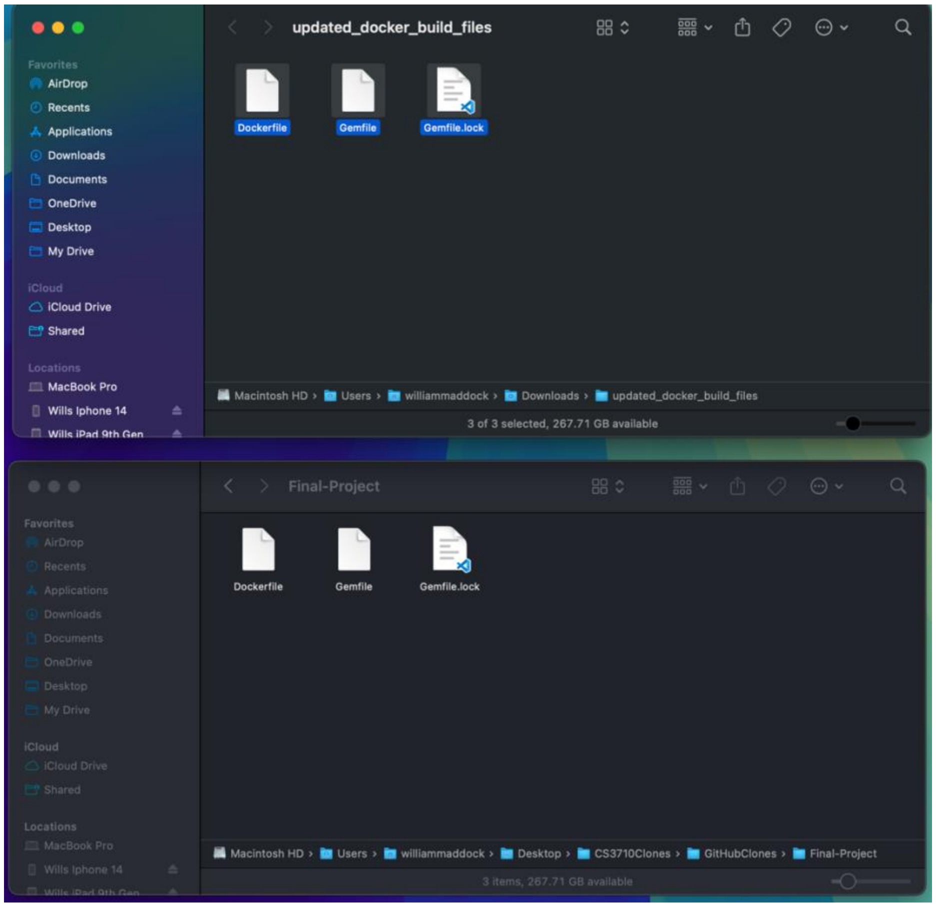
Step 1: Set Up Your Project Directory and Files

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

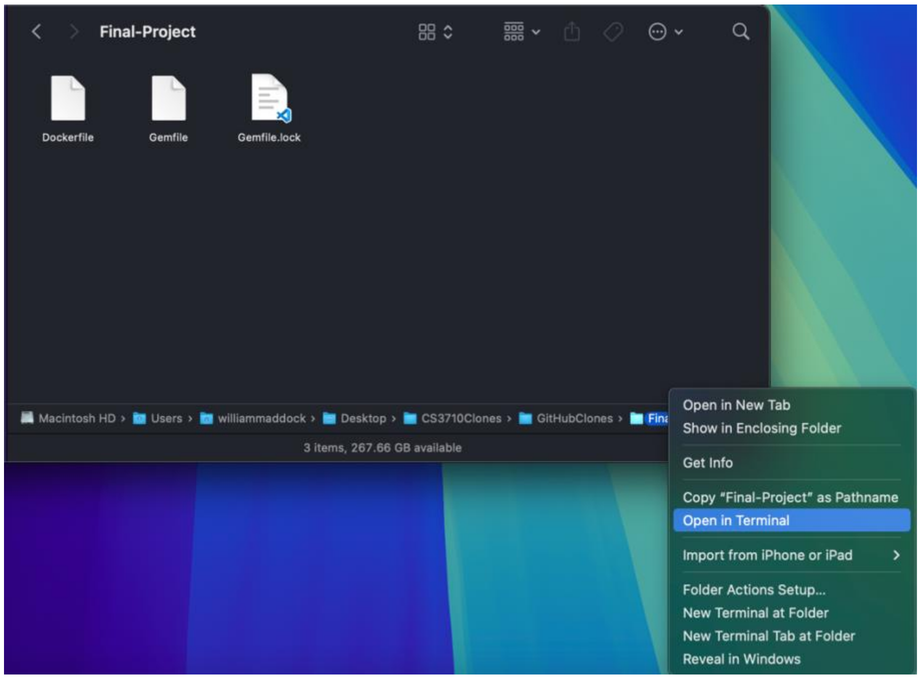
2. **Download Updated Docker Build Files**
If you haven't already downloaded the updated Docker build files, you can do so here.

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).


updated_docker_build_files
Dockerfile
Gemfile
Gemfile.lock
Macintosh HD › Users › williammaddock › Downloads › updated_docker_build_files
3 of 3 selected, 267.71 GB available
Final-Project
Dockerfile
Gemfile
Gemfile.lock
Macintosh HD › Users › williammaddock › Desktop › CS3710Clones › GitHubClones › Final-Project
3 items, 267.71 GB available

Step 2: Build and Run Your Docker Image

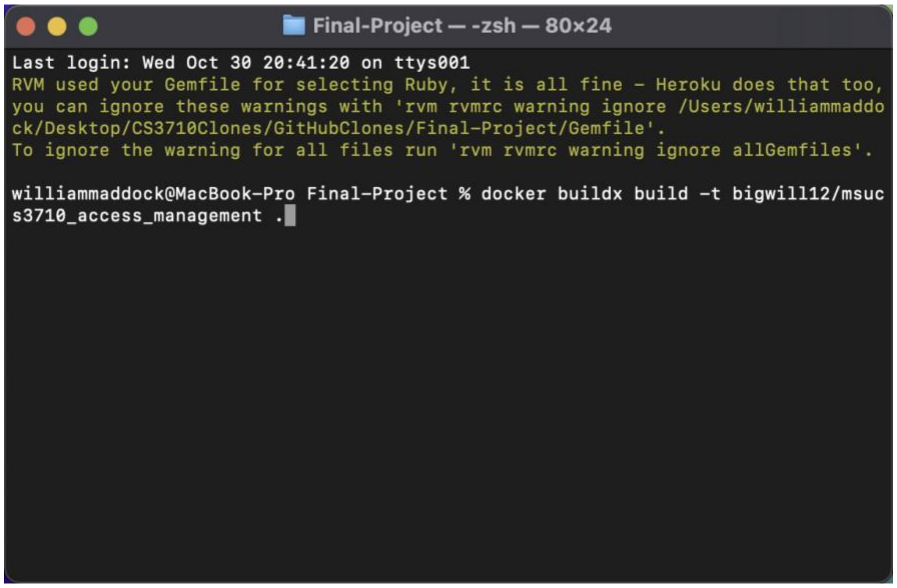
1. Navigate to Project Directory
Open your terminal and navigate to your project directory:
cd /Users/williammaddock/Desktop/CS3710Clones/GitHubClones/Final-Project


Final-Project
Dockerfile
Gemfile
Gemfile.lock
Macintosh HD › Users › williammaddock › Desktop › CS3710Clones › GitHubClones › Fina
3 items, 267.66 GB available
Open in New Tab
Show in Enclosing Folder
Get Info
Copy "Final-Project" as Pathname
Open in Terminal
Import from iPhone or iPad
Folder Actions Setup...
New Terminal at Folder
New Terminal Tab at Folder
Reveal in Windows

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 .

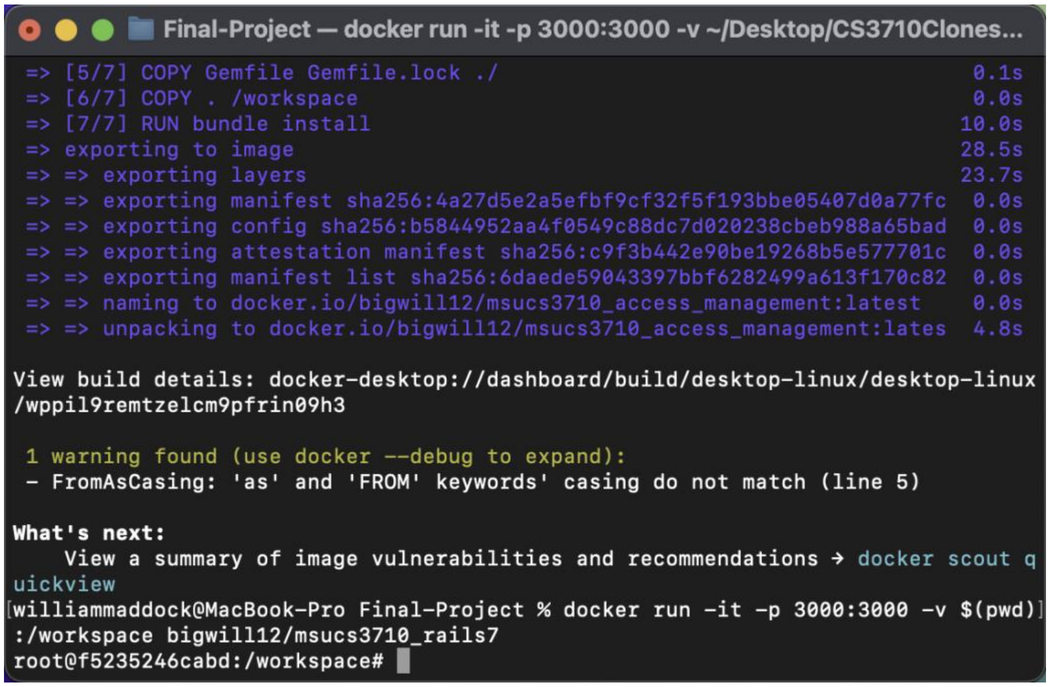


```
Last login: Wed Oct 30 20:41:20 on ttys001
RVM used your Gemfile for selecting Ruby, it is all fine - Heroku does that too,
you can ignore these warnings with 'rvm rvmrc warning ignore /Users/williammaddo
ck/Desktop/CS3710Clones/GitHubClones/Final-Project/Gemfile'.
To ignore the warning for all files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

williammaddock@MacBook-Pro Final-Project % docker buildx build -t bigwill12/msuc
s3710_access_management .
```

2. **Run Docker Container**

Start the Docker container:

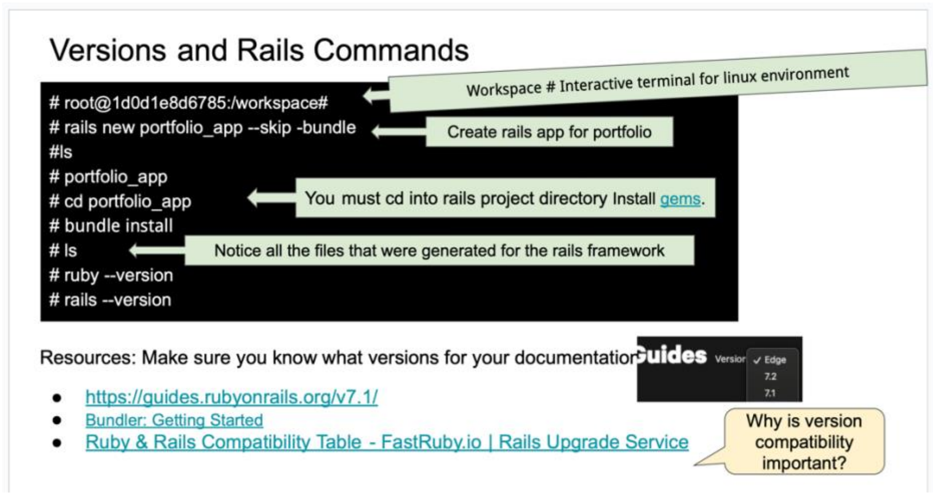
docker run -it -p 3000:3000 -v "$(pwd):/workspace" bigwill12/msucs3710_name_it_whatever



```
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
/wppil9remtzelcm9pfrin09h3

 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
Guides
Version
✓ Edge
7.2
7.1
Why is version compatibility important?

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

2. **Run Rails Commands**

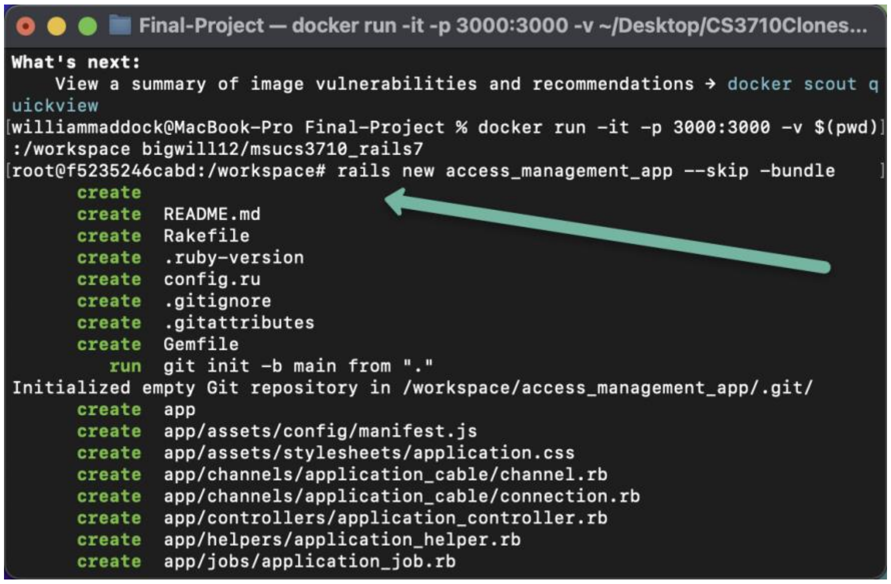
Use these commands in the terminal:

rails new access_management_app --skip -bundle (name it what ever you like)

cd access_management_app

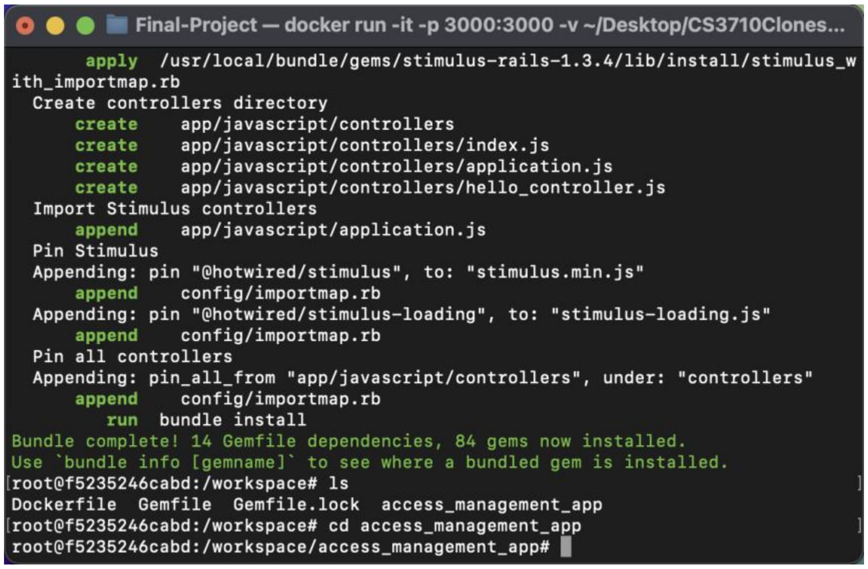
bundle install

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



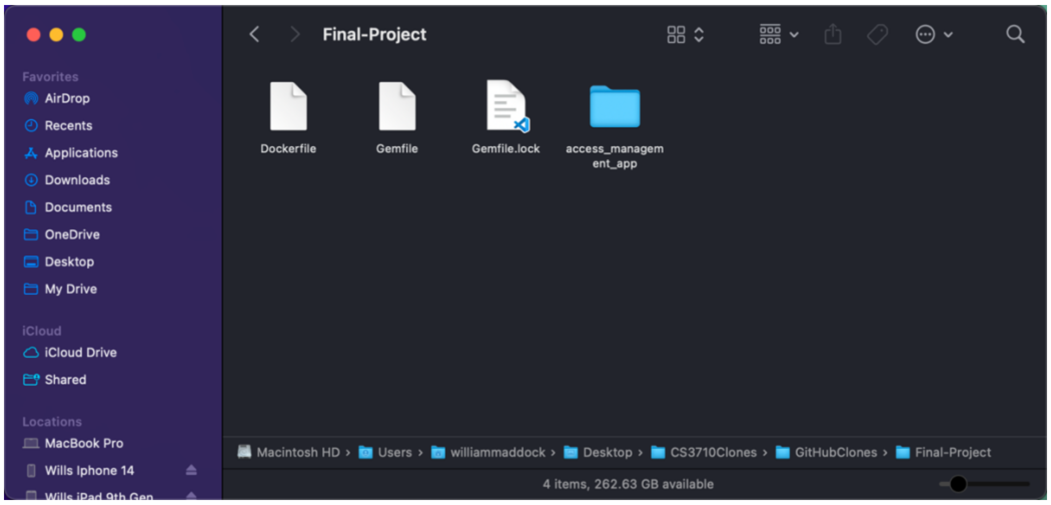
Final-Project — docker run -it -p 3000:3000 -v ~/Desktop/CS3710Clones...

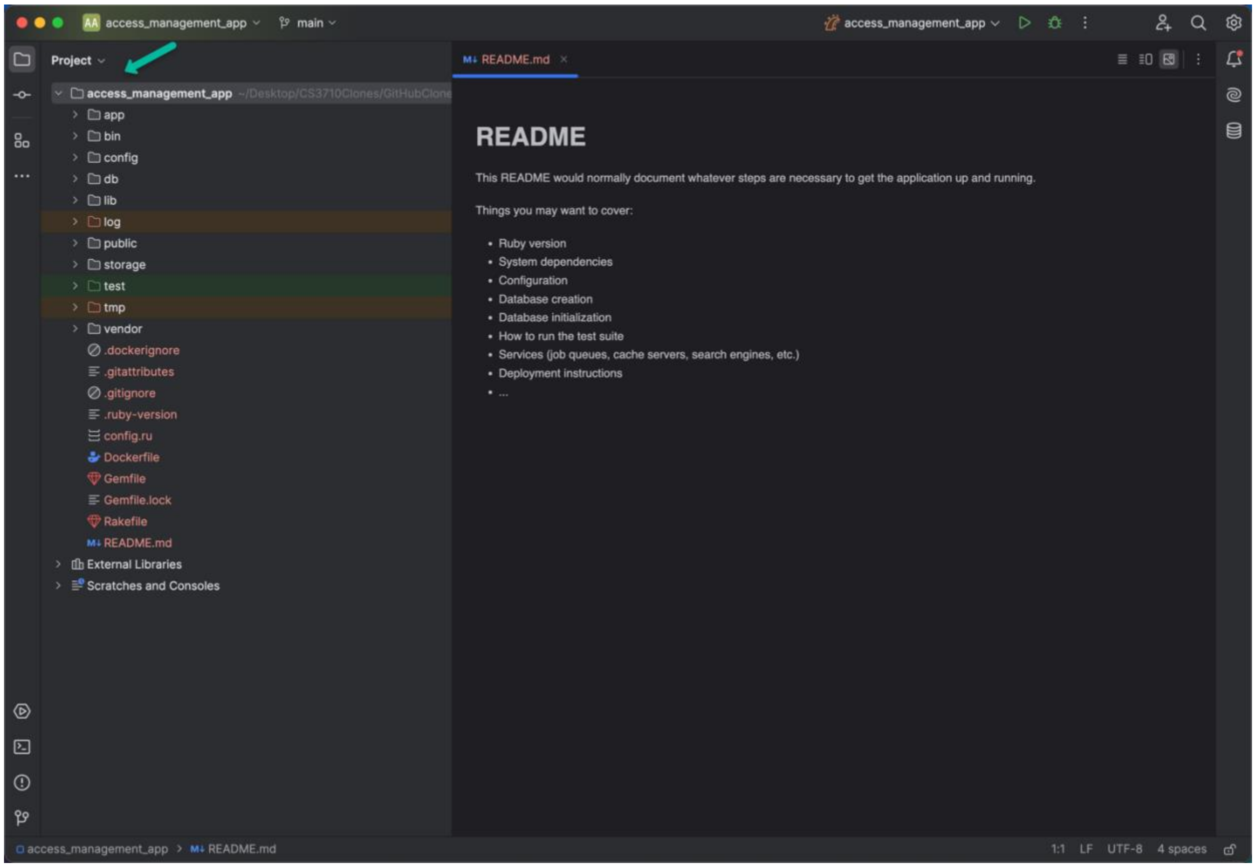
```
What's next:
    View a summary of image vulnerabilities and recommendations → docker scout q
uickview
williammaddock@MacBook-Pro Final-Project % docker run -it -p 3000:3000 -v $(pwd)
:/workspace bigwill12/msucs3710_rails7
root@f5235246cabd:/workspace# rails new access_management_app --skip -bundle
      create
      create  README.md
      create  Rakefile
      create  .ruby-version
      create  config.ru
      create  .gitignore
      create  .gitattributes
      create  Gemfile
         run  git init -b main from "."
Initialized empty Git repository in /workspace/access_management_app/.git/
      create  app
      create  app/assets/config/manifest.js
      create  app/assets/stylesheets/application.css
      create  app/channels/application_cable/channel.rb
      create  app/channels/application_cable/connection.rb
      create  app/controllers/application_controller.rb
      create  app/helpers/application_helper.rb
      create  app/jobs/application_job.rb
```



Final-Project — docker run -it -p 3000:3000 -v ~/Desktop/CS3710Clones...

```
       apply  /usr/local/bundle/gems/stimulus-rails-1.3.4/lib/install/stimulus_w
ith_importmap.rb
  Create controllers directory
      create    app/javascript/controllers
      create    app/javascript/controllers/index.js
      create    app/javascript/controllers/application.js
      create    app/javascript/controllers/hello_controller.js
  Import Stimulus controllers
      append    app/javascript/application.js
  Pin Stimulus
  Appending: pin "@hotwired/stimulus", to: "stimulus.min.js"
      append    config/importmap.rb
  Appending: pin "@hotwired/stimulus-loading", to: "stimulus-loading.js"
      append    config/importmap.rb
  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# 
```





README

This README would normally document whatever steps are necessary to get the application up and running.

Things you may want to cover:

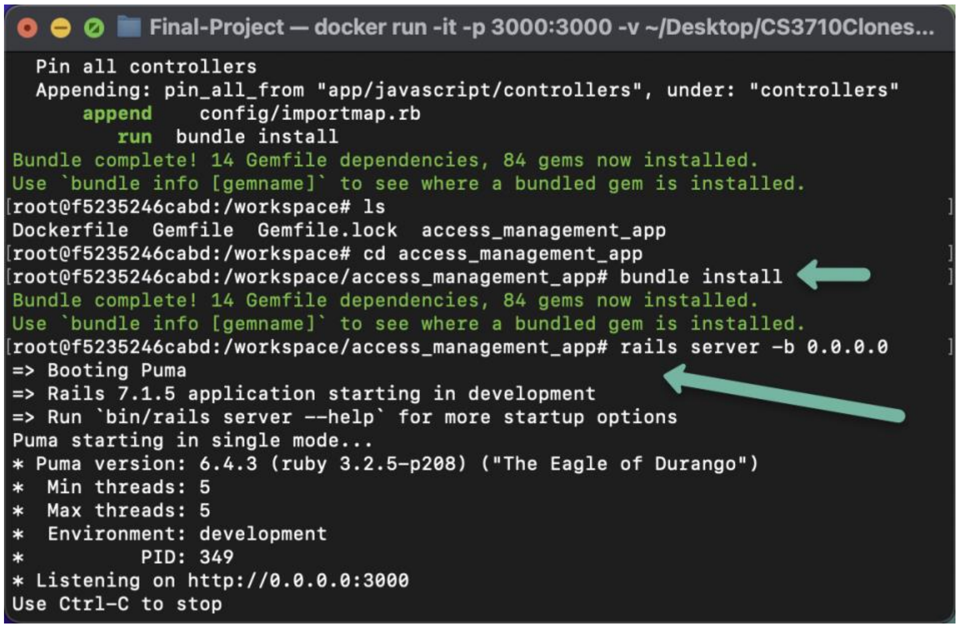
- Ruby version
- System dependencies
- Configuration
- Database creation
- Database initialization
- How to run the test suite
- Services (job queues, cache servers, search engines, etc.)
- Deployment instructions
- ...

Step 4: Start Rails Server

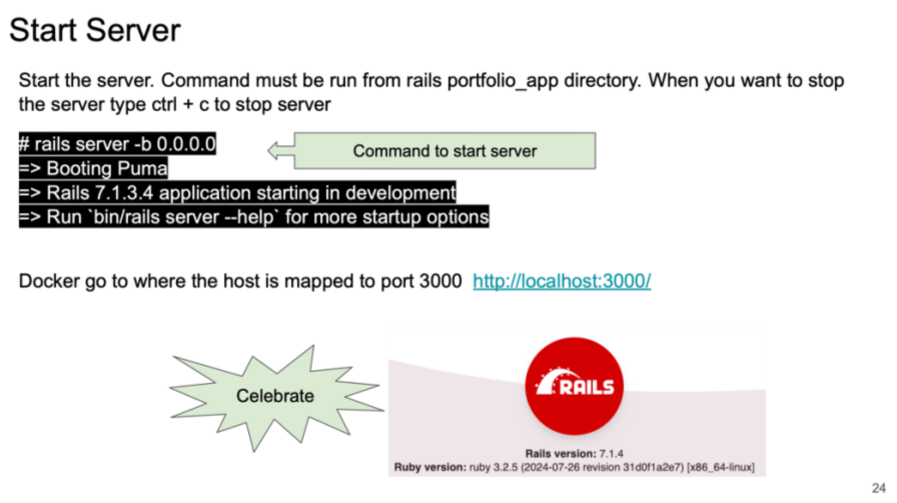
1. **Install Gems and Start Rails Server**

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

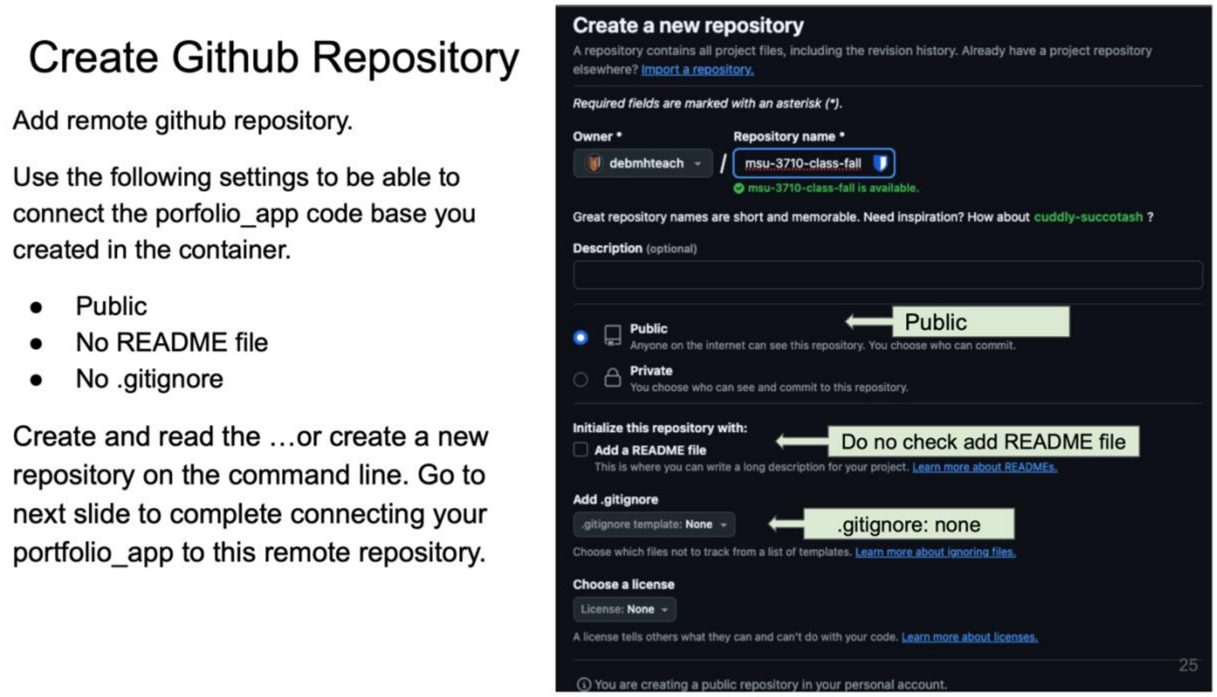
```
# rails server -b 0.0.0.0
=> Booting Puma
=> Rails 7.1.3.4 application starting in development
=> Run `bin/rails server --help` for more startup options
```

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

Celebrate
RAILS
Rails version: 7.1.4
Ruby version: ruby 3.2.5 (2024-07-26 revision 31d0f1a2e7) [x86_64-linux]

24

Step 5-8: Finalizing Project Setup



Create Github Repository

Add remote github repository.

Use the following settings to be able to connect the porfolio_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.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * debmhteach / Repository name * msu-3710-class-fall

msu-3710-class-fall is available.

Great repository names are short and memorable. Need inspiration? How about cuddly-succotash ?

Description (optional)

● Public — Anyone on the internet can see this repository. You choose who can commit. ← Public

○ Private — You choose who can see and commit to this repository.

Initialize this repository with:

☐ Add a README file — This is where you can write a long description for your project. Learn more about READMEs. ← Do no check add README file

Add .gitignore

.gitignore template: None ← .gitignore: none

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

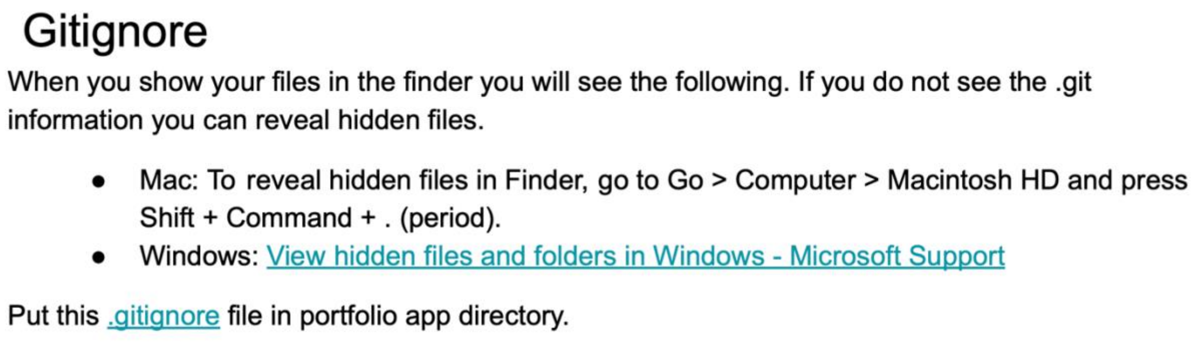
License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

25

You are creating a public repository in your personal account.

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.

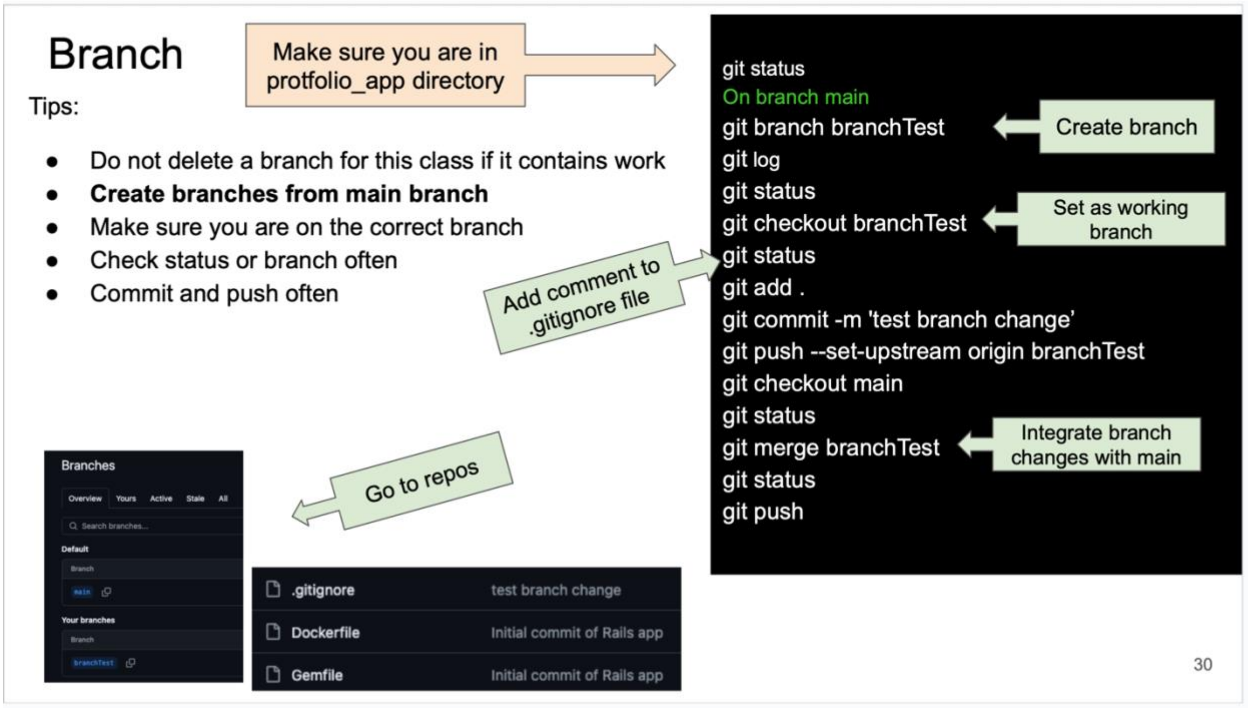
- Mac: To reveal hidden files in Finder, go to Go > Computer > Macintosh HD and press Shift + Command + . (period).
- Windows: [View hidden files and folders in Windows - Microsoft Support]

Put this [.gitignore] file in portfolio app directory.

here is the link https://drive.google.com/file/d/1ieOnXT7UqKIprcBOU_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

```
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
```

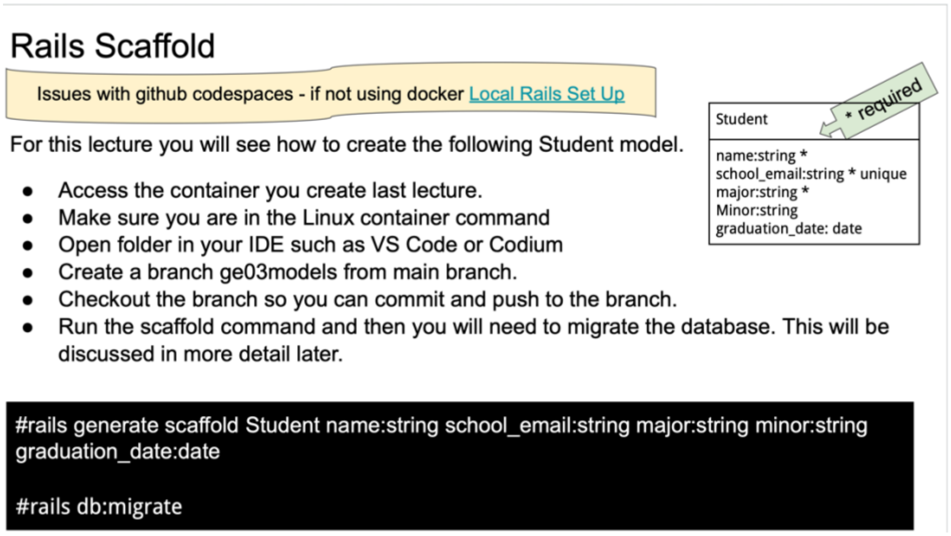
Make sure you are in protfolio_app directory
Create branch
Set as working branch
Add comment to .gitignore file
Integrate branch changes with main
Go to repos
Branches
Overview Yours Active Stale All
Default
main
Your branches
branchTest

```
.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_Hojo61oTCsI0kkHXqu0tQYpE8QGSAuN9-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
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!