

SPITRACK

Simplifying Research Techniques



Team Members



Mohammad Nabiluzzaman Neloy

1831251042

mohammad.nabiluzzaman@northsouth.edu



Rafid Ahmed

1831395642

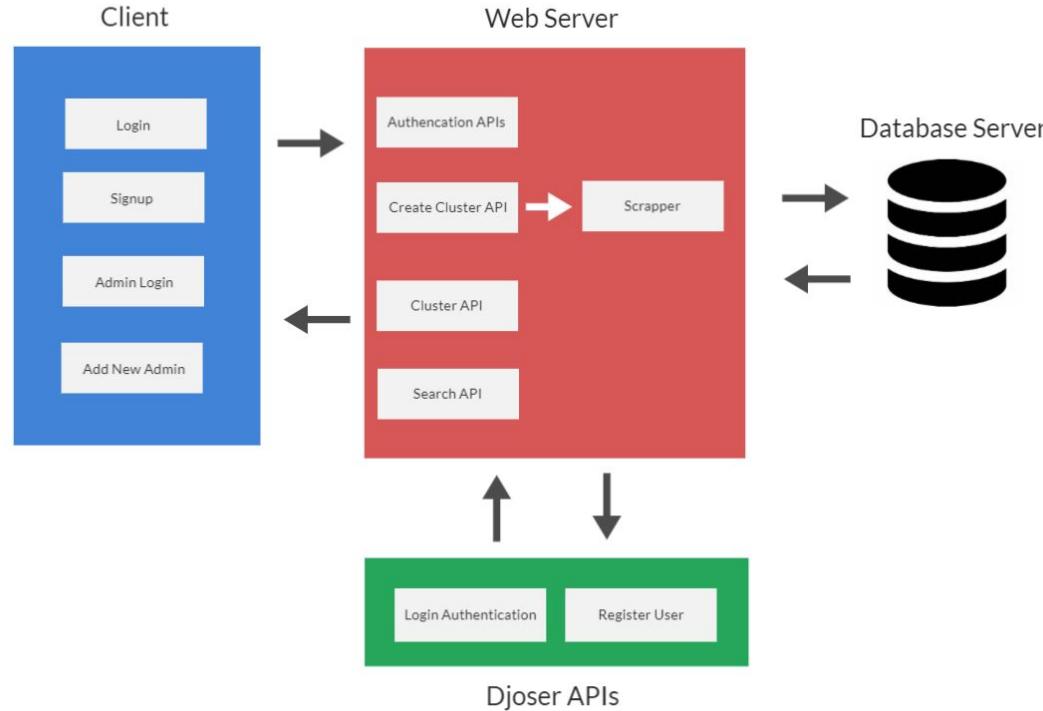
rafid.ahmed@northsouth.edu

Description

- Spittrack is an online system that enables researchers to collect necessary documents and information from a given cluster of URLs, and then perform some research on the collected information using a specific search engine.



High Level Software Architecture



Software Stack

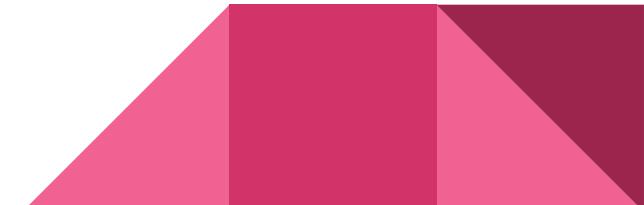
Software

- UI Design – Adobe XD
- Version Control System – Git
- Repository Hosting Service – Bitbucket
- Task Management – Trello
- Communication Platform – Slack
- Backend Development Using Python (IDE) – JetBrains PyCharm
- Frontend Development (IDE) – Visual Studio Code
- Android Development – Android Studio
- Local Database Server (SQL Database) – Apache XAMPP
- Backend Server – Heroku
- Frontend Hosting – Firebase
- Database Server – AWS RDS

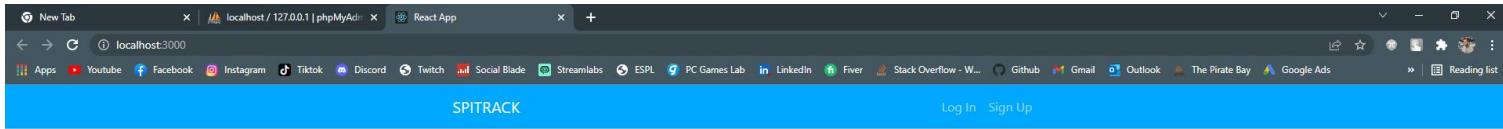
Software Stack

Technologies

- Django
- React JS
- BeautifulSoup
- HTML
- CSS
- Bootstrap
- JavaScript
- Python
- XML
- Java
- MySQL
- Django REST Framework



Project Results (Web-App)

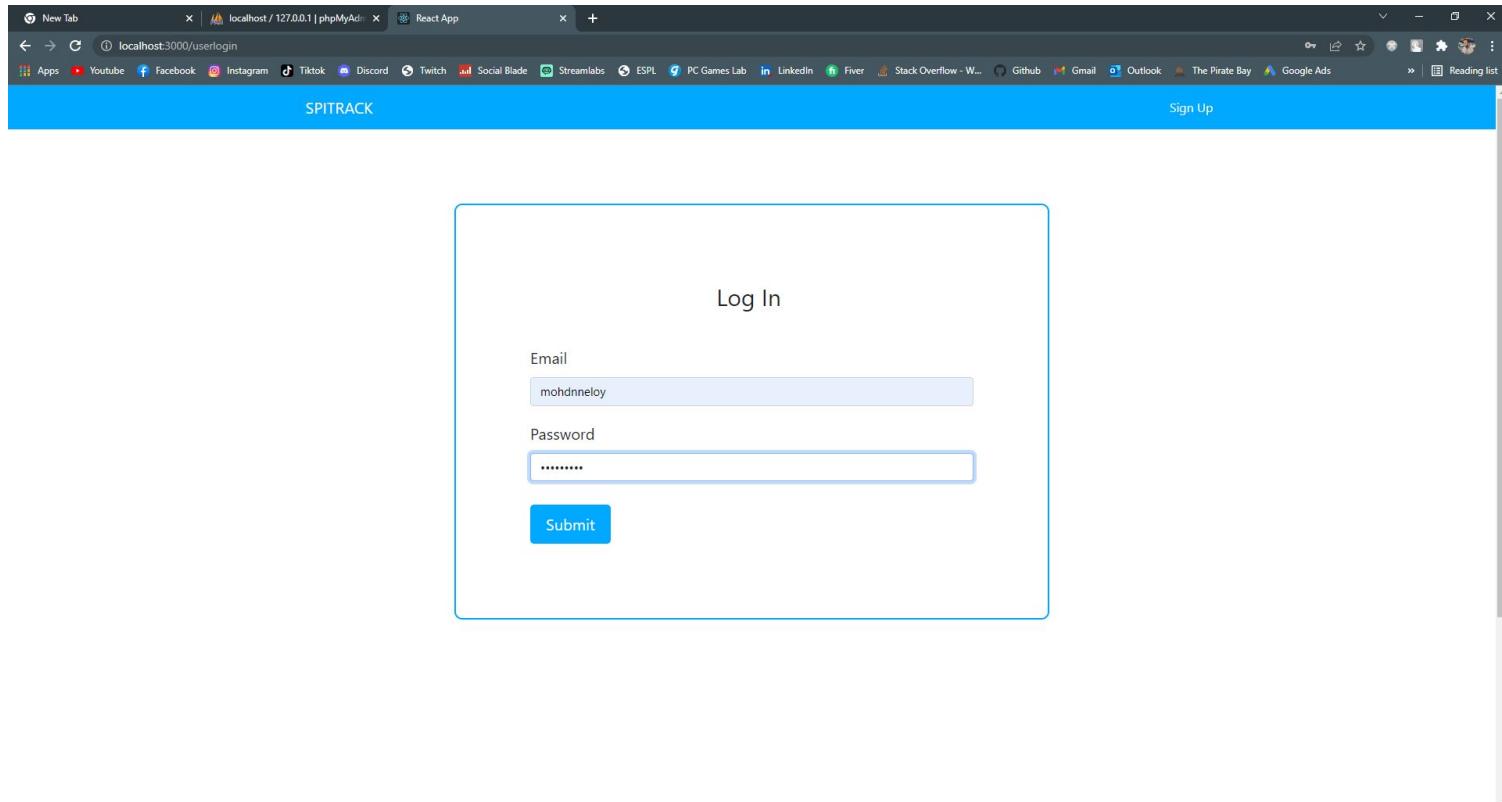


SPITRACK

Simplifying Research Techniques



Project Results (Web-App)



Project Results (Web-App)

The screenshot shows a web browser window with a sidebar and a main content area.

Sidebar (Left):

- SPITRACK
- Dashboard
- Clusters
- Search
- Logout**

Main Content Area:

Create Cluster

Cluster Name

URL1

URL2

URL3

Depth

Select Strategies

PDF (.pdf)
 Word (.docx)
 Powerpoint (.pptx)

Project Results (Web-App)

The screenshot shows a web browser window for a project named SPITRACK. The left sidebar contains navigation links: Dashboard, Clusters, Search, and Logout. The main content area is titled "localhost3000 says" and displays a message: "We will notify you through email when your cluster is ready for searching:" with an "OK" button. Below this, there are three input fields labeled URL1, URL2, and URL3, each containing a URL. There is also a "Depth" input field set to 3. A section titled "Select Strategies" contains checkboxes for PDF (.pdf), Word (.docx), Powerpoint (.pptx), Text (.txt), and Non-HTML Text (.htm). A "Create" button is located at the bottom of the form.

localhost3000 says

We will notify you through email when your cluster is ready for searching:

URL1
https://paperswithcode.com/paper/implicit-pdf-non-parametric-repre

URL2
https://www.daily-sun.com/online/sports

URL3
https://en.wikipedia.org/wiki/Operating_system

Depth
3

Select Strategies

PDF (.pdf)
 Word (.docx)
 Powerpoint (.pptx)
 Text (.txt)
 Non-HTML Text (.htm)

Create

Project Results (Web-App)

The screenshot shows a code editor interface with a dark theme. The left sidebar displays a project structure for a Python application named 'EINeloy/Programm'. The 'backend' directory contains several files: __init__.py, admin.py, apps.py, emailnotification.py, models.py, scrapers.py, serializers.py, tests.py, urls.py, and views.py. The 'clusters' directory also contains __init__.py, admin.py, apps.py, emailnotification.py, models.py, scrapers.py, serializers.py, tests.py, urls.py, and views.py. The 'frontend' directory includes ReadMe.txt, settings.py, and Test URLs. The 'External Libraries' section lists Requests and Scrapy. The main editor area shows a Python script for creating a new cluster. The terminal below shows the execution of the script, outputting a list of URLs and their depths, followed by crawling logs for two URLs.

```
===== API - Create New Cluster =====
@api_view(['POST'])
def createCluster(request):
    # Finding User Data From Database
    userid = request.user.id
    user = User.objects.filter(id=userid)
    print(userid)

    # Preparing data to store in Cluster Model
    username = user[0].username
    email = user[0].email
    name = user[0].first_name + " " + user[0].last_name
    cluster_name = request.data.get("cluster_name", None)
    url1 = request.data.get("url1", None)
    url2 = request.data.get("url2", None)
    url3 = request.data.get("url3", None)
    depth = request.data.get("depth", None)
    pdf = request.data.get("pdf", None)
    word = request.data.get("word", None)
    powerpoint = request.data.get("powerpoint", None)

    # Creating Cluster Model
    cluster = Cluster.objects.create(
        name=name,
        url1=url1,
        url2=url2,
        url3=url3,
        depth=depth,
        pdf=pdf,
        word=word,
        powerpoint=powerpoint
    )
    return Response({'status': 'Cluster Created'})

createCluster()

['url': 'https://twitter.com/paperswithcode', 'depth': 3}
{'url': 'https://abs.twimg.com/responsive-web/client-web-legacy/icon-svg.168b89d5.svg', 'depth': 3}
{'url': 'https://abs.twimg.com/responsive-web/client-web-legacy/icon-ios.b1fc7275.png', 'depth': 3}
{'url': 'https://help.twitter.com/using-twitter/twitter-supported-browsers', 'depth': 3}
{'url': 'https://twitter.com/tos', 'depth': 3}
{'url': 'https://twitter.com/privacy', 'depth': 3}
{'url': 'https://support.twitter.com/articles/20170514', 'depth': 3}
{'url': 'https://legal.twitter.com/impprint', 'depth': 3}
{'url': 'https://business.twitter.com/en/help/troubleshooting/how-twitter-ads-work.html?ref=web-twc-ao-gbl-adsinfo&utm_source=twc&utm_medium=web&utm_campaign=ao&utm_content=adsinfo', 'depth': 3}

Crawling URL Now: https://abs.twimg.com/responsive-web/client-web-legacy/polyfills.b93a0985.js at depth: 2 =====
Crawling URL Now: https://abs.twimg.com/responsive-web/client-web-legacy/vendors~main.35bca545.js at depth: 2 =====
```

File Edit View Navigate Code Refactor Run Tools Git Window Help Project - views.py

Project

Prj. Project EINeloy/Programm

ReadMe.txt settings.py backend/urls.py models.py serializers.py scrapers.py emailnotification.py views.py clusters/urls.py

File Local Terminal Event Log

48:44 CRLF UTF-8 4 spaces Python 3.8 master

Project Results (Web-App)

The screenshot shows the PyCharm IDE interface with the following details:

- Project Structure:** The left sidebar displays the project structure under "Project". It includes a "backend" folder containing "clusters" and "migrations" subfolders, and "users" and "manage.py" files. A "frontend" folder contains "Readme.txt" and "Test URLs".
- Code Editor:** The main window shows a Python file named "views.py". The code implements an API endpoint for creating a new cluster. It retrieves user data from the database, prepares data for the Cluster model, and handles optional parameters like URL, PDF, Word, and Powerpoint files.
- Terminal:** At the bottom, the terminal window shows the output of a script. It lists URLs to crawl, noting that one URL was not crawled due to max tries or authentication required. It also lists specific URLs found, categorized by type (PDF URLs).
- Status Bar:** The bottom status bar indicates "2 new notifications", the current time "48:44", and the Python version "Python 3.8".

Project Results (Web-App)

The screenshot shows a code editor interface with a Python project structure and a terminal window below it.

Project Structure:

- Project FIneloy/Programmer
- backend
- clusters
 - migrations
 - admin.py
 - apps.py
 - emailnotification.py
 - models.py
 - scrappers.py
 - serializers.py
 - tests.py
 - urls.py
 - views.py
- users
- management.py
- frontend
 - ReadMe.txt
 - Test URLs
- External Libraries
- Scratches and Consoles

Terminal Output:

```
skip
Entered
Done PDF Parsing: https://wiki.creativecommons.org/images/a/aa/Bylaws_-_Amended_and_Restated_March_4_2018_%28final%29.pdf
Entered
skip
Entered
skip
Entered
Entered
Done PDF Parsing: https://wiki.creativecommons.org/images/0/01/Creativecommons-licensing-and-marking-your-content_eng.pdf
Entered
UserWarning: startxref on same line as offset [pdf.py:1709]
Done PDF Parsing: https://creativecommons.org.au/content/attributionccmaterials.pdf
Entered
Done PDF Parsing: https://wiki.creativecommons.org/images/d/d6/Ccrel-1.0.pdf
```

Bottom Bar:

- Git
- TODO
- Problems
- Python Packages
- Python Console
- Terminal
- Show desktop
- Event Log

Project Results (Web-App)

The screenshot shows the PyCharm IDE interface with a Python project named "Project R\Neloy\Programmer". The project structure is displayed in the left sidebar, showing a "backend" directory containing "clusters" and "users" sub-directories, along with "migrations", "admin.py", "appsettings.py", "emailnotification.py", "models.py", "scrappers.py", "tests.py", "urls.py", and "views.py". The "clusters" directory contains files like "init_.py", "models.py", "scrappers.py", and "views.py". The "users" directory contains "management.py". The "views.py" file in the "clusters" directory is open in the main editor, showing code for creating a new cluster via a POST request. The code uses Django's User model to find user data and creates a Cluster model instance. The "Terminal" tab at the bottom shows the command "createCluster()", followed by several "Entered" and "skip" messages, indicating a test run. The status bar at the bottom right shows the current time as 40:44, file encoding as CRLF, character set as UTF-8, and spaces as 4 spaces, with Python 3.8 selected.

```
#===== API - Create New Cluster =====
@api_view(['POST'])
def createCluster(request):
    # Finding User Data From Database
    userid = request.user.id
    user = User.objects.filter(id=userid)
    print(userid)

    # Preparing data to store in Cluster Model
    username = user[0].username
    email = user[0].email
    name = user[0].first_name + " " + user[0].last_name
    cluster_name = request.data.get("cluster_name", None)
    url1 = request.data.get("url1", None)
    url2 = request.data.get("url2", None)
    url3 = request.data.get("url3", None)
    depth = request.data.get("depth", None)
    pdf = request.data.get("pdf", None)
    word = request.data.get("word", None)
    powerpoint = request.data.get("powerpoint", None)

    cluster = Cluster.objects.create(
        name=name,
        cluster_name=cluster_name,
        url1=url1,
        url2=url2,
        url3=url3,
        depth=depth,
        pdf=pdf,
        word=word,
        powerpoint=powerpoint
    )
    return Response(status=201)
```

```
Entered
skip
Entered
skip
Entered
skip
Entered
skip
Done with PDF URLs
[28/Feb/2022 23:37:38] "POST /cluster/createcluster HTTP/1.1" 201 347
1
[28/Feb/2022 23:37:39] "GET /cluster/clusterdetails HTTP/1.1" 200 1039
```

File Git TODO Problems Python Packages Python Console Terminal Event Log 40:44 CRLF UTF-8 4 spaces Python 3.8 master

Project Results (Web-App)

The screenshot shows a Gmail inbox with the following details:

- Compose** button
- Search all conversations** bar
- Inbox** tab selected (1,291 messages)
- From**: spitrackv1@gmail.com
- To**: (empty)
- Subject**: (no subject) **External**
- Message Content**: "Hello! Mohammad Nabuluzzaman Noley,
Your cluster 'Random Cluster' has been created. Now, you can perform your searches."
- Timestamp**: 11:37 PM (1 minute ago)
- Reply** and **Forward** buttons
- Left Sidebar**:
 - Starred
 - Snoozed
 - Sent
 - Drafts
 - Clutter
 - EEE154 Project
 - Events** (5 items)
 - Jobs - Internships
 - NSU S1 Fall 2019
 - NSU S2 Spring 2019
 - NSU S4 Fall 2020
 - NSU S5 Spring 2020
 - NSU S6 Summer 2020
 - NSU S7 Fall 2020
 - NSU S8 Spring 2021
 - NSU S9 Summer 2021
 - NSU S10 Fall 2021
 - CSE327
 - EEE111
- Bottom Navigation**: Chat, Spaces, Meet

Project Results (Web-App)

The screenshot shows a web application interface titled "SPITRACK". The left sidebar has a blue background with the following menu items: "SPITRACK", "Dashboard", "Clusters" (which is highlighted in blue), "Search", and "Logout". The main content area has a white background and displays three clusters:

- News**
 - Created On : 2022-02-26
 - Depth : 2
 - URL1 : <https://www.nytimes.com/international/section/politics>
 - URL2 : <https://www.daily-sun.com/online/world>
 - URL3 : <https://www.aljazeera.com/features/>
- Sports**
 - Created On : 2022-02-27
 - Depth : 2
 - URL1 : <https://www.aljazeera.com/news/>
 - URL2 : <https://www.daily-sun.com/online/sports>
 - URL3 : <https://edition.cnn.com/sport>
- Random Cluster**
 - Created On : 2022-02-28
 - Depth : 3
 - URL1 : <https://paperswithcode.com/paper/implicit-pdf-non-parametric-representation-of>
 - URL2 : <https://www.daily-sun.com/online/sports>
 - URL3 : https://en.wikipedia.org/wiki/Operating_system

Crawling Strategies: PDF, Word, Powerpoint, Text, Non-HTML Text

Project Results (Web-App)

The screenshot shows the phpMyAdmin interface running on a local host. The left sidebar displays a tree view of databases and tables, including 'spitrack' which contains 'clusters_cluster'. The main area shows the results of a SELECT query:

```
SELECT * FROM `clusters_cluster`
```

The results table has the following columns: id, username, cluster_name, url1, url2, url3, depth, pdf, word, powerpoint, text, non_html_text, date_of_creation, user_id. The data is as follows:

id	username	cluster_name	url1	url2	url3	depth	pdf	word	powerpoint	text	non_html_text	date_of_creation	user_id
27	mohdmeloy	News	https://www.nytimes.com/international/section/pol...	https://www.dailysun.com/online/world	https://www.aljazeera.com/features/	2	1	1	1	1	1	2022-02-26	1
42	mohdmeloy	Sports	https://www.aljazeera.com/news/	https://www.dailysun.com/online/sports	https://edition.cnn.com/sport	2	1	1	1	1	1	2022-02-27	1
48	mohdmeloy	Random Cluster	https://paperswithcode.com/paper/implicit-pdf-	https://www.dailysun.com/online/sports	https://en.wikipedia.org/wiki/Operating_system	3	1	0	0	0	0	2022-02-28	1

Below the results, there are sections for 'Query results operations' (Print, Copy to clipboard, Export, Display chart, Create view) and 'Bookmark this SQL query' (Label input, Let every user access this bookmark). At the bottom, there is a 'Console' section.

Project Results (Web-App)

The screenshot shows a browser window with the URL `localhost:127.0.0.1/spittrack`. The page title is "React App". The left sidebar of the phpMyAdmin interface lists databases: information_schema, mysql, performance_schema, phpmyadmin, spittrack, and test. The spittrack database is selected, and its tables are listed: New, auth_group, auth_group_permissions, auth_permission, auth_user, auth_user_groups, auth_user_user_permissions, clusters_cluster, clusters_crawleddata, django_admin_log, django_content_type, django_migrations, and django_session. The clusters_crawleddata table is selected and displayed in the main pane. The table has columns: id, username, date_of_creation, root_url, current_url, crawled_data_type, content, depth, cluster_id, and user_id. There are 25 rows of data, each showing a different URL from nytimes.com and its corresponding crawled data type (Non HTML). The content column contains snippets of news articles.

	id	username	date_of_creation	root_url	current_url	crawled_data_type	content	depth	cluster_id	user_id
1	222	mohdnneloy	2022-02-26	https://www.nytimes.com/international/section/pol...	https://www.nytimes.com/subscription/multiproduct/...	Non HTML	Unlimited access to all the journalism we offer. ...	2	27	1
2	223	mohdnneloy	2022-02-26	https://www.nytimes.com/international/section/pol...	https://www.nytimes.com/subscription/dg-cookie-pol...	Non HTML	List of Cookie Policies	2	27	1
3	224	mohdnneloy	2022-02-26	https://www.nytimes.com/international/section/pol...	https://www.nytimes.com/interactive/2022/world/eur...	Non HTML	Russia Attacks Ukraine live Updates: Troops Enter Ky...	2	27	1
4	225	mohdnneloy	2022-02-26	https://www.nytimes.com/international/section/pol...	https://www.nytimes.com/2022/02/25/us/politics/rus...	Non HTML	Russia Attacks Ukraine live Updates: Troops Enter Ky...	2	27	1
5	226	mohdnneloy	2022-02-26	https://www.nytimes.com/international/section/pol...	https://www.nytimes.com/2022/02/25/world/europe/kh...	Non HTML	Russia Attacks Ukraine live Updates: Troops Enter Ky...	2	27	1
6	227	mohdnneloy	2022-02-26	https://www.nytimes.com/international/section/pol...	https://www.nytimes.com/2022/02/25/world/europe/ze...	Non HTML	Russia Attacks Ukraine live Updates: Troops Enter Ky...	2	27	1
7	228	mohdnneloy	2022-02-26	https://www.nytimes.com/international/section/pol...	https://www.nytimes.com/2022/02/25/us/politics/bid...	Non HTML	Advertisement Continue reading the main story LIVE RU...	2	27	1
8	229	mohdnneloy	2022-02-26	https://www.nytimes.com/international/section/pol...	https://www.nytimes.com/2022/02/25/world/europe/uk...	Non HTML	Russia Attacks Ukraine live Updates: Troops Enter Ky...	2	27	1
9	230	mohdnneloy	2022-02-26	https://www.nytimes.com/international/section/pol...	https://www.nytimes.com/2022/02/25/world/empire-st...	Non HTML	Supreme Court Nomination Biden Picks Ketanji Brown ...	2	27	1
10	231	mohdnneloy	2022-02-26	https://www.nytimes.com/international/section/pol...	https://www.nytimes.com/2022/02/25/us/politics/bid...	Non HTML	Supreme Court Nomination Biden Picks Ketanji Brown ...	2	27	1
11	232	mohdnneloy	2022-02-26	https://www.nytimes.com/international/section/pol...	https://www.nytimes.com/2022/02/25/us/politics/ket...	Non HTML	Supreme Court Nomination Biden Picks Ketanji Brown ...	2	27	1
12	233	mohdnneloy	2022-02-26	https://www.nytimes.com/international/section/pol...	https://www.nytimes.com/2022/02/25/us/supreme-cour...	Non HTML	Advertisement Continue reading the main story DAILY ...	2	27	1
13	234	mohdnneloy	2022-02-26	https://www.nytimes.com/international/section/pol...	https://www.nytimes.com/2022/02/25/us/politics/ket...	Non HTML	Advertisement Continue reading the main story Suppor...	2	27	1
14	235	mohdnneloy	2022-02-26	https://www.nytimes.com/international/section/pol...	https://www.nytimes.com/2022/02/25/business/stock...	Non HTML	The Coronavirus Pandemic Covid-19 UpdatesCoronaviru...	2	27	1
15	236	mohdnneloy	2022-02-26	https://www.nytimes.com/international/section/pol...	https://www.nytimes.com/2022/02/25/business/econom...	Non HTML	Advertisement Continue reading the main story Suppor...	2	27	1
16	237	mohdnneloy	2022-02-26	https://www.nytimes.com/international/section/pol...	https://www.nytimes.com/2022/02/25/business/ukrain...	Non HTML	The Coronavirus Pandemic Advertisement Coronaviru...	2	27	1
17							The Coronavirus Pandemic			

Project Results (Web-App)

The screenshot shows a web application interface titled "Search". The left sidebar, titled "SPITRACK", contains navigation links: Dashboard, Clusters, Search (which is selected and highlighted in blue), and Logout. The main content area has a search bar with the query "hockey" and a "Search" button. Below the search bar, there is a section titled "Select Cluster(s)" with checkboxes for News, Sports, Random Cluster, and CBS News, where Sports and CBS News are checked.

Hockey players skating for

Depth : 1
Crawling Strategies : Non HTML

Content : Not in control of situation."Ovechkin is among many Russian ice hockey players skating for American teams, from the junior levels to the pros. Read MoreThe 36-year-old former first-round draft pick has

Root URL : <https://edition.cnn.com/sport>
Content URL : <https://www.cnn.com/2022/02/25/sport/ovechkin-nhl-war/index.html>

Hockey player Jarome Iginla

Depth : 1
Crawling Strategies : Non HTML

Content : Victory in the two-woman bobsled with Jill Bakken in 2002. Hockey player Jarome Iginla became the first Black man to win gold at the same Games when Canada triumphed over

Root URL : <https://edition.cnn.com/sport>
Content URL : <https://www.cnn.com/2022/02/21/sport/winter-olympics-elite-wealthy-intl-spt/index.html>

Hockey players celebrate after

Depth : 1

Hockey star's unconventional path

Depth : 1

Project Results (Mobile-App)

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The project is named "My Application". It contains an "app" module with "AndroidManifest.xml", "activity_main.xml", "activity_register.xml", "RegisterActivity.java", "ApiClient.java", "UserService.java", "RegisterRequest.java", and "RegisterResponse.java".
- Code Editor:** The "RegisterActivity.java" file is open, showing Java code for handling user registration. It includes methods for handling button clicks and registering users via an API client.
- Preview:** A virtual device (Pixel XL API 30) displays the mobile application's home screen, which has a blue geometric background and the word "Spitrack" in white.
- Toolbars and Panels:** The standard Android Studio toolbars and panels are visible, including the Project, Tools, and Device Manager.
- Bottom Status Bar:** The status bar shows "Launch succeeded" and "Android Studio Bumblebee | 2021.1.1 Patch 2 available".
- Bottom Navigation:** The navigation bar includes icons for TODO, Problems, Git, Terminal, Logcat, Build, Profiler, Run, and App Inspection.

Project Results (Mobile-App)

The screenshot displays the Android Studio interface with the project structure and code editor on the left, and the emulator showing the app's UI on the right.

Project Structure:

- app
- manifests
- java
- com.example.myapplication

 - ApiClient
 - LoginActivity
 - LoginRequest
 - LoginResponse
 - MainActivity
 - RegisterActivity
 - RegisterRequest
 - RegisterResponse
 - UserService

- com.example.myapplication (androidTest)
- com.example.myapplication (test)
- java (generated)
- res
- drawable
- layout
- mipmap
- values
- res (generated)
- Gradle Scripts
- build.gradle (Project: My_Application)
- build.gradle (Module: My_Application.app)
- gradle-wrapper.properties (Gradle Version)
- gradle-proguard-rules.pro (ProGuard Rules for My_Application.app)
- gradle.properties (Project Properties)

Code Editor (RegisterActivity.java):

```
public void onClick(View view) {
    startActivity(new Intent(getApplicationContext(), LoginActivity.this));
}

public void registerUser(RegisterRequest registerRequest) {
    Call<RegisterResponse> registerResponseCall = ApiClient.getService().registerUser(registerRequest);
    registerResponseCall.enqueue(new Callback<RegisterResponse>() {
        @Override
        public void onResponse(Call<RegisterResponse> call, Response<RegisterResponse> response) {
            if (response.isSuccessful()) {
                String message = "User has been Registered!";
                Toast.makeText(context, RegisterActivity.this, message, Toast.LENGTH_SHORT).show();
                startActivity(new Intent(getApplicationContext(), LoginActivity.this));
                finish();
            } else {
                String message = "Username already taken or Password is not strong enough!";
                Toast.makeText(context, RegisterActivity.this, message, Toast.LENGTH_SHORT).show();
            }
        }
    });
}
```

Emulator Screen (Register Activity):

The emulator shows the "Register" screen of the mobile application. It features a light blue header with the word "Register". Below it is a form with five input fields and a "REGISTER" button. The input fields are:

- Md Abdul
- Wahab
- abdulwahab
- md.abdul@northsouth.edu
- shimu

Below the inputs is a "REGISTER" button. At the bottom of the screen, there is a link "Already Have an Account?"

Project Results (Mobile-App)

The screenshot shows the Android Studio interface with the following details:

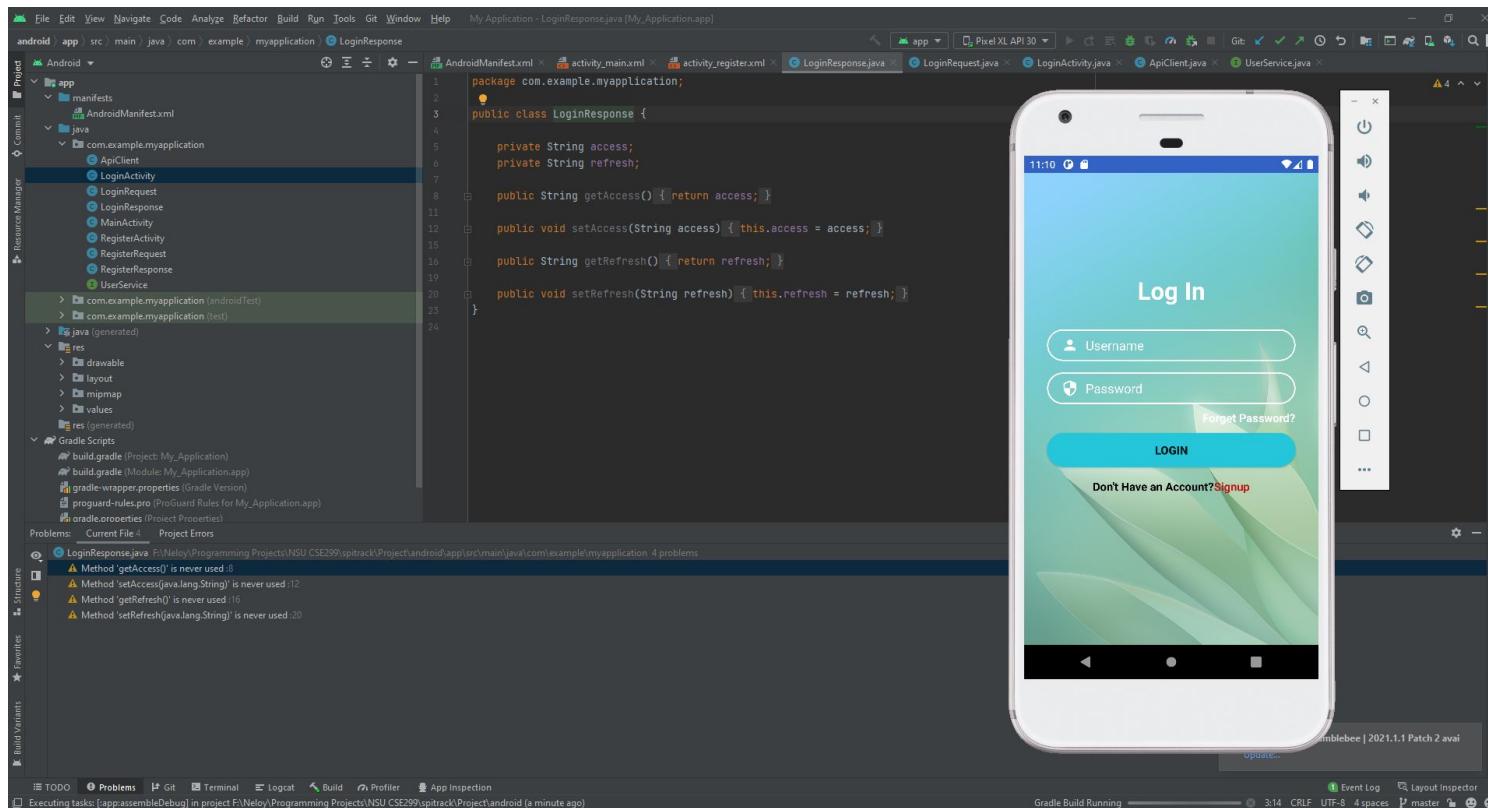
- Project Structure:** The project is named "My Application". It contains an "app" module with "src", "AndroidManifest.xml", "java", "res", and "gradle" directories.
- Code Editor:** The code for `RegisterActivity.java` is displayed. The code handles user registration, making a network call via `ApiClient` to a service that returns a `RegisterResponse`. It checks if the response is successful and displays a toast message accordingly.
- Run View:** A virtual device (Pixel XL API 30) is running the application. The screen shows a "Log In" page with fields for "Username" and "Password", a "Forget Password?" link, and a "LOGIN" button. Below the button is a "Don't Have an Account? Signup" link. A success message "User has been Registered!" is displayed at the bottom.
- Bottom Bar:** The Android Studio navigation bar includes "Launch succeeded", "Build Variants", "Structure", "File", "Edit", "View", "Navigate", "Code", "Analyze", "Refactor", "Build", "Run", "Tools", "Git", "Window", and "Help".
- Bottom Status Bar:** Shows "Launch succeeded (6 minutes ago)", "Event Log", "Layout Inspector", "71:31 (22 chars) CRLF", "UTF-8 4 spaces", "master", and other developer tools.

Project Results (Mobile-App)

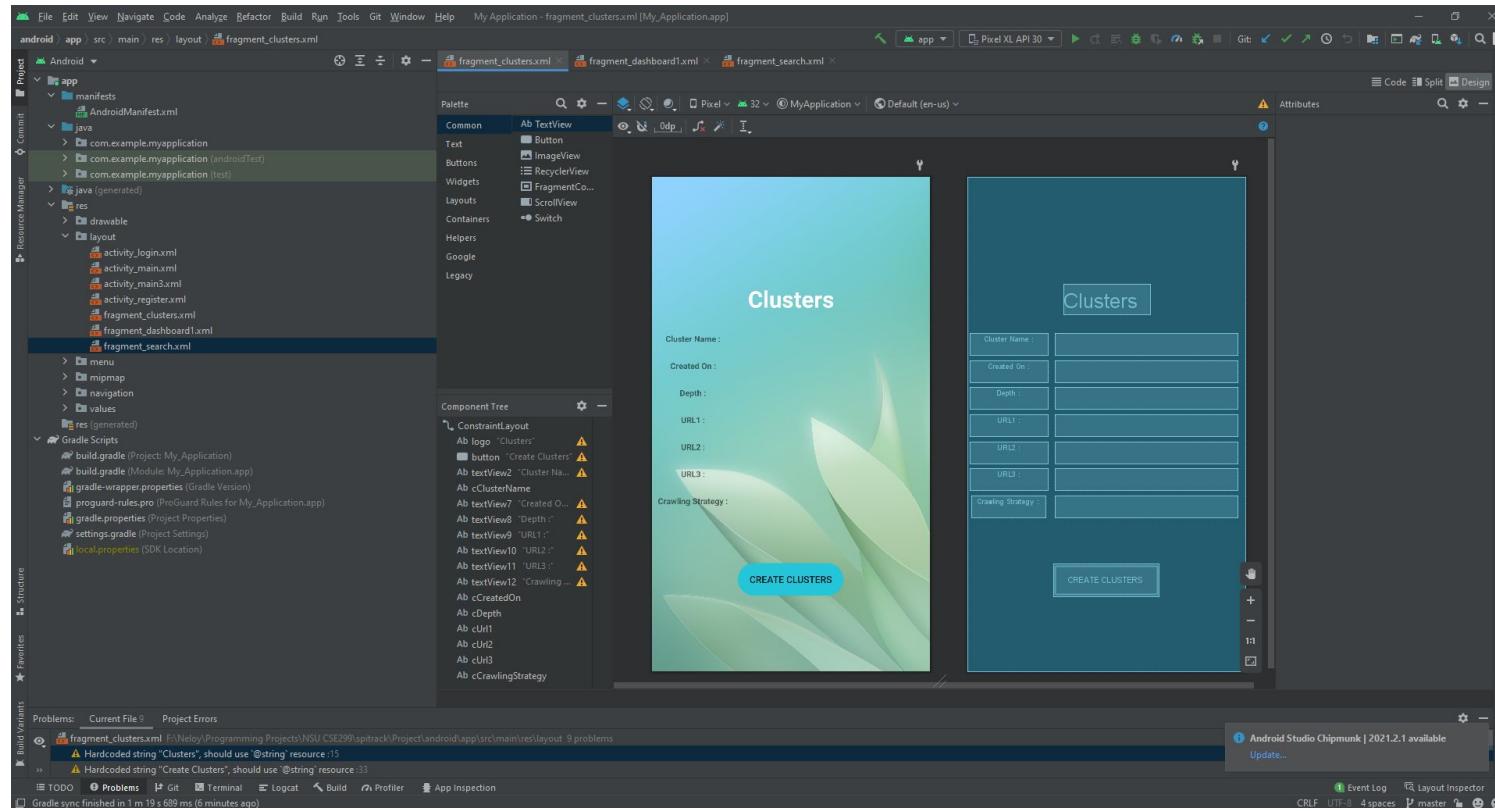
The screenshot shows the PhpMyAdmin interface for a MySQL database named 'spitrack'. The left sidebar displays the database schema with tables like information_schema, mysql, performance_schema, phpmyadmin, spitrack, and test. The main area is focused on the 'auth_user' table under the 'spitrack' database. The table structure includes columns: id, password, last_login, is_superuser, username, first_name, last_name, email, is_staff, is_active, and date_joined. The data grid shows five rows of user information, each with edit, copy, and delete options. The bottom of the screen features a toolbar with Print, Copy to clipboard, Export, Display chart, Create view, and a Console tab.

	id	password	last_login	is_superuser	username	first_name	last_name	email	is_staff	is_active	date_joined	
<input type="checkbox"/>	1	pbkdf2_sha256\$260000\$8RwKSK919nrOj0Cxd0BH\$3xasN...	NULL	0	mohdnneloy	Mohammad	Nabiluzzaman	Neloy	mohammed.nabiluzzaman@northsouth.edu	0	1	2022-02-17 15:34:18.055417
<input type="checkbox"/>	2	pbkdf2_sha256\$260000\$3kKvvsLLANpvcIKMjpHnb\$1pMAA4...	NULL	0	rafidahmed	Rafid	Ahmed	rafid.ahmed@northsouth.edu	0	1	2022-02-27 14:41:02.532083	
<input type="checkbox"/>	7	pbkdf2_sha256\$260000\$0xf2zjS3dGtly19wN7qy\$4d2jUT...	NULL	1	mohdnneloy	Mohd N	Neloy	neloy11@gmail.com	1	1	2022-03-19 06:38:07.676176	
<input type="checkbox"/>	10	pbkdf2_sha256\$260000\$GgB5GFALqionwxcC9wQR\$unwR9...	NULL	1	tammamhaque	Tamman	Haque	tammam.haque@northsouth.edu	1	1	2022-03-20 12:36:17.038925	
<input type="checkbox"/>	16	pbkdf2_sha256\$260000\$0EXIYLNuPESjQHxkT3ysj\$5\$Kh8wT...	NULL	0	abdulwahab	Md Abdul	Wahab	md.abdul@northsouth.edu	0	1	2022-03-28 15:02:59.85004	

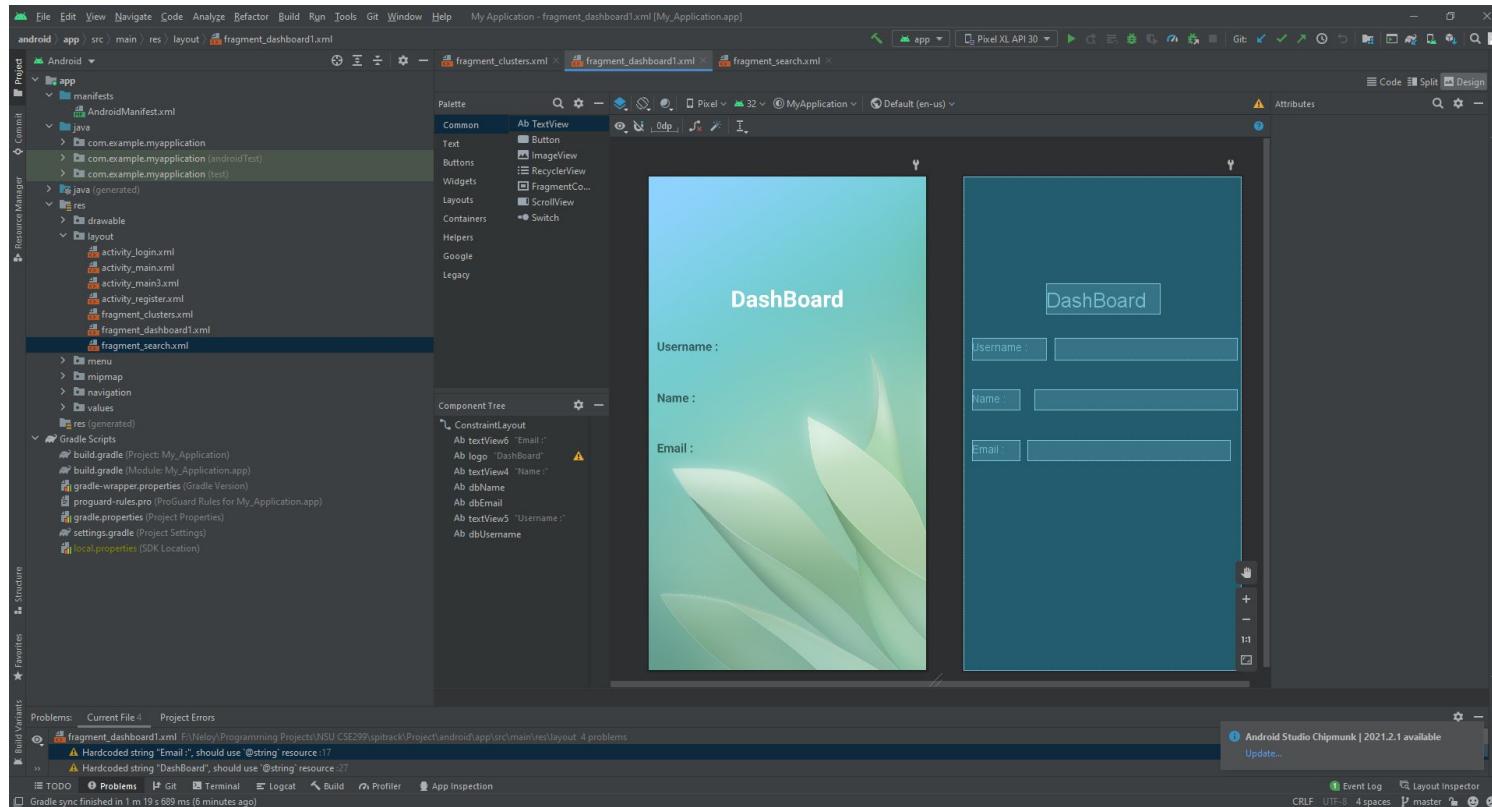
Project Results (Mobile-App)



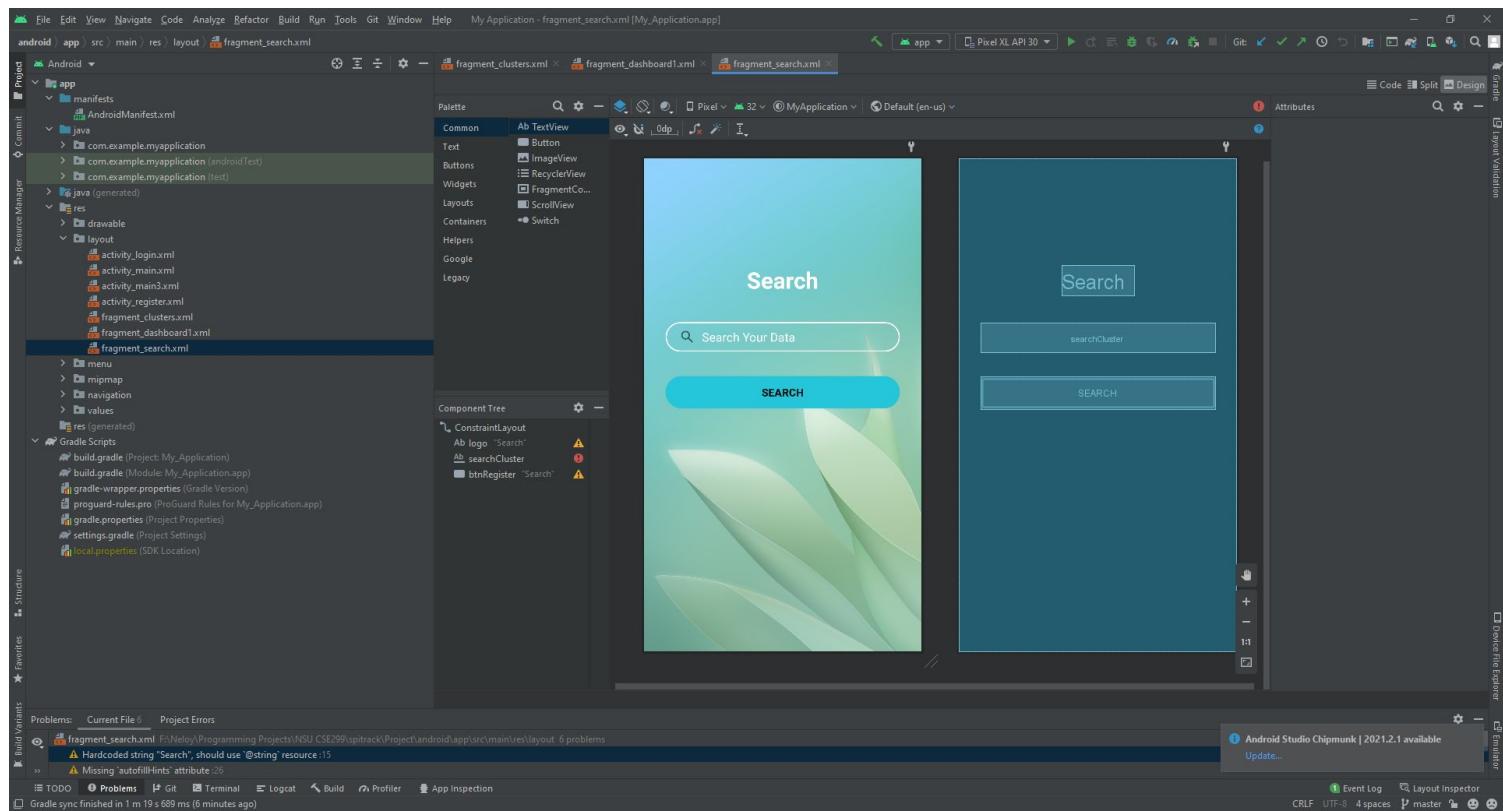
Project Results (Mobile-App)



Project Results (Mobile-App)



Project Results (Mobile-App)



Trello Board (Neloy) - 1

The screenshot shows a Trello board titled "Web Application" for the "CSE299 Junior Project". The board has the following structure:

- To Do:**
 - Voice Search
 - Create Django and React Project
 - Connect XAMPP Web Server With Django
 - User Sign Up
- Next Week's Work:**
 - Sign In With Google
 - Control Scrapping Strategies
- Doing:**
 - Bug Checks
 - Create Django and React Project
- Done:**
 - User Sign Up
 - User Login System
 - User Dashboard
 - Clusters View
 - Cluster Creation
 - Admin Login System
 - Search
 - Admin Login System

The sidebar on the left shows the following options:

- Boards
- Members
- Settings
- Workspace views
 - Table
 - Calendar
- Your boards
 - Android Application
 - Web Application

A purple "Try Premium free" button is located at the bottom left of the sidebar.

Trello Board (Neloy) - 2

The screenshot shows a Trello board titled "Web Application". The board has four lists: "To Do", "Next Week's Work", "Doing", and "Done".

- To Do:**
 - Voice Search
 - Sign In With Google
- Next Week's Work:**
 - Control Scrapping Strategies
 - Bug Checks
- Doing:**
 - Bug Checks
- Done:**
 - Clusters View
 - Cluster Creation
 - Search
 - Admin Login System
 - Add New Admin Feature
 - Hosting Front-End (React) to Firebase
 - Real Time Database Connection with Back-End (Django) in AWS RDS MySQL Database
 - Hosting Back-End (Django) to Heroku

The sidebar on the left shows other boards in the workspace, including "Android Application" and "Web Application".

Trello Board (Rafid)

The screenshot shows a Trello board titled "Android Application" for the "CSE299 Junior Project" workspace. The board has four lists:

- To Do**: Voice Search, Sign In With Google, Control Scapping Strategies, Voice Search.
- Next Weeks Work**: Add New Admin, Admin Log In.
- Doing**: Cluster Creation, Clusters View.
- Done**: User Log In, User Sign Up, Dashboard, Search, Calling API, User Dashboard, Voice Search.

A yellow tent is visible at the bottom of the board. The left sidebar shows the workspace navigation and lists like Boards, Members, and Settings. The top right has search and filter options.

Slack Conversation

junior-design-project

Tuesday, April 5th

Rafid Ahmed 1831395642 11:51 PM
https://www.youtube.com/watch?v=ixRXEoGAEZM&t=70s&ab_channel=CodingZest
YouTube | CodingZest
Modern Dashboard UI Design Android Studio Tutorial | Cardview Android Studio | Grid layout Android



Sunday, April 10th

Mohammad Nabiluzzaman Neloy 1831251042 12:44 AM
<https://www.youtube.com/watch?v=KX2YYO1Kwws&t=309s>
YouTube | codejay

B I \$ ⚡ ━ ━ ━ ━ ━ ━ ━ ━ ━

Send a message to @junior-design-project

+ | 🎉 | 🎧 | 😊 | @ Aa

Slack Conversation

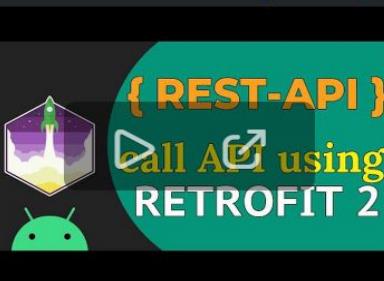
junior-design-project ▾ 2

Rafid Ahmed 1831395642 9:43 PM

https://www.youtube.com/watch?v=UCVNP2_bwOw

YouTube Md Jamal

How to call REST API in Android using RETROFIT | Retrofit Android Tutorial



<https://www.youtube.com/watch?v=fuyozXGz-mM>

YouTube Md Jamal

How to call REST API in Android | Volley Android | Call REST API using Volley



B I ⌂ | ⌂ | ⌂ | ⌂ | ⌂ | ⌂ | ⌂ | ⌂ | ⌂

Send a message to junior-design-project

+ | □ | ⌂ | ⌂ | ⌂ | ⌂ | ⌂ | ⌂ | ⌂

▶ | ↴

Slack Conversation

junior-design-project ▾

1 Pinned + Add a bookmark

Create a Navigation Drawer In Android Studio ▾



Today New

Mohammad Nabiluzzaman Neloy 1831251042 10:16 PM

Meeting at 10:30, Please be there!

We have to work on the final ppt!

I am updating the codes, there are some bugs that i need to fix

Rafid Ahmed 1831395642 11:15 PM

Glad you are there for me.

I'm having issues in android login part. I am also working on bug fix.

B I ⌂ | ⌂ | ⌂ | ⌂ | ⌂ | ⌂ | ⌂

Send a message to @junior-design-project

+ | ☰ | ☰ | ☰ | ☰ | ☰ | ☰