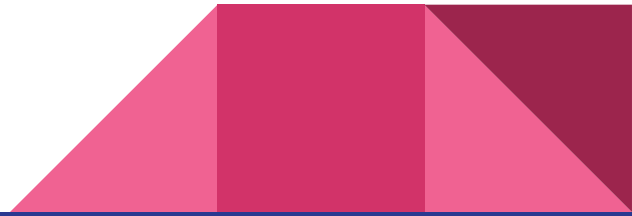
Streamer Discoverability

Milestone 2

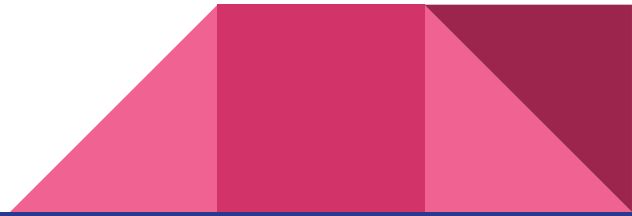
Introduction

1. Milestone 1 is considered achieved, front end and user interface in a satisfactory state
2. Current milestone is to finish backend development, create a web scraper, database, and connect them to the website
3. Some complications and challenges were encountered slowing progress, however milestone 2 is expected to be achieved on time



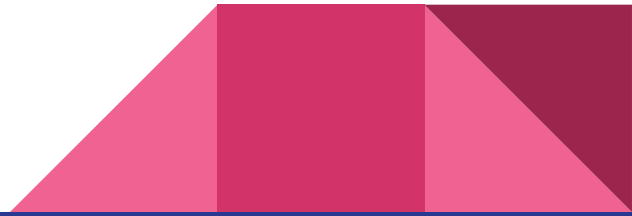
Current Working Features

1. We have the main structure of our website up and running.
2. We also have a functioning upload stream link button to our website for streamers to upload to our database.
3. We have our web scraper and database linked and running together smoothly.



Our Current Progress

A. Our current progress in our second milestone is that we are in a really good spot as we have both parts of our back end development done and communicating with one another.
B. We are currently working on getting our backend and frontend programs to communicate with each other and send the necessary information to the correct spots



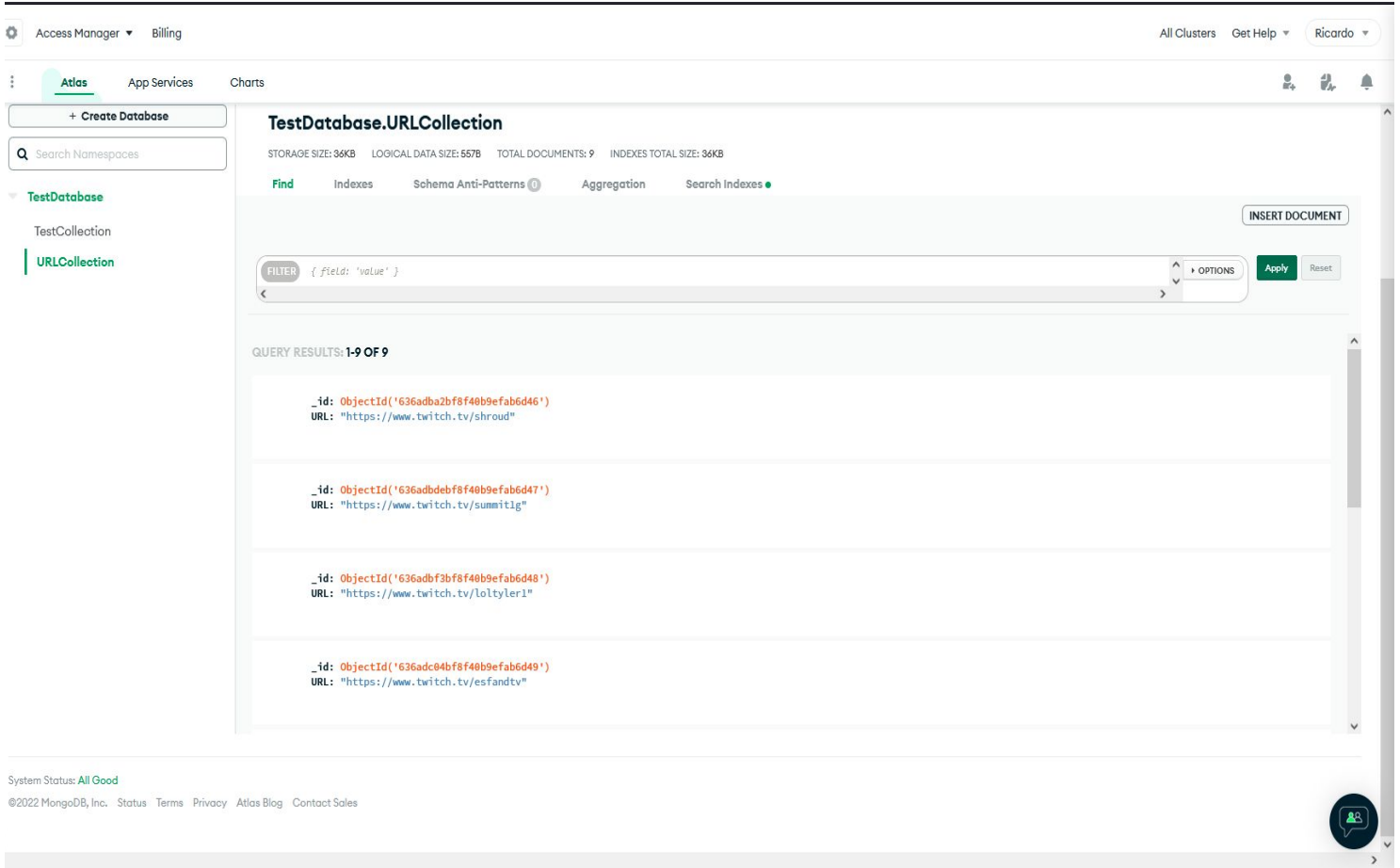

Access Manager
Billing
All Clusters
Get Help
Ricardo
Atlas
App Services
Charts
+ Create Database
Search Namespaces
TestDatabase
TestCollection
URLCollection
TestDatabase.URLCollection
STORAGE SIZE: 36KB
LOGICAL DATA SIZE: 557B
TOTAL DOCUMENTS: 9
INDEXES TOTAL SIZE: 36KB
Find
Indexes
Schema Anti-Patterns 0
Aggregation
Search Indexes
INSERT DOCUMENT
FILTER
{ field: 'value' }
OPTIONS
Apply
Reset
QUERY RESULTS: 1-9 OF 9
_id: ObjectId('636adba2bf8f40b9efab6d46')
URL: "https://www.twitch.tv/shroud"
_id: ObjectId('636adbdebf8f40b9efab6d47')
URL: "https://www.twitch.tv/summit1g"
_id: ObjectId('636adbf3bf8f40b9efab6d48')
URL: "https://www.twitch.tv/loltyler1"
_id: ObjectId('636adc04bf8f40b9efab6d49')
URL: "https://www.twitch.tv/esfandtv"
System Status: All Good
©2022 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

Photo 1

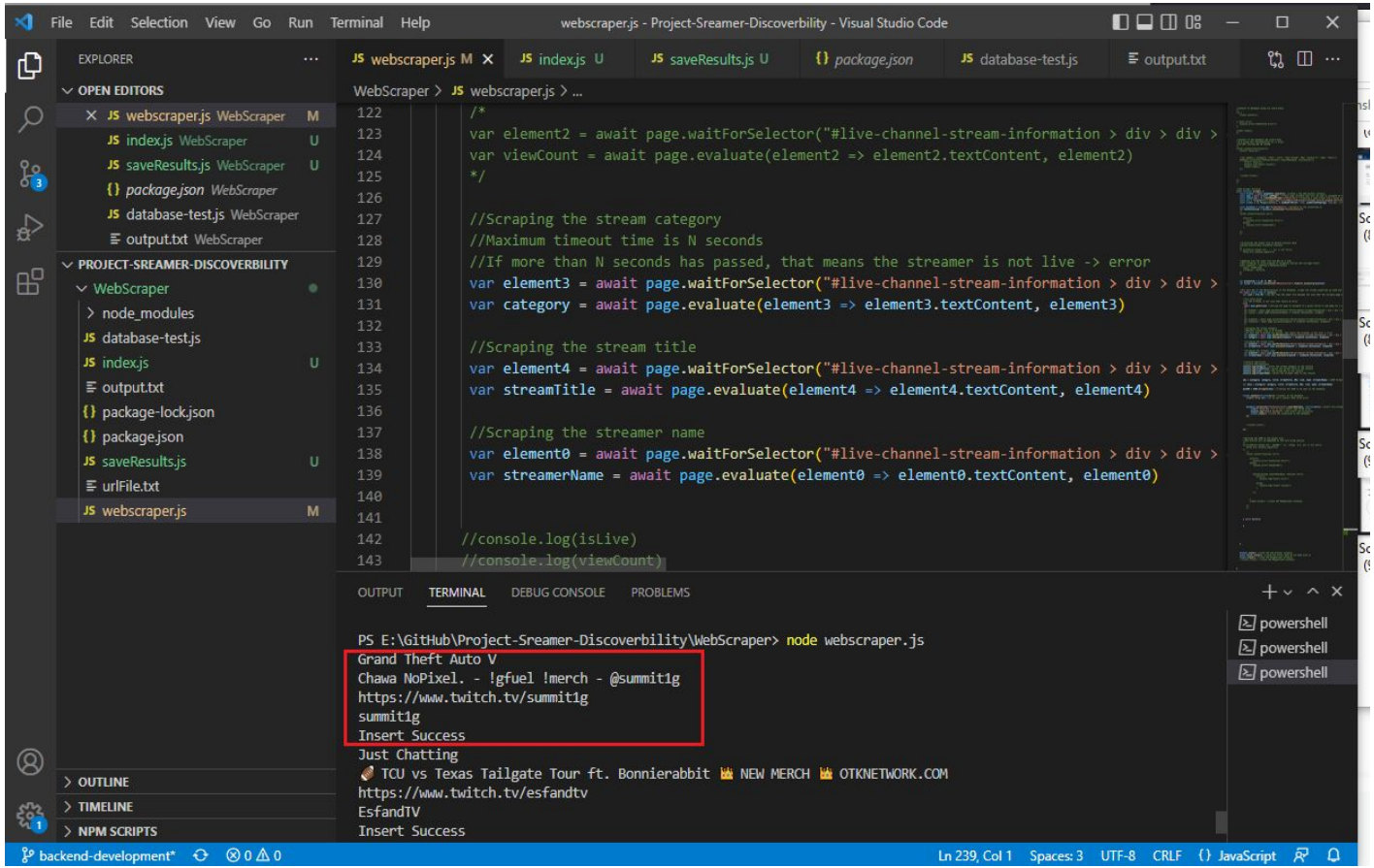

File Edit Selection View Go Run Terminal Help
webscraper.js - Project-Sreamer-Discoverbility - Visual Studio Code
EXPLORER
OPEN EDITORS
webscraper.js WebScraper M
index.js WebScraper U
saveResults.js WebScraper U
package.json WebScraper
database-test.js WebScraper
output.txt WebScraper
PROJECT-SREAMER-DISCOVERBILITY
WebScraper
node_modules
database-test.js
index.js U
output.txt
package-lock.json
package.json
saveResults.js U
urlFile.txt
webscraper.js M
OUTLINE
TIMELINE
NPM SCRIPTS
webscraper.js M
index.js U
saveResults.js U
package.json
database-test.js
output.txt
WebScraper > webscraper.js > ...
/*
var element2 = await page.waitForSelector("#live-channel-stream-information > div > div >
var viewCount = await page.evaluate(element2 => element2.textContent, element2)
*/
//Scraping the stream category
//Maximum timeout time is N seconds
//If more than N seconds has passed, that means the streamer is not live -> error
var element3 = await page.waitForSelector("#live-channel-stream-information > div > div >
var category = await page.evaluate(element3 => element3.textContent, element3)
//Scraping the stream title
var element4 = await page.waitForSelector("#live-channel-stream-information > div > div >
var streamTitle = await page.evaluate(element4 => element4.textContent, element4)
//Scraping the streamer name
var element0 = await page.waitForSelector("#live-channel-stream-information > div > div >
var streamerName = await page.evaluate(element0 => element0.textContent, element0)
//console.log(isLive)
//console.log(viewCount)
OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS
PS E:\GitHub\Project-Sreamer-Discoverbility\WebScraper> node webscraper.js
Grand Theft Auto V
Chawa NoPixel. - !gfuel !merch - @summit1g
https://www.twitch.tv/summit1g
summit1g
Insert Success
Just Chatting
TCU vs Texas Tailgate Tour ft. Bonnierabbit NEW MERCH OTKNETWORK.COM
https://www.twitch.tv/esfandtv
EsfandTV
Insert Success
powershell
powershell
powershell
backend-development*
Ln 239, Col 1 Spaces: 3 UTF-8 CRLF JavaScript

Photo 2

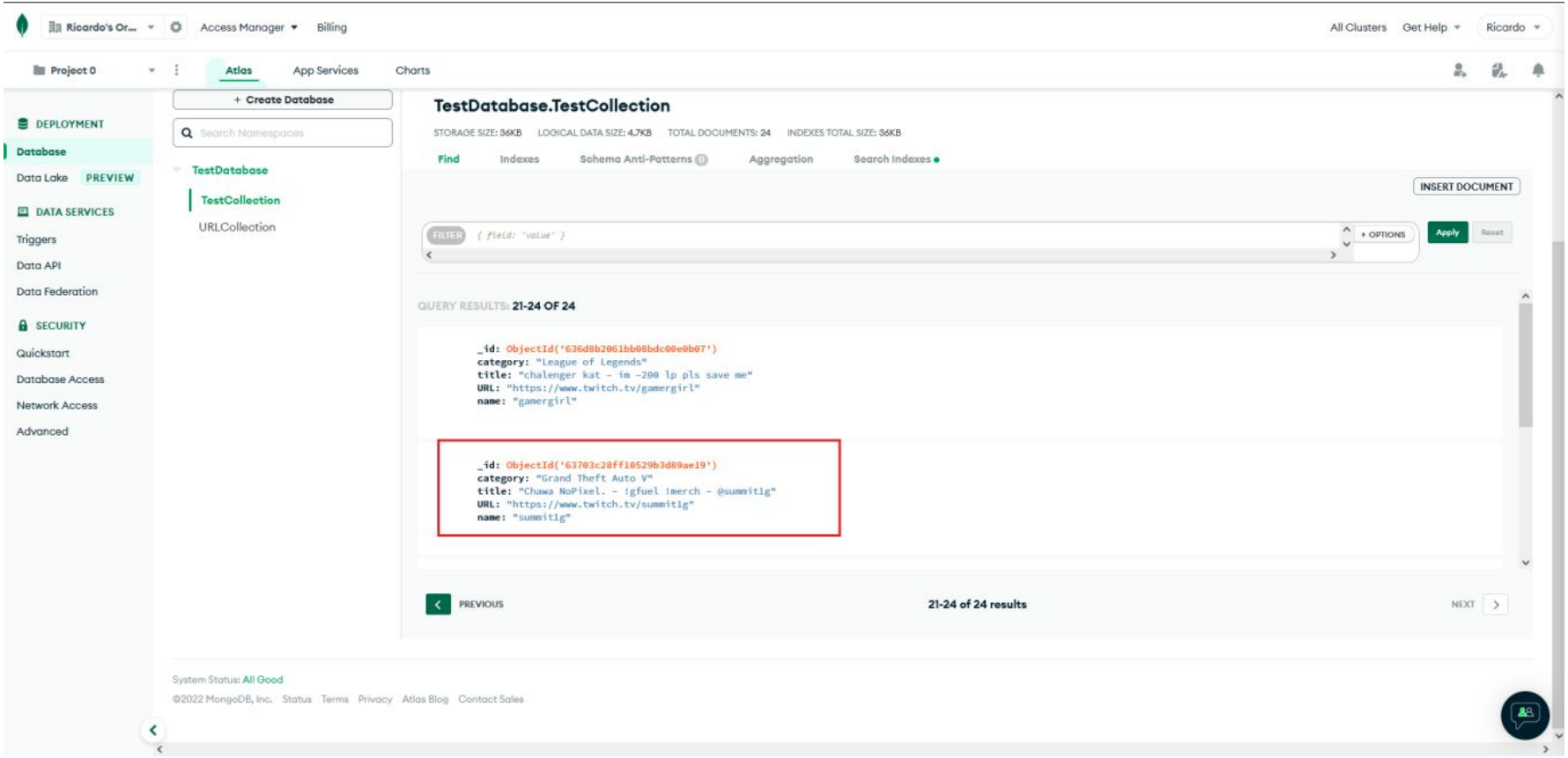

Ricardo's Or...
Access Manager
Billing
All Clusters
Get Help
Ricardo
Project 0
Atlas
App Services
Charts
DEPLOYMENT
Database
Data Lake
PREVIEW
DATA SERVICES
Triggers
Data API
Data Federation
SECURITY
Quickstart
Database Access
Network Access
Advanced
+ Create Database
Search Namespaces
TestDatabase
TestCollection
URLCollection
TestDatabase.TestCollection
STORAGE SIZE: 36KB
LOGICAL DATA SIZE: 4.7KB
TOTAL DOCUMENTS: 24
INDEXES TOTAL SIZE: 36KB
Find
Indexes
Schema Anti-Patterns
Aggregation
Search Indexes
INSERT DOCUMENT
FILTER
{ field: 'value' }
OPTIONS
Apply
Reset
QUERY RESULTS: 21-24 OF 24
_id: ObjectId('636d8b2061bb08bdc00e0b07')
category: "League of Legends"
title: "chalenger kat - im -200 lp pls save me"
URL: "https://www.twitch.tv/gamergirl"
name: "gamergirl"
_id: ObjectId('63703c28ff10529b3d89ae19')
category: "Grand Theft Auto V"
title: "Chawa NoPixel. - !gfuel !merch - @summit1g"
URL: "https://www.twitch.tv/summit1g"
name: "summit1g"
PREVIOUS
21-24 of 24 results
NEXT
System Status: All Good
©2022 MongoDB, Inc.
Status
Terms
Privacy
Atlas Blog
Contact Sales

Photo 3

Our Final Goals

- Our final goal in our project is to get our website functioning as we envisioned during our development stage which entails it giving users a document with different streamers twitch links allowing them to discover new streamers to watch in whatever category or game they choose to watch.

Challenges faced in Milestone 2

Carlos (UI/UX):

- Understanding the proper connection between a database and a PHP website
  - Acquiring the proper programs for connecting the two
  - Dealing with the issues of the database not connecting due to issues
- Getting the data from the tables to properly format to the style
  - Multiple testing to get the data to display properly first and then attempts for styling it to our skeleton mock-up
- Edit PHP internal files for proper connection
  - Looking for the proper setup for localhost to allow for connections and displaying PHP files correctly

Tien (Back-end):

- Creating connection between the web scraper and the database using MongoDB and Mongoose
- Getting the data from the web scraper and separating them into different collections on MongoDB based on their categories
- Having people access to the database using MongoDB Compass
- Understanding the connection between frontend and backend by using PHP

Other Challenges faced

Ricardo (Back-end):

- Learning how to use Node.js and building a web scraper with it.
- Establishing a connection to a MongoDB database from the web scraper program.
- Understanding how MongoDB works and how to read/insert data with the MongoDB Node.js module.

Uriel (back-end):

- Deciding on the best path to gather all of the data needed from Twitch website.
- Trying to resolve several unprecedented issues in connecting the MySQL database to the front end. For example:
  - Resetting MySQL root password
  - Giving team members access to database
  - Editing Mysql files eg. my.cnf

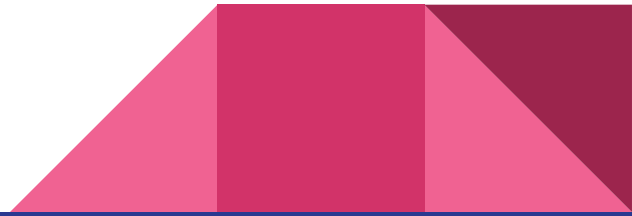
Roles

Carlos - UI/UX designer

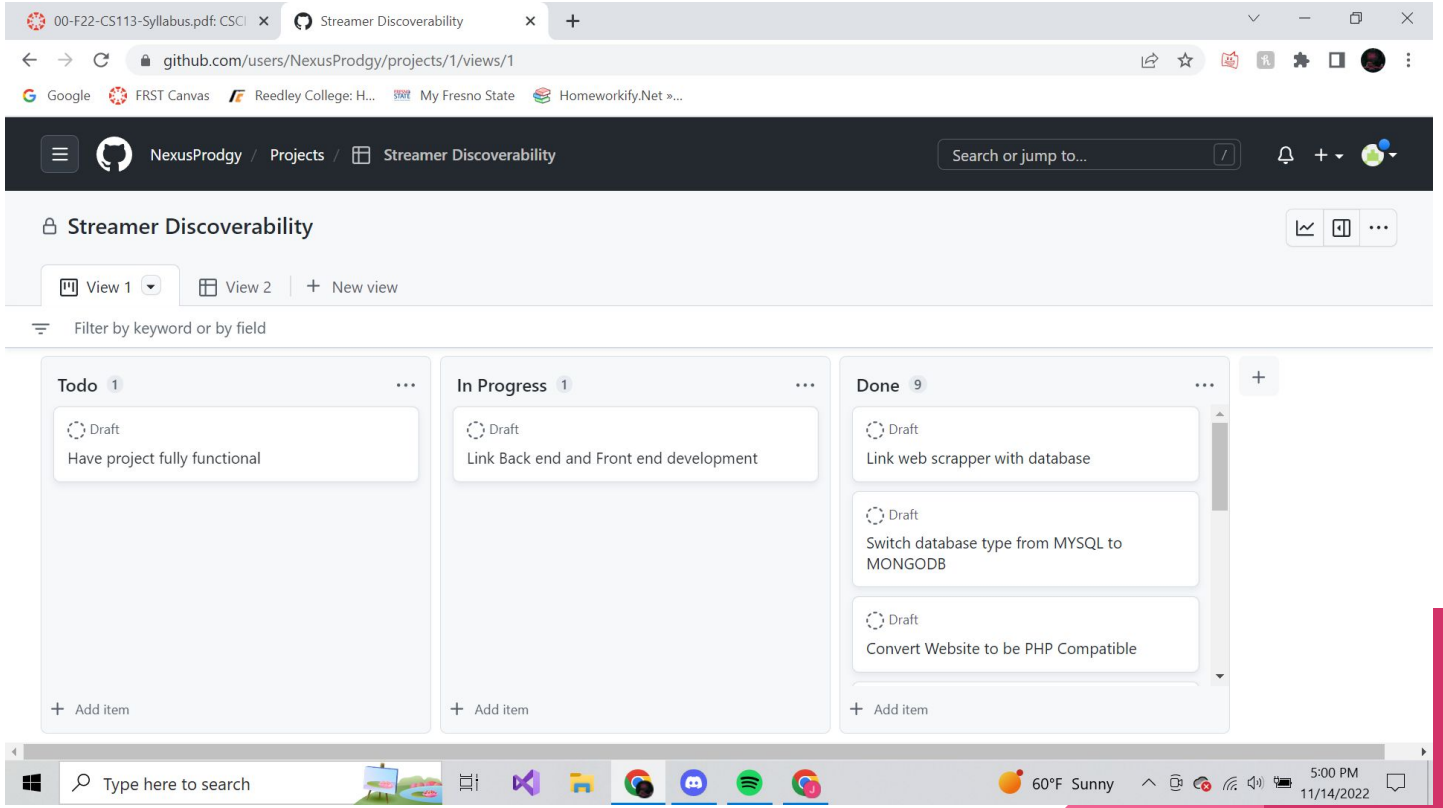
Ricardo,Uriel,Tien- backend developers

Javier-Project Manager

Joe- software tester



Project Board



Repository link

https://github.com/NexusProdgy/Project-Sreamer-Discoverbility/tree/main