New User Onboarding

Skillmate

Section 1: Additions

Necessary access: JIRA, GitHub

JIRA: send your email to Dixon Katzenberg: dkatezenberg@chapman.edu to get access added to the JIRA backlog. Once access is approved, use this link to access it. For more details, check this doc

GitHub: send your email to Julianna Larios: jlarios@chapman.edu to get access added to the GitHub. Once access is approved, use this link to access it. A branch will be made with your name on it that you will use to make your contributions.

Section 2: Setting up the Dev Environment

Necessary Software: Xcode, DataGrip, MySQLWorkbench*

**If already installed, move to Section 3.*

Installing Xcode:

-Open up the App Store
-Search Xcode
-Click Install

Installing MySQLWorkbench:

-Install the app using this link
-Install MySQL using "brew install mysql" in Terminal *(see "Installing Homebrew" if not already installed)*

Installing DataGrip:

-Install the app using this link
-For help on how to use DataGrip, use this link

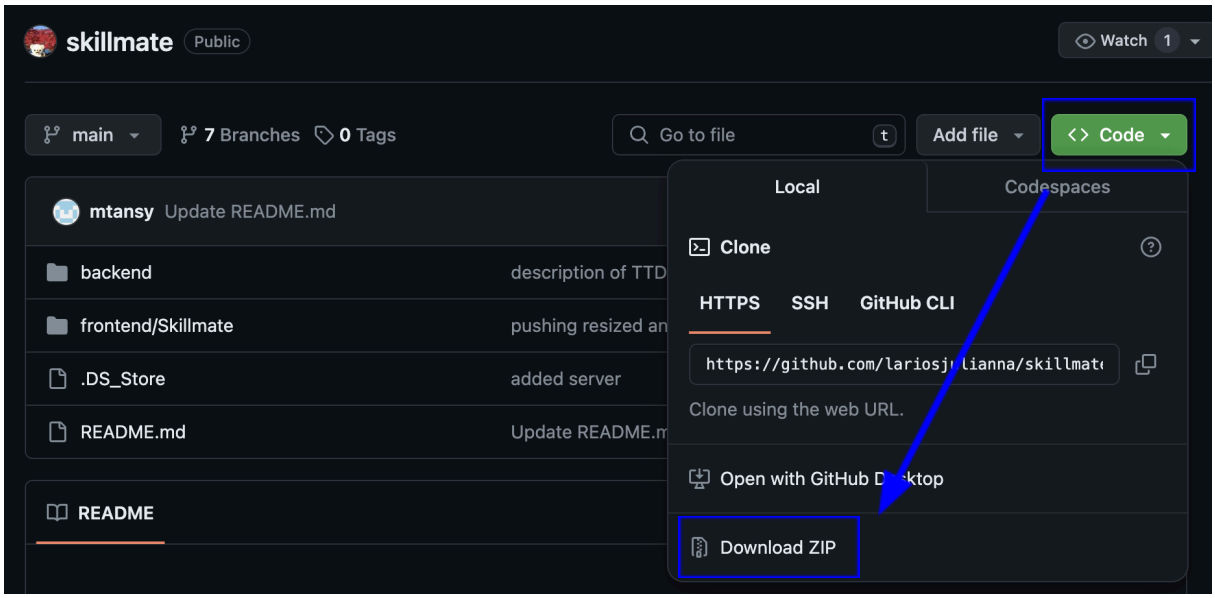
Installing Node: Follow the instructions using this link to install
Installing Homebrew: Follow the instructions using this link to install

If there are any issues or other questions on how to navigate/use any of the necessary software, reach out to a current group member for extra help.

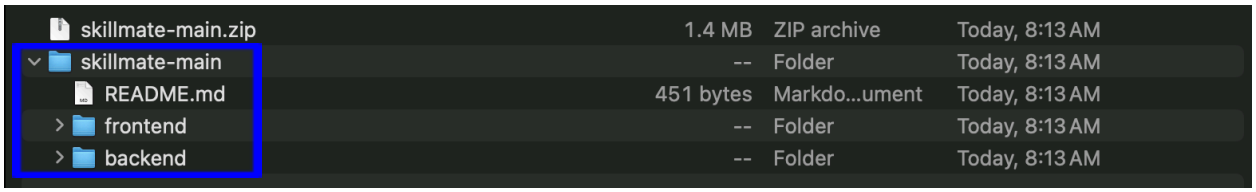
Section 2.5: Setting up the Environment / Running the Project

**Double check all necessary software is installed before beginning*

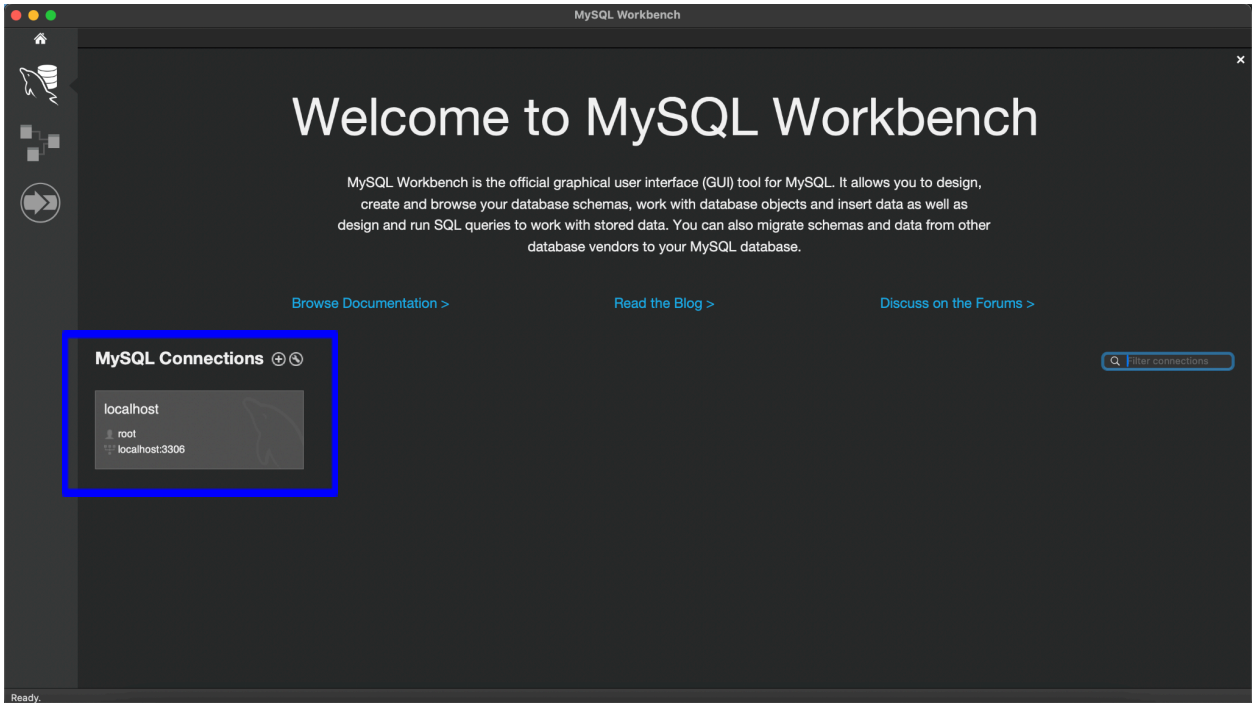
Start by downloading the necessary files from GitHub. On the home page, follow the picture below to download the zip file.



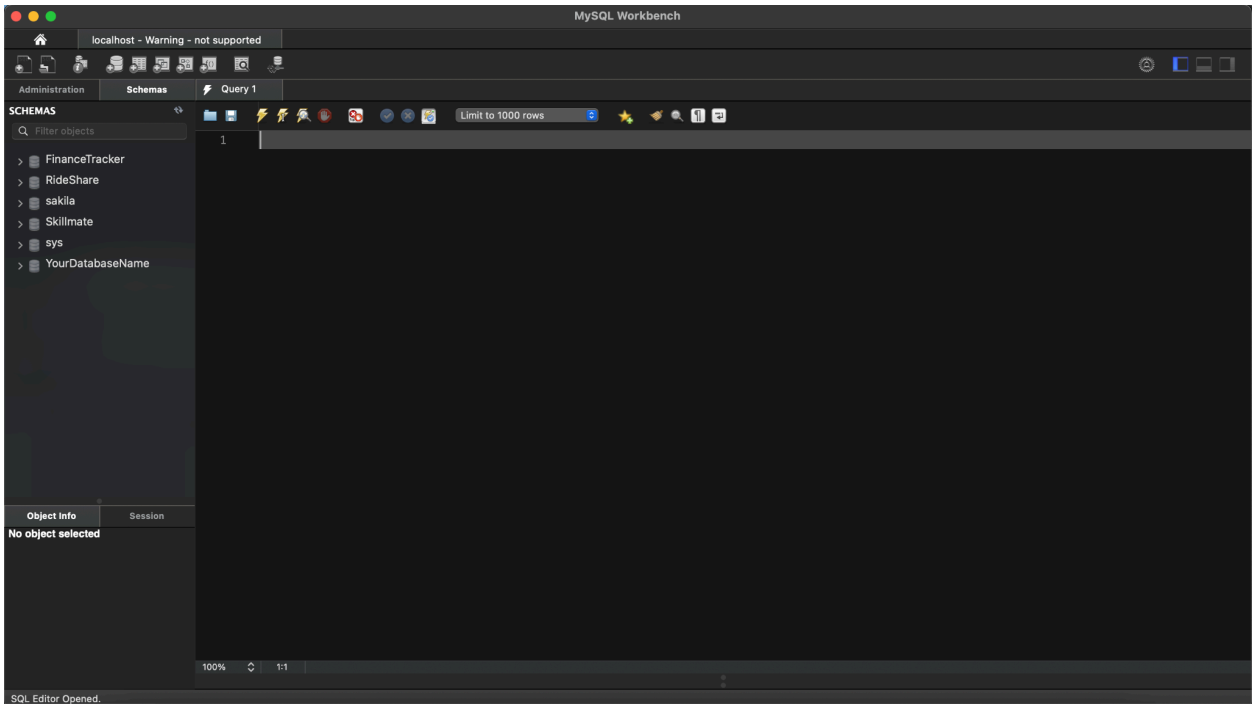
Once downloaded, decompress the files and be sure the following files are present - mainly the frontend/backend files. Be sure to save them in an easily accessible/memorable spot!



Open up MySQLWorkbench and start your server.



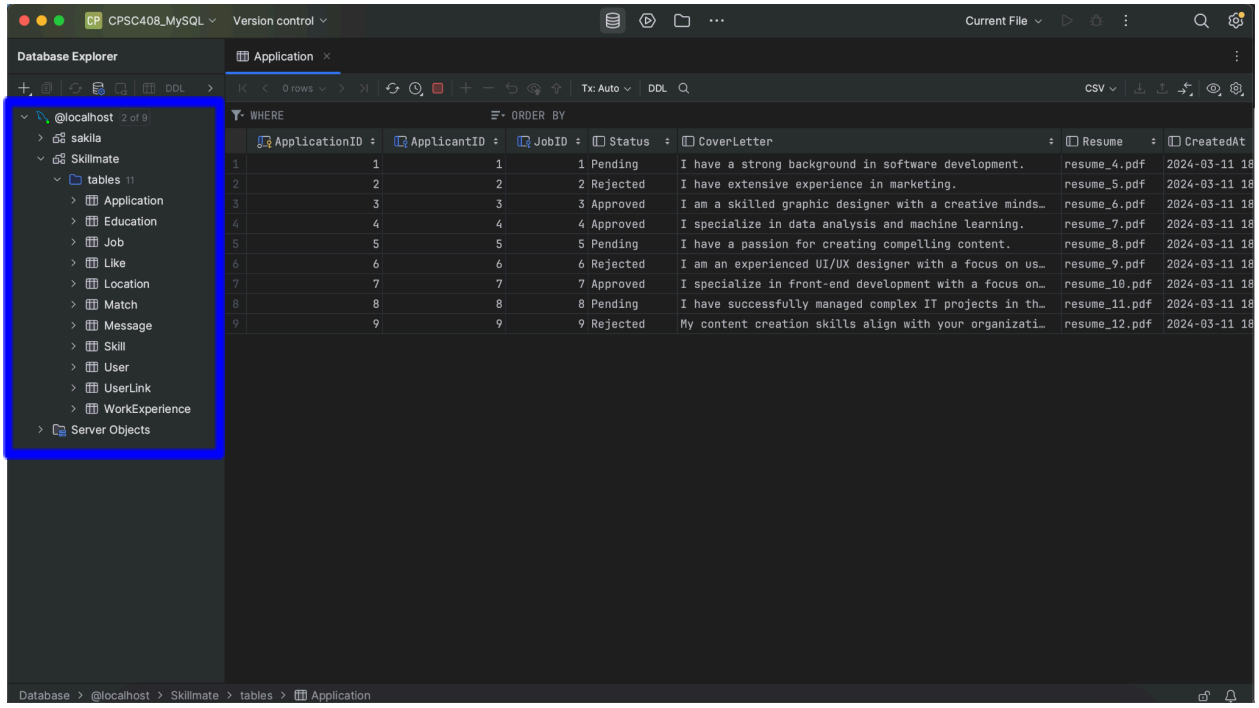
Once the server is running, it should look similar to this.



Open up the backend folder from GitHub in an IDE of your choice (VSCode is recommended). Run the following commands to set up the database and the database server connection. The database server allows Xcode to connect to the database. These will not work unless the MySQL server is running.

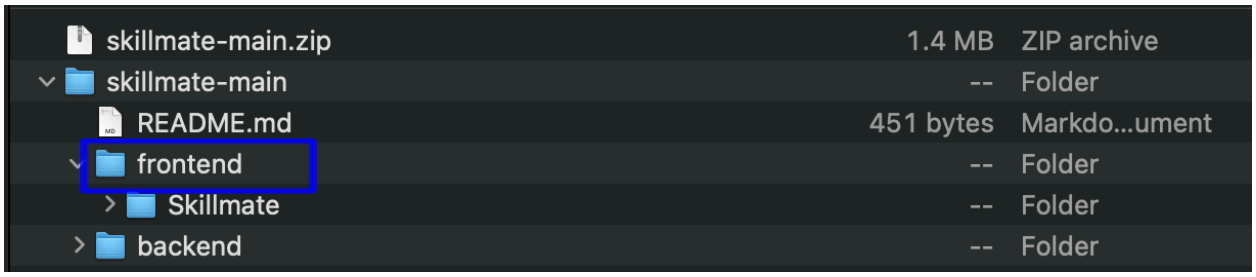
```
madi@madis-MacBook-Pro-2 full skillmate project % cd backend
madi@madis-MacBook-Pro-2 backend % python3 setup.py
madi@madis-MacBook-Pro-2 backend % node server.js
Server is running on http://localhost:3000
```

Open up DataGrip and refresh the page. Make sure that the Skillmate database pops up.

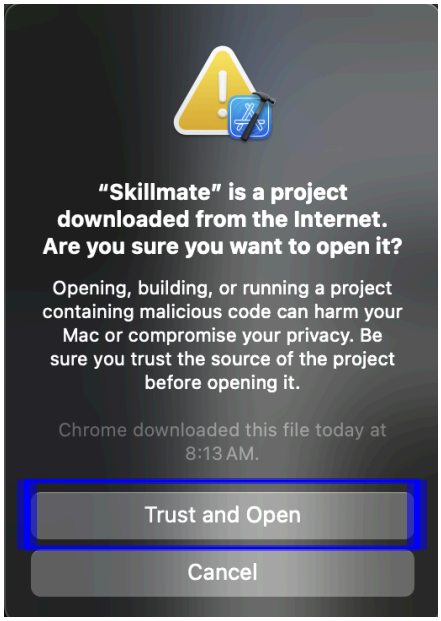


Now that the database is set up, open up Xcode.

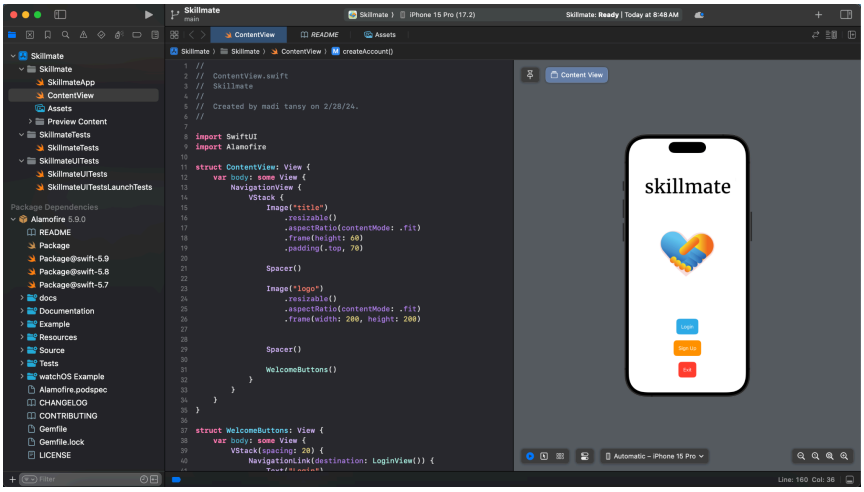
Navigate to where the zip file was downloaded and open the frontend folder “Skillmate”. This folder contains all the necessary files for the Xcode project.



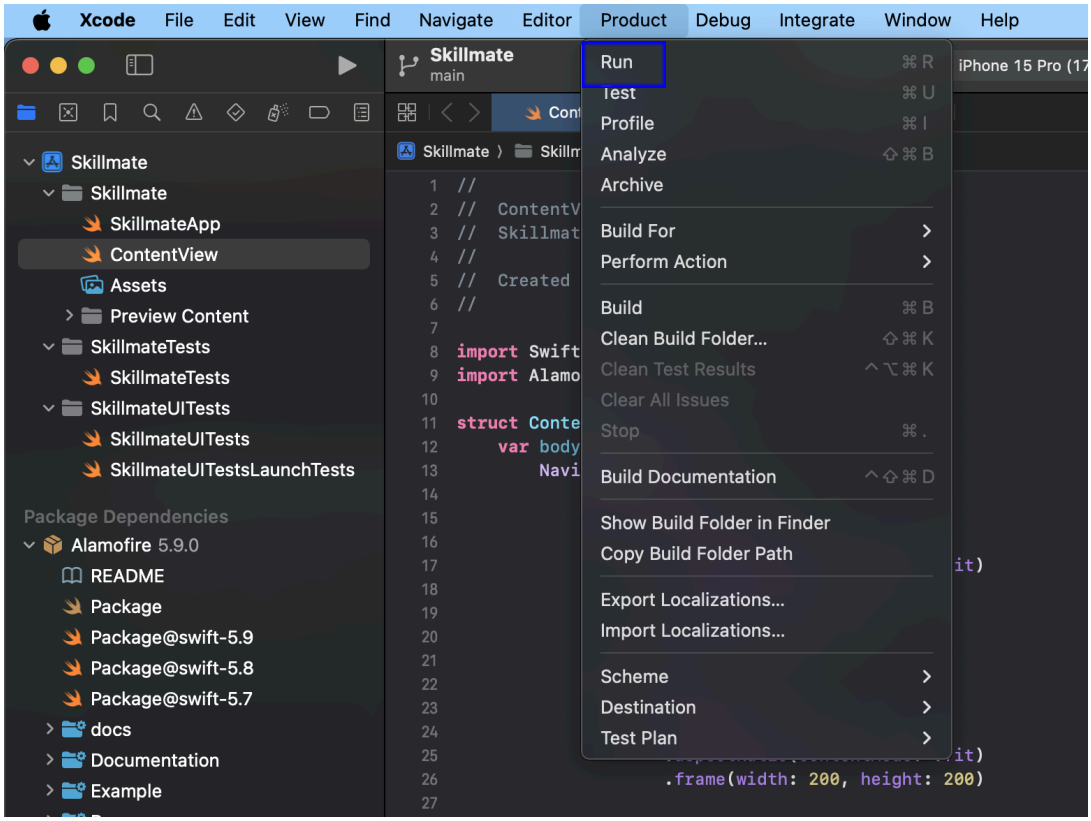
If the following pops up, click “Trust and Open”.



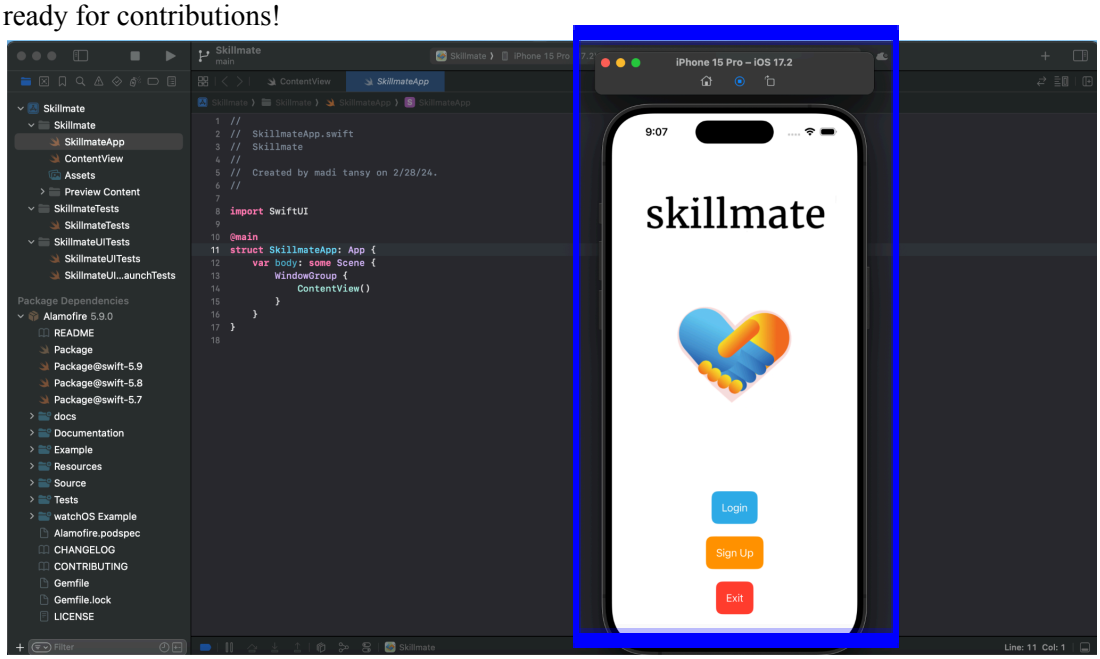
Once loaded, it should look similar to this. (Actual content view may look different at the time of viewing)



To run the project, hover over “Product” and select “Run”.



A pop-up should pop up that looks like the following. This is the simulated version of the app. If this pops up with no problems, everything is working perfectly! Skip Section 4 and move to Section 5 - you are all ready for contributions!

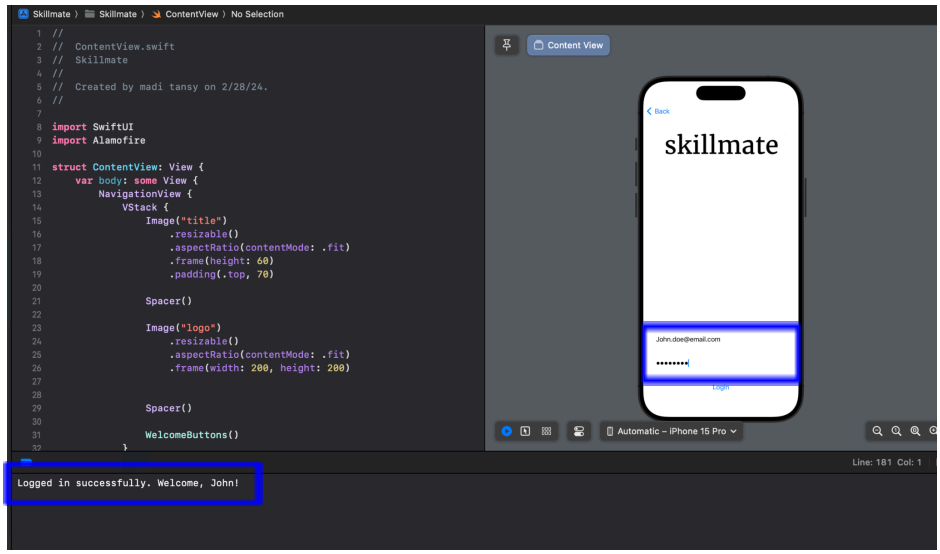


Section 3: Testing/Validating the System

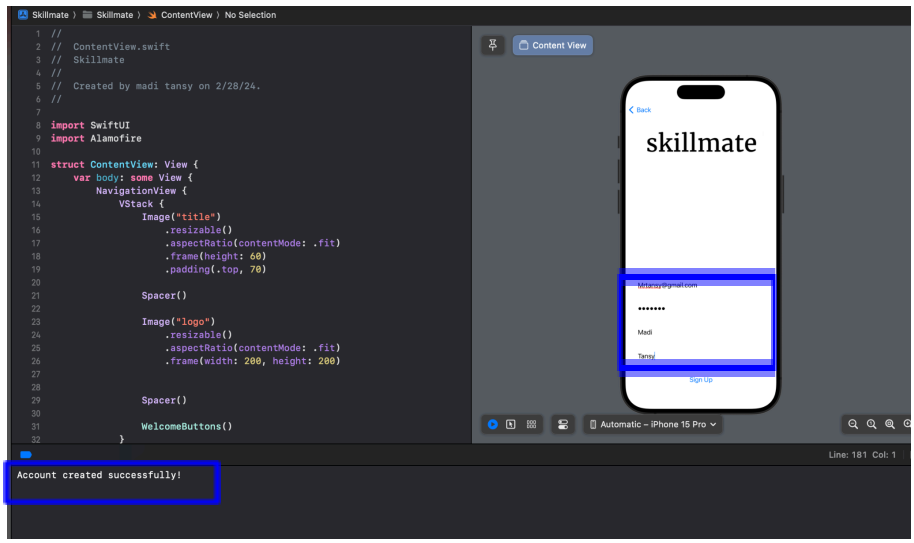
**If there were no errors during set-up, skip this section.*

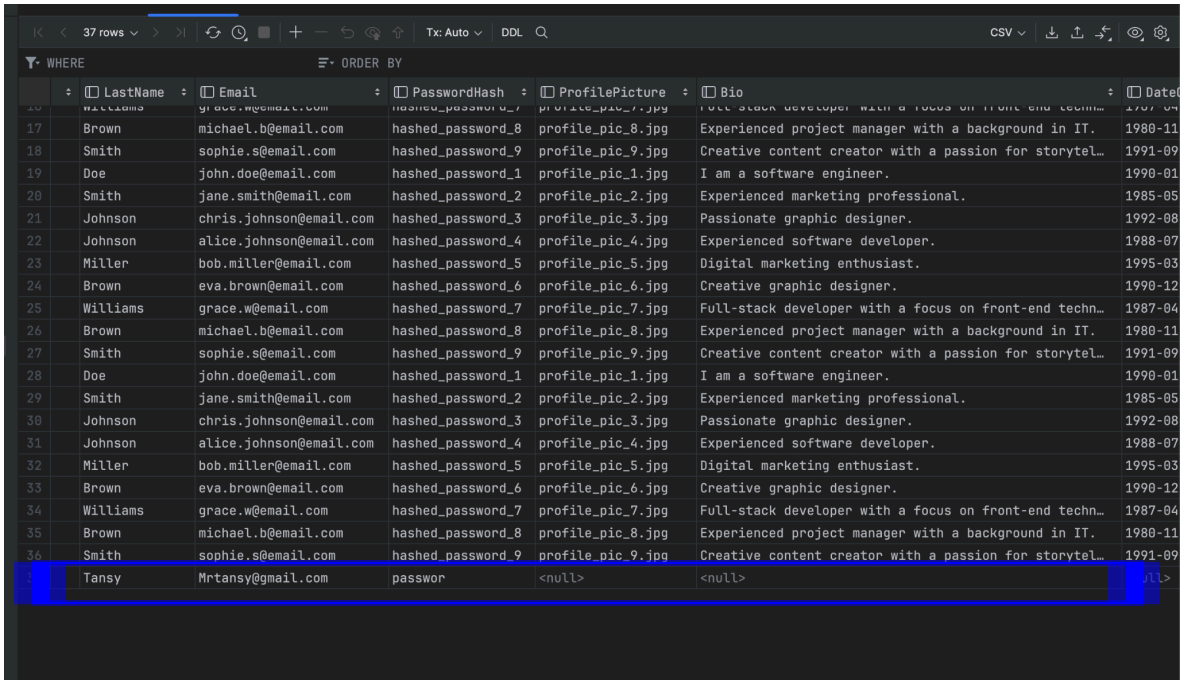
In Xcode, there are built-in unit tests that are run when the product is made. Some other tests that can be run manually test functionality, such as logging in/signing up.

-**Test #1:** Check to be sure that the database is running by opening DataGrip and refreshing the page (done during setup)
-**Test #2:** Test the login feature using a premade user/password from DataGrip. If it works, a welcome message will be printed.



-**Test #3:** Test the sign-up feature using a random email/password. Open up DataGrip to make sure that the account was made.





	LastName	Email	PasswordHash	ProfilePicture	Bio	Date
17	Brown	michael.b@email.com	hashed_password_8	profile_pic_8.jpg	Experienced project manager with a background in IT.	1980-11
18	Smith	sophie.s@email.com	hashed_password_9	profile_pic_9.jpg	Creative content creator with a passion for storytel…	1991-09
19	Doe	john.doe@email.com	hashed_password_1	profile_pic_1.jpg	I am a software engineer.	1990-01
20	Smith	jane.smith@email.com	hashed_password_2	profile_pic_2.jpg	Experienced marketing professional.	1985-05
21	Johnson	chris.johnson@email.com	hashed_password_3	profile_pic_3.jpg	Passionate graphic designer.	1992-08
22	Johnson	alice.johnson@email.com	hashed_password_4	profile_pic_4.jpg	Experienced software developer.	1988-07
23	Miller	bob.miller@email.com	hashed_password_5	profile_pic_5.jpg	Digital marketing enthusiast.	1995-03
24	Brown	eva.brown@email.com	hashed_password_6	profile_pic_6.jpg	Creative graphic designer.	1990-12
25	Williams	grace.w@email.com	hashed_password_7	profile_pic_7.jpg	Full-stack developer with a focus on front-end techn…	1987-04
26	Brown	michael.b@email.com	hashed_password_8	profile_pic_8.jpg	Experienced project manager with a background in IT.	1980-11
27	Smith	sophie.s@email.com	hashed_password_9	profile_pic_9.jpg	Creative content creator with a passion for storytel…	1991-09
28	Doe	john.doe@email.com	hashed_password_1	profile_pic_1.jpg	I am a software engineer.	1990-01
29	Smith	jane.smith@email.com	hashed_password_2	profile_pic_2.jpg	Experienced marketing professional.	1985-05
30	Johnson	chris.johnson@email.com	hashed_password_3	profile_pic_3.jpg	Passionate graphic designer.	1992-08
31	Johnson	alice.johnson@email.com	hashed_password_4	profile_pic_4.jpg	Experienced software developer.	1988-07
32	Miller	bob.miller@email.com	hashed_password_5	profile_pic_5.jpg	Digital marketing enthusiast.	1995-03
33	Brown	eva.brown@email.com	hashed_password_6	profile_pic_6.jpg	Creative graphic designer.	1990-12
34	Williams	grace.w@email.com	hashed_password_7	profile_pic_7.jpg	Full-stack developer with a focus on front-end techn…	1987-04
35	Brown	michael.b@email.com	hashed_password_8	profile_pic_8.jpg	Experienced project manager with a background in IT.	1980-11
36	Smith	sophie.s@email.com	hashed_password_9	profile_pic_9.jpg	Creative content creator with a passion for storytel…	1991-09
	Tansy	Mrtansy@gmail.com	passwor	<null>	<null>	

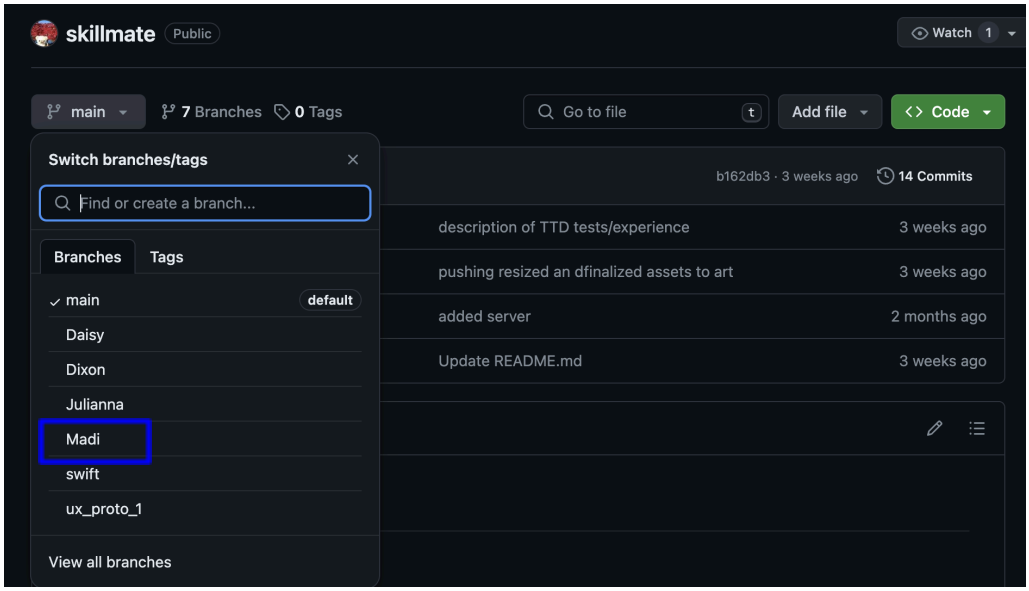
More tests will be added, and members will make notes of where tests are/what tests to run on each new contribution to test if it works.

Section 4: Contributions

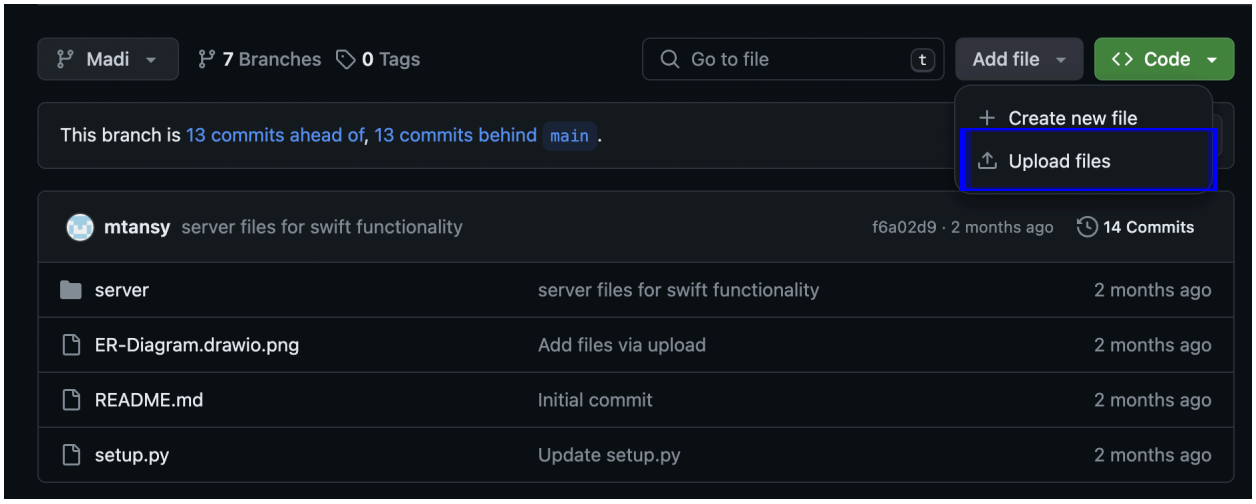
To contribute to the project, be sure to keep track of the files that were changed, and upload to GitHub once done.

To export files manually*:

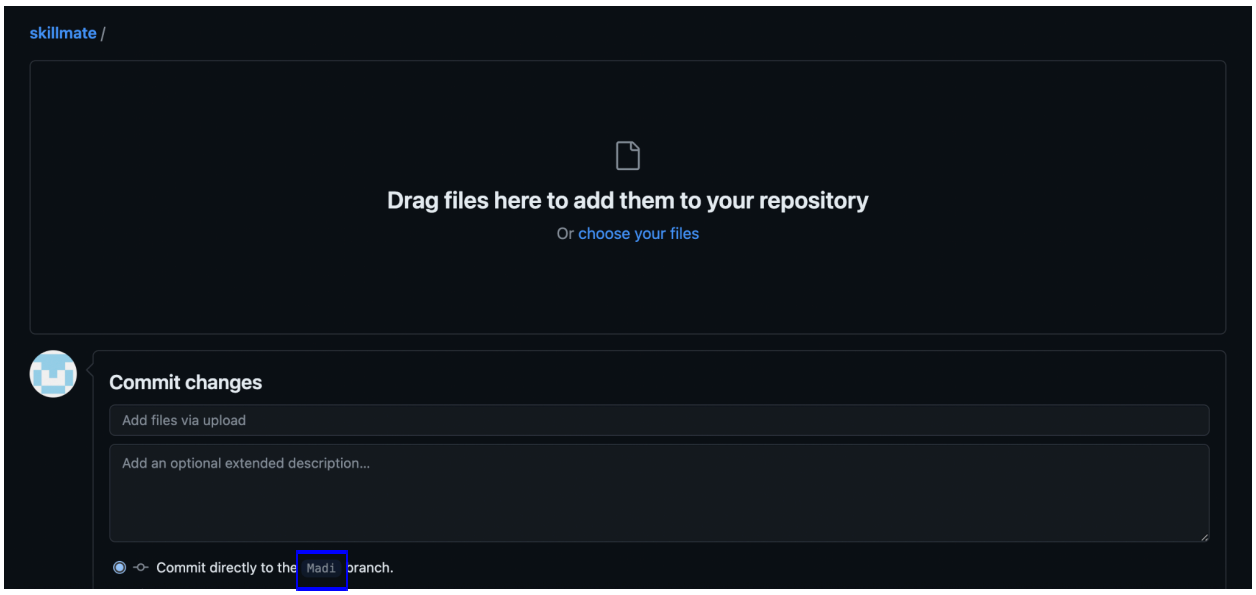
Be sure the file is saved, and locate it in Finder. Open up GitHub, and navigate to your branch.



Once you are on your branch, you can upload new files by clicking “Add file” and then “Upload new files”.



Drag a file into the box or click “choose your files” to choose the files. Be sure that you add a meaningful comment/description explaining what changes you have made, and send a message to the group explaining that you made a new contribution. This ensures that the other members know there is a possible new version made in case multiple members are working on the same section (ex. frontend UI/UX design or backend functionality).



*If you want to commit to GitHub remotely, you are more than welcome to, just be sure to still add meaningful comments and keep in contact with the rest of the group about new progress made

For all contributions: make sure that as you work on the project, add in necessary tests/unit tests. Make a note of what tests you made somewhere, either in your description or a README on your branch.

Contributions to the Main branch:

Before each new iterable is due (typically every 2 weeks) group members will get together to look over all contributions made during the iteration period. During this, progress will be shared and merged to the main branch. Keeping the separate branches helps keep track of individual members' progress, along with allowing more help when debugging. For example, if part of the project from the main branch isn't working, and a member knows that the section that isn't working was a part of their branch and it worked previously, members can go back and get the old version to fix it.