it -p 3000:3000 -v ~/Desktop/CS3710Clones...
Pin all controllers
Appending: pin_all_from "app/javascript/controllers", under: "controllers"
  append    config/importmap.rb
  run      bundle install
Bundle complete! 14 Gemfile dependencies, 84 gems now installed.
Use `bundle info [gemname]` to see where a bundled gem is installed.
[root@f5235246cabd:/workspace# ls
Dockerfile  Gemfile  Gemfile.lock  access_management_app
[root@f5235246cabd:/workspace# cd access_management_app
[root@f5235246cabd:/workspace/access_management_app# bundle install
Bundle complete! 14 Gemfile dependencies, 84 gems now installed.
Use `bundle info [gemname]` to see where a bundled gem is installed.
[root@f5235246cabd:/workspace/access_management_app# rails server -b 0.0.0.0
=> Booting Puma
=> Rails 7.1.5 application starting in development
=> Run `bin/rails server --help` for more startup options
Puma starting in single mode...
* Puma version: 6.4.3 (ruby 3.2.5-p208) ("The Eagle of Durango")
* Min threads: 5
* Max threads: 5
* Environment: development
* PID: 349
* Listening on http://0.0.0.0:3000
Use Ctrl-C to stop
```

Start Server

Start the server. Command must be run from rails portfolio_app directory. When you want to stop the server type ctrl + c to stop server

```
# rails server -b 0.0.0.0
=> Booting Puma
=> Rails 7.1.3.4 application starting in development
=> Run `bin/rails server --help` for more startup options
```

Docker go to where the host is mapped to port 3000 <http://localhost:3000/>



Step 5-8: Finalizing Project Setup

Create Github Repository

Add remote github repository.

Use the following settings to be able to connect the portfolio_app code base you created in the container.

- Public
- No README file
- No .gitignore

Create and read the ...or create a new repository on the command line. Go to next slide to complete connecting your portfolio_app to this remote repository.

The screenshot shows the GitHub 'Create a new repository' form. The 'Owner' is set to 'debmhteach' and the 'Repository name' is 'msu-3710-class-fall'. A green checkmark indicates 'msu-3710-class-fall is available'. The 'Description' field is empty. The 'Public' option is selected, with a green box and arrow pointing to it. The 'Initialize this repository with:' section has 'Add a README file' unchecked, with a green box and arrow pointing to it. The 'Add .gitignore' section has '.gitignore template: None' selected, with a green box and arrow pointing to it. The 'Choose a license' section has 'License: None' selected. A footer note states 'You are creating a public repository in your personal account.' The page number '25' is in the bottom right corner.

Step 6(optional and not sure if you need this):

Gitignore

When you show your files in the finder you will see the following. If you do not see the .git information you can reveal hidden files.

- Mac: To reveal hidden files in Finder, go to Go > Computer > Macintosh HD and press Shift + Command + . (period).
- Windows: [View hidden files and folders in Windows - Microsoft Support](#)

Put this [.gitignore](#) file in portfolio app directory.

here is the link https://drive.google.com/file/d/1ieOnXT7UqKlprcBOU_fiKeC7Q22i0zfG/view

Step 7:

https://docs.google.com/presentation/d/1AzXk4v6y0Sy4xz-uAHHStU3RPbhBOaY3si5AF5Xlz_E/edit#slide=id.g2855ae74598_0_669

Branch

Tips:

- Do not delete a branch for this class if it contains work
- Create branches from main branch**
- Make sure you are on the correct branch
- Check status or branch often
- Commit and push often

Make sure you are in portfolio_app directory

Add comment to .gitignore file

Go to repos

```
git status
On branch main
git branch branchTest
git log
git status
git checkout branchTest
git status
git add .
git commit -m 'test branch change'
git push --set-upstream origin branchTest
git checkout main
git status
git merge branchTest
git status
git push
```

Create branch

Set as working branch

Integrate branch changes with main

Branches

Overview Yours Active State All

Search branches...

Default

Branch

main

Your branches

branchTest

.gitignore test branch change

Dockerfile Initial commit of Rails app

Gemfile Initial commit of Rails app

30

Step 8:

Continue to M03 L05 Team Roles and Rails Project Slide 18:

https://docs.google.com/presentation/d/1b_GX_Hojo61oTCsl0kkHXqu0tQYpE8QGSAuN9-QJ_EA/edit#slide=id.g2f1a7c4d69f_1_769

Rails Scaffold

Issues with github codespaces - if not using docker [Local Rails Set Up](#)

For this lecture you will see how to create the following Student model.

- Access the container you create last lecture.
- Make sure you are in the Linux container command
- Open folder in your IDE such as VS Code or Codium
- Create a branch ge03models from main branch.
- Checkout the branch so you can commit and push to the branch.
- Run the scaffold command and then you will need to migrate the database. This will be discussed in more detail later.

Student

* required

```
name:string *
school_email:string * unique
major:string *
Minor:string
graduation_date: date
```

```
#rails generate scaffold Student name:string school_email:string major:string minor:string graduation_date:date
#rails db:migrate
```

Only use your new Scaffolding based on your UML for your final project.

This concludes the foundation before the scaffolding and has created a GitHub repository main branch for your final project.

Good Luck Everyone and hope this pdf helps!