CMPE 352

Milestone Report 2

Group 9

Ahmed Bera Pay
Ahmet Abdullah Susuz
Ali Mert Geben
Arda Arslan
Hakan Emre Aktaş
Leyla Yayladere
Mehmet Süzer
Rafet Oğuz Pançuk
Ömer Şükrü Uyduran
Zülal Molla

1. Executive Summary

1.1 Summary of project

In this project, we developed a web application which uses third party APIs. We chose APIs which are about articles and journals because we wanted our web application to be related to our main project which is a scientific collaboration platform. In our web application, users can create an account using their ORCID ID. After creating an account on the sign-up page, they are redirected to the sign-in page. Those who sign in can create paper lists, add papers to their paper lists, follow other users, and follow other lists. Searching papers doesn't require authorization, meaning that those who don't have an account can search papers using our API. Having an account allows users to search and display papers on UI. A user can select which API to use while searching papers. For instance, a user may prefer DOAJ over other APIs. However, all responses are in the same format regardless of the API chosen.

1.2 Overall status

Following the initial milestone report, which focused on gathering requirements and designing the project, we began developing a basic web application called practice-app. The primary goal of this project was to gain experience in the basics of web development such as front-end and back-end design, utilizing third-party APIs, and deployment. Our first step was to research RESTful APIs related to our main project, which is a collaborative science platform, offering access to several paper and article databases. Next, we implemented GET methods that utilized these third-party APIs, which provided us with an opportunity to learn about using them in our web application. We also gained experience in managing and securing API keys. After implementing the GET methods, we held a team meeting to discuss the POST methods we would implement in our web application. We distributed these methods among ourselves and considered the UI design phase of development. For the frontend and backend, we used Django, using its view and template models for the frontend, and its model class and sqlite database for the backend. After implementing the GET and POST methods, along with the UI design, we dockerized our project-app web application and deployed it to an Amazon AWS EC2 instance.

In short, the web application that we have developed is ready to use and functional in terms of searching papers, searching users, adding papers to lists etc. Its GET and POST methods are working, and they are documented properly.

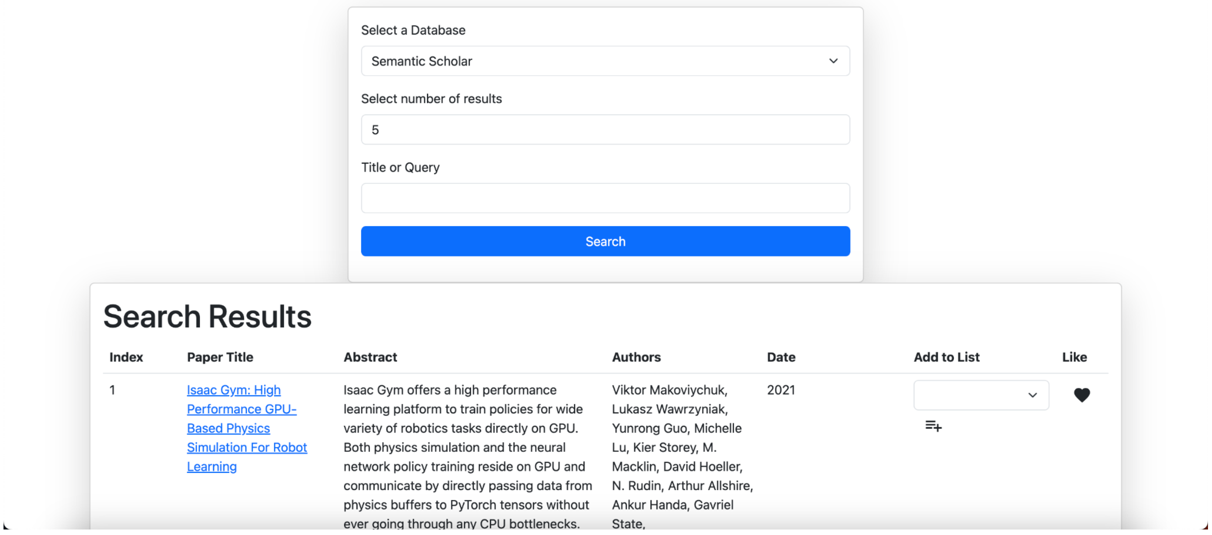
1.3 Description of the basic functionality of your project (can include screenshots)

A user can sign up with his/her ORCID ID. Once users successfully sign up, they are redirected to a sign in page. An authorized user can search papers using the third party APIs that we have utilized. A list of papers coming from the database of the selected third party API is given when the search paper is used. In order to search papers, one doesn't have to be signed in. He/she can use the API we have developed (/api/<third-party-api>/). The API will return a 4xx response if the parameters are not given correctly.

A signed in user can search other users and send a follow request to them. He/she also can create a list in which favorite papers are listed. Users can follow other users' paper lists. Each user has a profile page. Users can add interests in their profile page.

All GET and POST methods are documented in /api/swagger-ui/ page. Those who want to learn how to use our API can check the documentation.

Search Paper



Select a Database: Semantic Scholar

Select number of results: 5

Title or Query:

Search Results

Index	Paper Title	Abstract	Authors	Date	Add to List	Like
1	Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning	Isaac Gym offers a high performance learning platform to train policies for wide variety of robotics tasks directly on GPU. Both physics simulation and the neural network policy training reside on GPU and communicate by directly passing data from physics buffers to PyTorch tensors without ever going through any CPU bottlenecks.	Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, M. Macklin, David Hoeller, N. Rudin, Arthur Allshire, Ankur Handa, Gavriel State,	2021		

1.3.1 URL of practice app code and the tag

Code:
https://github.com/bounswe/bounswe2023group9/tree/main/practice_app

Tag:
https://github.com/bounswe/bounswe2023group9/releases/tag/Group9-Practice-App-Release-v.0.1

1.3.2 URL of your application that was deployed with docker

http://35.171.9.117:8000

1.3.3 Any information we may need to test your application

ORCID ID’s should be retrieved from orcid.org.

CORE API has a rate limit and should be tested accordingly.

1.3.4 Instructions for building the application with docker.

To build the application using docker, you should type:

- docker build –tag <your-tag> .

in the practice_app folder.

Our image also exists in Docker Hub and you can pull it using;

- docker pull hakanaktas0/practice-app

To run the app in its full functionality some environment variables must be set before running the app. These are:

1. serp_api_key
2. core_api_key
3. zenodo_api_key
4. SECRET_KEY

SECRET_KEY can be any string, but rest of them should be acquired from third party providers SerpApi , core, zenodo.
To run : docker run <your-tag>

1.3.5 API URL (the URL should take us to a documented API)

We created the documentation using swagger. You can build it using the json file in practice_app/api/static/swagger.json. We also served it in our app which you can access in /api/swagger-ui/
Remote link: http://35.171.9.117:8000/api/swagger-ui/

1.3.6 Links to meeting notes related to practice app

1- Meeting#9
2- Meeting#10
2- Meeting#11
3- Meeting#12
4- Meeting#13
5- Meeting#14

1.3.7 Lessons learned: evaluation of tools and processes utilized to manage the team project.

1.3.7.1 Evaluation of Tools

Discord

We mainly stayed in touch with our team members through Discord. The progress of the web application that we have developed is monitored by team members on Discord. Different chat channels were created to collect the relevant messages, links and documents together. For instance, those who worked on frontend development discussed the issues that they encountered in the “frontend” channel. We collected the development environment and package versions of each user in a channel to determine the requirements of the practice app we have developed.

For most of us, developing a web application was a thing with which we have no experience. Therefore, we needed lots of resources about Django, frontend development, and API utilization. Those who found useful resources shared the links through Discord.

Whatsapp

Whatsapp is mainly used to access a team member instantly. The urgent things were announced in Whatsapp, since some team members don't check their Discord messages regularly. Meeting locations and dates, issues that can be closed, and pull requests needing reviews were told to other team members via Whatsapp. In addition, those who needed help and were confused about a part of the project asked for assistance from other team members using Whatsapp.

Postman

We used Postman to document and test our GET and POST methods. It is easy to use and has a nice user interface. We created a Postman account for the team since creating a workspace for more than 3 users is a premium feature. Every team member used the same account to document their API methods. We found Postman useful in the sense that it makes it easier to test the functionality of GET and POST methods. Those who need to use methods which were written by another team member can easily learn the format of URLs and responses coming from the API.

Github

Github is the platform where we collected the code of the web application that we developed. First, an initial push is made to the main branch so that every team member can work on the project with similar files. Then, once a team member completed his/her implementation, he/she created a branch and made a pull request. Codes written by team members were checked by reviewers. In most of the pull requests, 2 or more reviewers were assigned to eliminate errors as much as possible. Additionally, the meeting notes were listed in the wiki page of our Github repository for those who missed the meeting or wanted to take a look again.

We created a milestone with a due date 12.05.2023 and linked our issues to the milestone to keep the progress of the practice app. Issues helped us to distribute the workload among the team members and to inform other team members about the progress of tasks.

Swagger UI

Swagger UI is an open source tool that allows users to create interactive documentations for APIs. Although we used Postman to document our API methods, we decided to add a swagger page to our app to increase usability of our API. After all team members added and documented their API methods to Postman, Hakan used Swagger

UI to create an interactive swagger documentation. After creating the JSON file, Hakan added a swagger page to our app using Django’s built-in methods.

VSCode

Most of us used VsCode to implement the project. It is easy to use and has a basic user interface. It supports many programming languages. Autofill feature of VSCode leads to a decrease in the development process. The most helpful feature of the VSCode for our project was the git feature. One can easily create branches and push them, which allows us to combine our work into a single code in our Github repository. As a result, we found VSCode as an excellent IDE for our project.

1.3.7.2 Evaluation of Process

The implementation of a web application was a challenging project for us. We learned many things throughout the process. API implementation using Django, documentation of an API, implementation of GET and POST methods, and writing test cases, and using Postman and Swagger are some of the skills that we have learned during the development of the project. We believe that we did our best in a limited time.

We tried to distribute the tasks as evenly as possible. Every team member implemented a GET and a POST method for our web application. Also, every team member but two contributed to frontend of our project.

During the development of the project, we encountered many challenges. Since most of them are common, we dealt with these problems together to speed up the process. We were helpful and friendly in terms of collaborative work.

In short, we learnt backend and frontend development, version management, API documentation, dockerization, and deployment, while we improved ourselves in issue management and collaborative code development.

1.3.7.3 Challenges you met as a group

Most of us were inexperienced in backend and frontend development. The web application development was a thing that we haven’t done before. We selected Django for our project, and we had a lot to learn about it such as database management, get and post methods, and creation of interactive user interface. Some of the team members had difficulties about API keys since the APIs that they will utilize need authentication. Furthermore, we needed to determine a standard for the response for our API since different APIs return responses in different formats. For instance, some APIs that we utilized have only year as “date”, while some of them have a full date.

Another challenge that we encountered as a group was the implementation of test cases. Again, we had to learn the details of the implementation of test cases for our GET and POST methods since we hadn’t done this before. Additionally, the database of Django is different from what we have learned. It supports “ManyToManyField”, which contains lists of an instance. Accessing and modifying the “ManyToManyField”s was a challenge for us.

Lastly, some of the APIs that we utilized had limited use. Therefore, implementation and testing of the methods were cumbersome.

2. Team Member Contributions

Hakan Emre Aktaş
Important Issues:

- Third Party API research #99
- İnitial commit of our project #108
- ImplementedGET method that utilizes a third party API #111
- Setting up GA for automated testing #112
- Opened the main issue for POST methods #140
- Corrected the get methods to achieve a uniform interface between GET methods #138
- Implemented a POST method that allows the users to save the search results #141
- Implemented the followers and following pages for the UI #167
- Separated the header from rest of the pages to ease the development of UI #168
- Fixed Author fields of the GET method responses to achieve uniform response across all GET methods #188

The name, route, description of utilized third party URIs
I utilized a third party API called SerpAPI which is mainly used for getting results from search engines. I used it to get results from Google Scholar. I picked it because Google Scholar doesn't have a native API. Users need an API token to use the API. API is normally paid, but for my implementation I used the trial which provides 100 free API requests.
Their website: https://serpapi.com/
Route to API: https://serpapi.com/search.json
Example:
'https://serpapi.com/search.json?engine=google_scholar&q=sad&hl=en&num=3&api_key=<api-key>

The name, route, description of the created API functions

- **GET method : Google Scholar:**
  This method allows users to send search queries to Google Scholar. This API method utilizes SerpAPI. Function takes 2 parameters called 'title' and 'rows'. title is the search word and rows is the number of results returned. If rows is not given, 3 results are returned.
  If the search is successful a response with status code 200 is returned. If the title is missing 404 is returned. If third party api returns a response with status code other than 200 or 404, a response with status code 500 is returned.
  route: /api/google-scholar/
- **POST method : Post papers:**

This method allows users to send a search query to any one of the databases we utilized and save the results of that search to the database. This function takes 3 parameters named db, title and rows in the request body as form-data. if rows is not given 3 papers are saved. This method also requires authentication which can either be achieved by calling /log-in/ first or adding valid username and password fields to the header of the HttpRequest.
If the save operation is successful the function returns status code 200. If credentials are invalid it returns status code 401 and If the credentials are missing it returns status code 407. It can also redirect the response received from the GET method it called if the response doesn't have status code 200.
route: /api/post-papers/

Detailed information about the route, parameters and responses of the api functions can be found in the swagger page of our app which can be accessed in /api/swagger-ui/

Description of unit tests
In the unit tests of the GET method I utilized, I mainly checked 2 things: status code and content of the response. For this I both checked 200 and 404 responses.
For the 200 cases, I first send a valid request through my API function. I then checked whether all the fields of the json response were there or not and the status code of the responses. Then I send another request with the same parameters directly to the third-party api. After that I checked the content of the responses.
For the 404 cases, I sent invalid requests to the API and checked the status codes.
A 404 test : self.assertEquals(self.c.get("/api/google-scholar/?").status_code, 404)
A 200 test:
response = self.c.get("/api/google-scholar/?title=test&rows=3")
self.assertEquals(response.status_code, 200)

In the unit tests of the POST method, I again checked for 200 and 4xx responses. For the 200 I send valid requests to all databases we utilized respectively and checked the status codes of the responses and also checked whether the database is populated with those entries or not. For the 4xx cases, I checked all responses with 4xx status codes namely 401, 404 and 407. For 401 I put invalid credentials to the header of the request. For 404 I sent requests with invalid parameters (etc. missing title, missing db) and for 407 I sent a request with missing credentials. After each one of these requests I checked the status codes of the responses.
A 200 test:
response = self.client.post("/api/post-papers/", {'db': 'google-scholar', 'title': 'sad', 'rows': 6}, headers={'username': "0009-0005-5924-0000", 'password': 'strongpassword'})
self.assertEquals(response.status_code, 200)
self.assertEquals(len(models.Paper.objects.filter(source='google_scholar')), 6)

A 401 test:

```
response = self.client.post("/api/post-papers/", {'db': 'zenodo', 'title': 'sad', 'rows': 5},
headers={'username': "0009-0005-5924-000", 'password': 'strongpassword'})
self.assertEquals(response.status_code,401)
self.assertEquals(json.loads(response.content.decode("UTF-8")), {'status' : 'user
credentials are incorrect.'})
```

A 407 test:

```
response = self.client.post("/api/post-papers/", {'db': 'zenodo', 'title': '', 'rows': 3},
headers={ 'password': 'strongpassword'}) self.assertEquals(response.status_code, 407)
self.assertEquals(json.loads(response.content.decode("UTF-8")),{'status': 'username and
password fields can not be empty'})
```

Sample calls:
API I Utilized:

- request: https://serpapi.com/search.html?engine=google_scholar&q=corona&hl=en&num=2&api_key=<api-key>
- response : (status code = 200) (json)
{ "search_metadata": { "id": "645e8420216a9d66b10abd58", "status": "Success", "json_endpoint": "https://serpapi.com/searches/620be35afe1bf293/645e8420216a9d66b10abd58.json", "created_at": "2023-05-12 18:23:28 UTC", "processed_at": "2023-05-12 18:23:28 UTC", "google_scholar_url": "https://scholar.google.com/scholar?q=corona&hl=en&num=2", "raw_html_file": "https://serpapi.com/searches/620be35afe1bf293/645e8420216a9d66b10abd58.html", "total_time_taken": 1.11 }, "search_parameters": { "engine": "google_scholar", "q": "corona", "hl": "en", "num": "2" }, "search_information": { "organic_results_state": "Results for exact spelling", "total_results": 2550000, "time_taken_displayed": 0.04, "query_displayed": "corona" }, "organic_results": [{ "position": 0, "title": "Corona discharge processes", "result_id": "d8_j3yqNqHgJ", "link": "https://ieeexplore.ieee.org/abstract/document/125038/", "snippet": "Applications of corona discharge induced plasmas and unipolar ions are reviewed. Corona process applications emphasize one of two aspects of the discharge: the ions produced or …", "publication_info": { "summary": "JS Chang, PA Lawless… - IEEE Transactions on …, 1991 - ieeexplore.ieee.org" }, "resources": [{ "title": "psu.edu", "file_format": "PDF", "link": "https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=148aa1e0084ba28ffcc23b32417d5a36afc59902" }], "inline_links": { "serpapi_cite_link": "https://serpapi.com/search.json?engine=google_scholar_cite&q=d8_j3yqNqHgJ", "cited_by": { "total": 1295, "link": "https://scholar.google.com/scholar?cites=8694354295923134327&as_sdt=2005&sciodt=0,5&hl=en&num=2", "cites_id": "8694354295923134327", "serpapi_scholar_link": "https://serpapi.com/search.json?as_sdt=2005&cites=8694354295923134327&e

ngine=google_scholar&hl=en&num=2" }, "related_pages_link":
"https://scholar.google.com/scholar?q=related:d8_j3yqNqHgJ:scholar.google.co
m/&scioq=corona&hl=en&num=2&as_sdt=0,5", "serpapi_related_pages_link":
"https://serpapi.com/search.json?as_sdt=0%2C5&engine=google_scholar&hl=en
&num=2&q=related%3Ad8_j3yqNqHgJ%3Ascholar.google.com%2F", "versions": {
"total": 13, "link":
"https://scholar.google.com/scholar?cluster=8694354295923134327&hl=en&nu
m=2&as_sdt=0,5", "cluster_id": "8694354295923134327", "serpapi_scholar_link":
"https://serpapi.com/search.json?as_sdt=0%2C5&cluster=869435429592313432
7&engine=google_scholar&hl=en&num=2" } } }, { "position": 1, "title": "Solar activity
and the corona", "result_id": "7P5fKrLoHAcJ", "link":
"https://link.springer.com/article/10.1007/BF00146338", "snippet": "… the corona
as an electrically highlyconducting atmosphere. The freezing of magnetic flux
into the huge volume of the corona … products of activity in the corona - flares
and coronal mass …", "publication_info": { "summary": "BC Low - Solar Physics,
1996 - Springer", "authors": [{ "name": "BC Low", "link":
"https://scholar.google.com/citations?user=Eh90_ecAAAAJ&hl=en&num=2&oi=sr
a", "serpapi_scholar_link":
"https://serpapi.com/search.json?author_id=Eh90_ecAAAAJ&engine=google_sch
olar_author&hl=en", "author_id": "Eh90_ecAAAAJ" }] }, "resources": [{ "title":
"harvard.edu", "file_format": "PDF", "link":
"https://adsabs.harvard.edu/pdf/1996SoPh..167..217L" }], "inline_links": {
"serpapi_cite_link":
"https://serpapi.com/search.json?engine=google_scholar_cite&q=7P5fKrLoHAcJ
", "cited_by": { "total": 605, "link":
"https://scholar.google.com/scholar?cites=512540309526150892&as_sdt=2005
&sciodt=0,5&hl=en&num=2", "cites_id": "512540309526150892",
"serpapi_scholar_link":
"https://serpapi.com/search.json?as_sdt=2005&cites=512540309526150892&en
gine=google_scholar&hl=en&num=2" }, "related_pages_link":
"https://scholar.google.com/scholar?q=related:7P5fKrLoHAcJ:scholar.google.co
m/&scioq=corona&hl=en&num=2&as_sdt=0,5", "serpapi_related_pages_link":
"https://serpapi.com/search.json?as_sdt=0%2C5&engine=google_scholar&hl=en
&num=2&q=related%3A7P5fKrLoHAcJ%3Ascholar.google.com%2F", "versions": {
"total": 4, "link":
"https://scholar.google.com/scholar?cluster=512540309526150892&hl=en&num
=2&as_sdt=0,5", "cluster_id": "512540309526150892", "serpapi_scholar_link":
"https://serpapi.com/search.json?as_sdt=0%2C5&cluster=512540309526150892
&engine=google_scholar&hl=en&num=2" } } }], "pagination": { "current": 1, "next":
"https://scholar.google.com/scholar?start=2&q=corona&hl=en&num=2&as_sdt=0,
5", "other_pages": {
"https://scholar.google.com/scholar?start=2&q=corona&hl=en&num=2&as_sdt=0,
5",
"https://scholar.google.com/scholar?start=4&q=corona&hl=en&num=2&as_sdt=0,

5",
"https://scholar.google.com/scholar?start=6&q=corona&hl=en&num=2&as_sdt=0,5",
"https://scholar.google.com/scholar?start=8&q=corona&hl=en&num=2&as_sdt=0,5",
"https://scholar.google.com/scholar?start=10&q=corona&hl=en&num=2&as_sdt=0,5",
"https://scholar.google.com/scholar?start=12&q=corona&hl=en&num=2&as_sdt=0,5",
"https://scholar.google.com/scholar?start=14&q=corona&hl=en&num=2&as_sdt=0,5",
"https://scholar.google.com/scholar?start=16&q=corona&hl=en&num=2&as_sdt=0,5",
"https://scholar.google.com/scholar?start=18&q=corona&hl=en&num=2&as_sdt=0,5" } }, "serpapi_pagination": { "current": 1, "next_link":
"https://serpapi.com/search.json?as_sdt=0%2C5&engine=google_scholar&hl=en&num=2&q=corona&start=2", "next":
"https://serpapi.com/search.json?as_sdt=0%2C5&engine=google_scholar&hl=en&num=2&q=corona&start=2", "other_pages": {
"https://serpapi.com/search.json?as_sdt=0%2C5&engine=google_scholar&hl=en&num=2&q=corona&start=2",
"https://serpapi.com/search.json?as_sdt=0%2C5&engine=google_scholar&hl=en&num=2&q=corona&start=4",
"https://serpapi.com/search.json?as_sdt=0%2C5&engine=google_scholar&hl=en&num=2&q=corona&start=6",
"https://serpapi.com/search.json?as_sdt=0%2C5&engine=google_scholar&hl=en&num=2&q=corona&start=8",
"https://serpapi.com/search.json?as_sdt=0%2C5&engine=google_scholar&hl=en&num=2&q=corona&start=10",
"https://serpapi.com/search.json?as_sdt=0%2C5&engine=google_scholar&hl=en&num=2&q=corona&start=12",
"https://serpapi.com/search.json?as_sdt=0%2C5&engine=google_scholar&hl=en&num=2&q=corona&start=14",
"https://serpapi.com/search.json?as_sdt=0%2C5&engine=google_scholar&hl=en&num=2&q=corona&start=16",
"https://serpapi.com/search.json?as_sdt=0%2C5&engine=google_scholar&hl=en&num=2&q=corona&start=18" } } }

GET Method I Provide:

- request to local= http://127.0.0.1:8000/api/google-scholar/?title=corona&rows=2
- response : (status code = 200) (json)

{"results": [{"source": "google_scholar", "authors": [{"name": "JS Chang"}, {"name": "PA Lawless\u2026"}], "id": "d8_j3yqNqHgJ", "date": 1991, "abstract": "Applications of corona discharge induced plasmas and unipolar ions are reviewed. Corona process applications emphasize one of two aspects of the discharge: the ions produced or \u2026", "title": "Corona discharge processes", "url": "https://ieeexplore.ieee.org/abstract/document/125038/", "pos": 0}, {"source": "google_scholar", "authors": [{"name": "BC Low"}], "id": "7P5fKrLoHAcJ", "date": 1996, "abstract": "\u2026 the corona as an electrically highlyconducting atmosphere. The freezing of magnetic flux into the huge volume of the corona \u2026 products of activity in the corona - flares and coronal mass \u2026", "title": "Solar activity and the corona", "url": "https://link.springer.com/article/10.1007/BF00146338", "pos": 1}]}

POST Method I Provide:

- request to local= http://localhost:8000/api/post-papers/
- request body :
  db=semantic-scholar
  title=sad
  rows=3
- header:
  username:admin
  password:123 (there is a user with these credentials)
- response : (status code = 200) (json)
  { "status": "Requested papers are saved successfully." }

Any other significant work related to this release

I was responsible for the dockerization and deployment of the app. For the deployment I used AWS. I was also responsible for creating the Swagger page and Swagger documentation.

Challenges:

It was hard to be organized as a team. People sometimes missed meetings which resulted in them not knowing what we decided. Because of that we had to create couple of PRs to fix non uniform parts of the API methods.

Arda Arslan

- Links to important issues (git issues) and wiki documentation related to practice application
- The name, route, description of utilized third party URIs
- The name, route, description of the created API functions
- Description of unit tests. You can include code fragments to illustrate.
- Sample calls: Show a transcript of a request and response for the API functions you used and that you provide. The transcript should include the request and the detailed response (headers, status code, json etc.)
- Any other significant work related to this release (such as Docker, AWS, DB, Framework)
- Describe any challenges you met during implementation.

Ali Mert Geben

Issues that I have contributed

- Third-Party API (Zenodo) Research - #103,#94
- Zenodo API GET Method - #116
- POST Method for Adding Interest into Profile - #156
- Addition of Add Interest Button into Profile Page - #179
- Documentation of GET Method that Utilizes Zenodo API - #197
- Documentation for POST Method that Adds Interest into Profile - #198
- Addition of an empty database file into repository - #199
- Implementation of GET methods - #110
- Implementation of POST methods -#140
- Handling paper content html file and view - #229
- Documentation of APIs - #185
- Documentation of member contribution - #196

The name, route, description of my utilized third party URLs

I utilized the Zenodo API which is a general-purpose open repository developed under the European OpenAIRE program and operated by CERN. It allows any researcher to deposit their papers and their API allows various search methods. I utilized their search methods and integrated it into our API. You can search by author, title, related subjects etc. but we only used search by title for our purposes. Their GET methods require authentication but anyone can get an API key after a simple registration process. I used three parameters while calling the Zenodo API. q: which is the name of wanted paper, size: which is the number of papers to be returned and sort: which is the sorting method of returned results and I chose ‘bestmatch’ for this parameter. Zenodo API returns a JSONresponse after this call and I decoded this response in order to get wanted attributes from the papers which we agreed as a team.

URL of the their website : https://zenodo.org/

Route of their api : https://zenodo.org/api/records

An example:
https://zenodo.org/api/records?q=”title”&sort=”bestmatch”&size=3&access_token=<youraccestoken>

The name, route, description of the created API functions

GET Method

I implemented a GET method that utilizes the third party Zenodo API. My GET method takes two parameters: title and rows. Title is the name of the paper that is to be searched and rows is the number of papers to be returned. I use these 2 parameters to search from Zenodo API and return a JSON response with our own format. This GET method returns a 404 error if title parameter is empty or search is unsuccessful and returns a 503 error if an error occurred from third party API.

Endpoint: /api/zenodo/

Example: http://localhost:8000/api/zenodo/?title=astronomy&rows=3

Parameters:

title: Specifies the title of the papers to be searched. It must be provided.

rows: Specifies the count of the papers to be returned. It is optional.

Response Format and Fields

A call to this endpoint returns a JSON response of the papers. Each paper record contains the following fields:

id: The id of the paper in the Zenodo database.

title: The title of the paper.

author: Authors of the papers.

source: The source of the paper which is Zenodo in this case.

date: The publication year of the paper.

abstract: The abstract section of the paper.

position: Order of the paper in the resulting paper list.

Post Method

I implemented a POST method endpoint in order to allow users to add interests into their profile page. It gets the parameters required for authentication from headers and value for interest name from body.

This method works by first checking the validity of username and password of the user to authenticate. It raises an error with appropriate status code if they are invalid. Then it continues to add the interest given in the body to the database with user id. If the interest is empty or the user has already added that interest, it raises an error accordingly.

Example request: http://localhost:8000/api/add-interest/

headers = {“username”: “admin”, “password”: “123”}

body = {“interest”=”astronomy”}

Response:

If the user credentials are invalid:

{ “status”: “User credentials are incorrect” **}** status_code=401

If the user credentials are empty:

{“status”: “Username and password fields can not be empty”} status_code=407

If the interest field is empty:

{“status”: “Name of the interest can’t be empty”} status_code=400

If the interest has been added before:

{“status”: “This interest has been already added”} status_code=407

If it is successful:

{“status”: “Interest has been added to profile successfuly!”} status_code=200

Description of unit tests

Description of the tester for GET method:

1. Tester sets up a client object upon which calls the requests then it makes an API call to self with url `"/api/zenodo/?title=test&rows=3"`
2. It checks the status code of the response and asserts it equals to 200.
3. It gets the results field of the response and asserts that its length equals to 3.
4. Then for every paper in the result, tester checks its fields and asserts that the format is the way we wanted.
5. If any of the steps fail, the test also fails. If not, the test ends successfully.

Description of the tester for POST method:

1. Firstly, tester sets up a client object to call the request then it creates a dummy user in order to pass the credential tests.
2. Then it tries to add an interest normally with a self API call, and checks that status code and response is the same as expected.
3. After this, it makes two self API call with same interest field and checks for status code and response.
4. Then it tries to add an empty interest field and asserts that status code and response is the same as expected.
5. After completing tests for interest part, it continues with test for user credentials. It tries to make a request with missing credentials and checks the status code and response again.
6. And lastly, it tries to make a request with wrong credentials, wrong password in my test, and checks the status code and response.
7. If all tests are successful, the test ends successfully.

Sample calls:

Example GET Request to Zenodo API

Request URL: https://zenodo.org/api/records?q=astronomy&sort=bestmatch&size=1&api_key=<MYAPIKEY>

Parameters:

q=astronomy

sort=bestmach

size=1

api_key=<MYAPIKEY> (Hidden due to privacy)

Status Code: 200

Response (It is in raw from due to its length):

{"aggregations":{"access_right":{"buckets":[{"doc_count":4175,"key":"open"},{"doc_count":33,"key":"closed"},{"doc_count":21,"key":"restricted"},{"doc_count":1,"key":"embargoed"}],"doc_count_error_upper_bound":0,"sum_other_doc_count":0},"file_type":{"buckets":[{"doc_count":2633,"key":"pdf"},{"doc_count":695,"key":"zip"},{"doc_count":249,"key":"gz"},{"doc_count":176,"key":"txt"},{"doc_count":155,"key":"md"},{"doc_count":103,"key":"h5"},{"doc_count":97,"key":"png"},{"doc_count":85,"key":"csv"},{"doc_count":75,"key":"def"},{"doc_count":74,"key":"fits"}],"doc_count_error_upper_bound":0,"sum_other_doc_count":1169},"keywords":{"buckets":[{"doc_count":362,"key":"astronomy"},{"doc_count":123,"key":"Astronomy"},{"doc_count":122,"key":"astrophysics"},{"doc_count":75,"key":"ExoMol"},{"doc_count":75,"key":"line list"},{"doc_count":74,"key":"partition function"},{"doc_count":72,"key":"ExoMolOP"},{"doc_count":62,"key":"opacity"},{"doc_count":46,"key":"Exoplanets"},{"doc_count":45,"key":"Biodiversity"}],"doc_count_error_upper_bound":0,"sum_other_doc_count":9017},"type":{"buckets":[{"doc_count":1571,"key":"publication","subtype":{"buckets":[{"doc_count":931,"key":"article"},{"doc_count":178,"key":"conferencepaper"},{"doc_count":178,"key":"thesis"},{"doc_count":86,"key":"report"},{"doc_count":39,"key":"preprint"},{"doc_count":31,"key":"other"},{"doc_count":30,"key":"section"},{"doc_count":24,"key":"book"},{"doc_count":18,"key":"technicalnote"},{"doc_count":14,"key":"deliverable"},{"doc_count":13,"key":"workingpaper"},{"doc_count":10,"key":"datamanagementplan"},{"doc_count":10,"key":"taxonomictreatment"},{"doc_count":4,"key":"proposal"},{"doc_count":3,"key":"softwaredocumentation"},{"doc_count":2,"key":"peerreview"}],"doc_count_error_upper_bound":0,"sum_other_doc_count":0}},{"doc_count":846,"key":"dataset","subtype":{"buckets":[],"doc_count_error_upper_bound":0,"sum_other_doc_count":0}},{"doc_count":748,"key":"presentation","subtype":{"buckets":[],"doc_count_error_upper_bound":0,"sum_other_doc_count":0}},{"doc_count":490,"key":"poster","subtype":{"buckets":[],"doc_count_error_upper_bound":0,"sum_other_doc_count":0}},{"doc_count":410,"key":"software","subtype":{"buckets":[],"doc_count_error_upper_bound":0,"sum_other_doc_count":0}},{"doc_count":64,"key":"image","subtype":{"buckets":[{"doc_count":38,"key":"figure"},{"doc_count":9,"key":"photo"},{"doc_count":9,"key":"plot"},{"doc_count":8,"key":"other"}],"doc_count_error_upper_bound":0,"sum_other_doc_count":0}},{"doc_count":43,"key":"other","subtype":{"buckets":[],"doc_count_error_upper_bound":0,"sum_other_doc_count":0}},{"doc_count":41,"key":"video","subtype":{"buckets":[],"doc_count_error_upper_bound":0,"sum_other_doc_count":0}},{"doc_count":13,"key":"lesson","subtype":{"buckets":[],"doc_count_error_upper_bound":0,"sum_other_doc_count":0}},{"doc_count":4,"key":"workflow","subtype":{"buckets":[],"doc_count_error_upper_bound":0,"sum_other_doc_count":0}}],"doc_count_er

ror_upper_bound":0,"sum_other_doc_count":0}},"hits":{"hits":[{"conceptdoi":"10.5281/zenodo.7095699","conceptrecid":"7095699","created":"2022-09-20T07:06:52.612972+00:00","doi":"10.5281/zenodo.7095700","files":[{"bucket":"24f91848-0037-4229-bb38-fc9e20233627","checksum":"md5:a2a0bdad2c1512a7f61040ed9fcdd359","key":"SSAASS_Lect_hoeft.pdf","links":{"self":"https://zenodo.org/api/files/24f91848-0037-4229-bb38-fc9e20233627/SSAASS_Lect_hoeft.pdf"},"size":24751712,"type":"pdf"}],"id":7095700,"links":{"badge":"https://zenodo.org/badge/doi/10.5281/zenodo.7095700.svg","bucket":"https://zenodo.org/api/files/24f91848-0037-4229-bb38-fc9e20233627","conceptbadge":"https://zenodo.org/badge/doi/10.5281/zenodo.7095699.svg","conceptdoi":"https://doi.org/10.5281/zenodo.7095699","doi":"https://doi.org/10.5281/zenodo.7095700","html":"https://zenodo.org/record/7095700","latest":"https://zenodo.org/api/records/7095700","latest_html":"https://zenodo.org/record/7095700","self":"https://zenodo.org/api/records/7095700"},"metadata":{"access_right":"open","access_right_category":"success","creators":[{"affiliation":"Matthias","name":"Hoeft"}],"description":"<p>Presentation given at the Sub-Saharan African Astronomy Summer School</p>","doi":"10.5281/zenodo.7095700","license":{"id":"CC-BY-4.0"},"publication_date":"2022-09-20","related_identifiers":[{"identifier":"10.5281/zenodo.7095699","relation":"isVersionOf","scheme":"doi"}],"relations":{"version":[{"count":1,"index":0,"is_last":true,"last_child":{"pid_type":"recid","pid_value":"7095700"},"parent":{"pid_type":"recid","pid_value":"7095699"}}]},"resource_type":{"title":"Presentation","type":"presentation"},"title":"Active Galactic Nuclei"},"owners":[410313],"revision":3,"stats":{"downloads":9.0,"unique_downloads":8.0,"unique_views":7.0,"version_downloads":9.0,"version_unique_downloads":8.0,"version_unique_views":7.0,"version_views":8.0,"version_volume":222765408.0,"views":8.0,"volume":222765408.0},"updated":"2022-09-20T14:26:18.641588+00:00"}],"total":4230},"links":{"next":"https://zenodo.org/api/records/?sort=bestmatch&q=astronomy&page=2&size=1","self":"https://zenodo.org/api/records/?sort=bestmatch&q=astronomy&page=1&size=1"}}

Example GET Request to Our API

Request URL: `http://localhost:8000/api/zenodo/?title=astronomy&rows=3`

Parameter:

title=astronomy

rows=3

Status Code : 200

Response:

```
{
  "results": [
    {
      "id": 7095700,
      "title": "Active Galactic Nuclei",
      "url": "https://doi.org/10.5281/zenodo.7095700",
    "authors": [
      {
        "name": "Hoeft"
      }
    ],
    "abstract": "<p>Presentation given at the Sub-Saharan African Astronomy Summer School</p>",
    "date": 2022,
    "position": 0,
    "source": "Zenodo"
  },
  {
    "id": 7915476,
    "title": "The Rights and Duties of Members of the UWC Astronomy Group",
    "url": "https://doi.org/10.5281/zenodo.7915476",
    "authors": [
      {
        "name": "The UWC Astronomy Group Staff"
      }
    ],
    "abstract": "<p>This document outlines&nbsp;The Rights and Duties of Members of the UWC Astronomy Group.</p>",
    "date": 2023,
    "position": 1,
    "source": "Zenodo"
  },
      {
         "id": 897031,
         "title": "THE FLARING ACTIVITY OF M DWARFS IN KEPLER FIELD",
         "url": "https://doi.org/10.5281/zenodo.897031",
         "authors": [
            {
               "name": "Yang,Huiqin"
            }
         ],
         "abstract": "<p>The Kepler flux, M dwarfs that include flares,the format consist of four
columns:<br>\n<br>\nBKJD  Relative Flux  Fit line   Flare Flux</p>\n\n<p>the Relative flux wil be empty when
flares occur, the flares flux will be empty when quiescent flux.</p>\n\n<p>the file lamost_obsid is the spectra
used in this work, one can use the obsid to download the fits through the following website:
http://dr4.lamost.org/</p>",
         "date": 2017,
         "position": 2,
         "source": "Zenodo"
      }
   ]
}
```

Example POST Request to Our API to Add Interest

Request URL: http://localhost:8000/api/add-interest/

Status Code: 401

Parameters:

headers = {“username”: “mario”, “password”: “strongpassword”}

body = {“interest”=”russian”}

Response (Since there are no user named mario):

```
{
        “status”: “User credentials are incorrect.”
}
```

Other possible responses:

If the user credentials are empty:

{“status”: “Username and password fields can not be empty”} status_code=407

If the interest field is empty:

{“status”: “Name of the interest can’t be empty”}   status_code=400

If the interest has been added before:

{“status”: “This interest has been already added”} status_code=407

If it is successful:

{“status”: “Interest has been added to profile successfuly!”} status_code=200

My other significant work related to this release

In addition to implementation of API methods, I worked on the front-end implementation of profile page which Mehmet designed and I implemented an add interest button to front-end design in order to allow users to add their interests to their profile from any profile page. I also reviewed and tested some pull requests from my teammates to help fix some of the problems in them.

Challenges I met during implementation

Like most of my teammates, I didn’t know anything about Django or Web design prior to this project and learning to use new tools was a challenge and it took a considerably long time. Response format I got from Zenodo API was really complicated and getting the attributes I wanted was also hard. I wrote unit tests for the first time and learning them from scratch was challenging, though with the help and examples from my teammates, I was able to handle it in a relatively short time. It was also the first time, I used git to such extent and while it seems easy now, such things as merging, committing, pulling and sending some pull requests with the changes I made was a bit hard at the beginning and I made mistakes.

Zülal Molla
Significant issues that I contributed:

- ORCID API GET Method [Issue #119](https://github.com/bounswe/bounswe2023group9/issues/119)
- Registration Methods[Issue #130](https://github.com/bounswe/bounswe2023group9/issues/130)
- Design and Implementation of Sign-up Page[Issue #135](https://github.com/bounswe/bounswe2023group9/issues/135)
- Design and Implementation of Sign-in Page[Issue #136](https://github.com/bounswe/bounswe2023group9/issues/136)
- Search Paper Page[Issue #152](https://github.com/bounswe/bounswe2023group9/issues/152)
- Search User Page[Issue #174](https://github.com/bounswe/bounswe2023group9/issues/174)
- Design and implementation of list content page [Issue #184](https://github.com/bounswe/bounswe2023group9/issues/184)

- The name, route, description of utilized third party URIs

ORCID API GET Method

I utilized ORCID API to get user information with provided ORCID ID. ORCID is a public platform that allows scientists to represent themselves online. By creating an ORCID ID and profile, scientists may sign in platforms that supports ORCID ID verification.

Their URL: https://orcid.org

Route: `https://orcid.org/`

`Example:` https://orcid.org/0000-0002-5022-178X

headers: {“Accept”: “application/json”}

- The name, route, description of the created API functions

ORCID GET Method

I created ORCID function to utilize third party ORCID API. To use the function, user_id should be provided as a valid ORCID ID. On our platform, we get users’ names and last names from their profile pages.

Endpoint: /api/orcid_api/

Example: http://localhost:8000/api/orcid_api/?user_id=0000-0002-5022-178X

Parameters:

user_id: should be a valid ORCID ID

Response format and fields:

Returns a JsonResponse with these fields:

user_id: same id with provided as input

name: user’s name

surname: user’s surname

Login POST Method

Allows users to log in by using Django's built-in functions. User credentials should be provided in headers. If there is a user registered to our database with the provided username and password, authenticates that user. Else, gives an error.

Endpoint: /api/log_in/

Example: http://localhost:8000/api/log-in/

headers = {“username”: “admin”, “password”: ”123” }

Response Format:

Returns a JsonResponse.

If user id is not provided then returns `{"status":"ORCID ID should be provided.")` `with status = 404`

`If password is not provided then returns {"status":"Password should be provided."} with status = 404`

`If user cannot be authenticated with the given credentials then returns {"status":"Authentication failed"} with status = 404`

`If user can be successfully authenticated then user log in. Returns {"status":"User logged in."} with status = 200`

Logout POST Method

Allows users to log out by using Django's built-in functions. Does not need any parameters or headers since it uses directly the user sent the request.

Endpoint: /api/log_out/

Example Call: http://localhost:8000/api/log-out/

Response Format:

Returns a JsonResponse.

`{"status":"User logged out."} with status = 200`

Register POST Method

Allows users to register to the platform. A valid and unique ORCID ID should be provided in headers. Users can be registered at most once with the same ORCID ID. Gets user’s name and surname from ORCID GET method call. Creation of user on the database is handled by Django's built_in functions that encrypt passwords.

Endpoint: /api/user_registration/

Example call:

http://localhost:8000/api/user-registration/

headers = {“username”: “0000-0002-5022-178X”,”password”:”123”}

Response Format:

Returns a JsonResponse.

If an invalid ORCID ID provided then returns `{"status":"Valid ORCID ID should be provided."} with status = 404`

`If username is not provided then returns {"status":"ORCID ID should be provided."} with status = 404`

`If password is not provided then returns  {"status":"Password should be provided."} with status = 404`

`If user credentials are valid then new user is created and {"status":"User created"} with status = 200 is returned.`

`If username is already taken then returns {"status":"Username is already taken."}, status = 409`

- Description of unit tests. You can include code fragments to illustrate.

ORCID GET Method Tests:

1. *I created a client to send requests to the api.*
2. *I checked 404 responses by passing incomplete parameters.*
3. *I tested the situations when third party ORCID API gives 404 status code that are resulted from invalid ORCID ID’s.*
4. *I tested the function when valid ORCID ID provided. When a unique ID provided, function should return user_id, name and surname with status code 200.*
5. *I compared the responses that are returned by our API and ORCID API by directly calling ORCID API. I checked if the fields are set correctly in the response of our API.*

```
        response =
self.c.get("/api/orcid-api/?user_id=0009-0005-5924-1831")
```

```
        Headers = {"Accept": "application/json"}

        orcid_api_response =
requests.get("https://orcid.org/0009-0005-5924-1831",
headers=Headers).json()

        response = response.json()

        self.assertEquals(response["name"],
orcid_api_response["person"]["name"]["given-names"]["value"])

        self.assertEquals(response["user_id"],
"0009-0005-5924-1831")

        if orcid_api_response["person"]["name"]["family-name"] !=
None:

            self.assertEquals(response["surname"],
orcid_api_response["person"]["name"]["family-name"]["value"])

        else:

            self.assertEquals(response["surname"], None)
```

LOGIN POST Method Tests:

1. *Firstly, I created a client object to send requests and a user.*
2. *I tested 404 responses by giving incomplete user credentials.*
3. *I tested the function by passing a valid username and password. Compared the status code with 200.*

```
    def test_valid_login(self):

        Headers = {'username': "0009-0005-5924-1831", "password":
"strongpassword"}

        response = self.c.post("/api/log-in/",headers = Headers)

        self.assertEquals(response.status_code, 200)
```

4. *I tested invalid login by passing invalid credentials.*

LOGOUT POST Method Tests:

1. *I created a client object to send requests.*
2. *I tested if it successfully returns status code 200 without any parameters.*

REGISTER POST Method Tests:

1. *I created a client object to send requests and a user.*
2. *I tested 404 responses by giving incomplete parameters.*
3. *I tested successful function call by an ORCID ID different from the ID of the user that is created at the first step. I provided the unique ID and the password in the headers. After calling function, I checked if its status is 200. Then I checked if a new user created on the database by filtering User objects with the ID.*
4. *I tested invalid username case by trying to create a user with ORCID ID same as the one of user that is created at the first step.*

- Sample calls: Show a transcript of a request and response for the API functions you used and that you provide. The transcript should include the request and the detailed response (headers, status code, json etc.)

Sample ORCID API call:

Request:

> ***url:*** https://orcid.org/0000-0002-5022-178X
>
> ***headers:*** {“Accept”: “application/json”}

Response:

status: 200

{"orcid-identifier":{"uri":"https://orcid.org/0000-0002-5022-178X","path":"0000-0002-5022-178X","host":"orcid.org"},"preferences":{"locale":"en"},"history":{"creation-method":"DIRECT","completion-date":null,"submission-date":{"value":1572093728158},"last-modified-date":{"value":1681765195567},"claimed":true,"source":null,"deactivation-date":null,"verified-email":true,"verified-primary-email":true},"person":{"last-modified-date":{"value":1681765195563},"name":{"created-date":{"value":1572093728387},"last-modified-date":{"value":1572093728387},"given-names":{"value":"Mehmet Utkan"},"family-name":{"value":"Gezer"},"credit-name":null,"source":null,"visibility":"public","path":"0000-0002-5022-178X"},"other-names":{"last-modified-date":null,"other-name":[],"path":"/0000-0002-5022-178X/other-names"},"biography":null,"researcher-urls":{"last-modified-date":null,"researcher-url":[],"path":"/0000-0002-5022-178X/researcher-urls"},"emails":{"last-modified-date":null,"email":[],"path":"/0000-0002-5022-178X/email"},"addresses":{"last-modified-date":null,"address":[],"path":"/0000-0002-5022-178X/address"},"keywords":{"last-modified-date":null,"keyword":[],"path":"/0000-0002-5022-178X/keywords"},"external-identifiers":{"last-modified-date":{"value":1681765195563},"external-identifier":[{"created-date":{"value":1681765195563},"last-modified-date":{"value":1681765195563},"source":{"source-orcid":null,"source-client-id":{"uri":"https://orcid.org/client/APP-1DSJAQCZ1PW1VN5X","path":"APP-1DSJAQCZ1PW1VN5X","host":"orcid.org"},"source-name":{"value":"Web of Science"},"assertion-origin-orcid":null,"assertion-origin-client-id":null,"assertion-origin-name":null},"external-id-type":"ResearcherID","external-id-value":"AAD-8285-2020","external-id-url":{"value":"https://www.webofscience.com/wos/author/record/AAD-8285-2020"},"external-id-relationship":"self","visibility":"public","path":"/0000-0002-5022-178X/external-identifiers/3020382","put-code":3020382,"display-index":0}],"path":"/0000-0002-5022-178X/external-identifiers"},"path":"/0000-0002-5022-178X/person"},"activities-summary":{"last-modified-date":{"valu

e":1681765187292},"distinctions":{"last-modified-date":null,"affiliation-group":[],"path":"/0000-0002-5022-178X/distinctions"},"educations":{"last-modified-date":null,"affiliation-group":[],"path":"/0000-0002-5022-178X/educations"},"employments":{"last-modified-date":{"value":1572093940206},"affiliation-group":[{"last-modified-date":{"value":1572093940206},"external-ids":{"external-id":[]},"summaries":[{"employment-summary":{"created-date":{"value":1572093940206},"last-modified-date":{"value":1572093940206},"source":{"source-orcid":{"uri":"https://orcid.org/0000-0002-5022-178X","path":"0000-0002-5022-178X","host":"orcid.org"},"source-client-id":null,"source-name":{"value":"Mehmet Utkan
Gezer"},"assertion-origin-orcid":null,"assertion-origin-client-id":null,"assertion-origin-name":null},"put-code":8982636,"department-name":"Computer Engineering","role-title":"Research
Assistant","start-date":{"year":{"value":"2019"},"month":{"value":"01"},"day":{"value":"01"}},"end-date":null,"organization":{"name":"Boğaziçi
University","address":{"city":"Istanbul","region":"İstanbul","country":"TR"},"disambiguated-organization":{"disambiguated-organization-identifier":"grid.11220.30","disambiguation-source":"GRID"}},"url":null,"external-ids":null,"display-index":"1","visibility":"public","path":"/0000-0002-5022-178X/employment/8982636"}}]}],"path":"/0000-0002-5022-178X/employments"},"fundings":{"last-modified-date":null,"group":[],"path":"/0000-0002-5022-178X/fundings"},"invited-positions":{"last-modified-date":null,"affiliation-group":[],"path":"/0000-0002-5022-178X/invited-positions"},"memberships":{"last-modified-date":null,"affiliation-group":[],"path":"/0000-0002-5022-178X/memberships"},"peer-reviews":{"last-modified-date":null,"group":[],"path":"/0000-0002-5022-178X/peer-reviews"},"qualifications":{"last-modified-date":null,"affiliation-group":[],"path":"/0000-0002-5022-178X/qualifications"},"research-resources":{"last-modified-date":null,"group":[],"path":"/0000-0002-5022-178X/research-resources"},"services":{"last-modified-date":null,"affiliation-group":[],"path":"/0000-0002-5022-178X/services"},"works":{"last-modified-date":{"value":1681765187292},"group":[{"last-modified-date":{"value":1665140132899},"external-ids":{"external-id":[{"external-id-type":"doi","external-id-value":"10.1016/j.ic.2021.104744","external-id-normalized":{"value":"10.1016/j.ic.2021.104744","transient":true},"external-id-normalized-error":null,"external-id-url":{"value":"https://doi.org/10.1016/j.ic.2021.104744"},"external-id-relationship":"self"}]},"work-summary":[{"put-code":91156252,"created-date":{"value":1616604897658},"last-modified-date":{"value":1665140132899},"source":{"source-orcid":null,"source-client-id":{"uri":"https://orcid.org/client/0000-0001-9884-1913","path":"0000-0001-9884-1913","host":"orcid.org"},"source-name":{"value":"Crossref"},"assertion-origin-orcid":null,"assertion-origin-client-id":null,"assertion-origin-name":null},"title":{"title":{"value":"Constant-space, constant-randomness verifiers with arbitrarily small
error"},"subtitle":null,"translated-title":null},"external-ids":{"external-id":[{"external-id-type":"doi","external-id-value":"10.1016/j.ic.2021.104744","external-id-normalized":{"value":"10.1016/j.ic.2021.104744","transient":true},"external-id-normalized-error":null,"external-id-url":{"value":"https://doi.org/10.1016/j.ic.2021.104744"},"external-id-relationship":"self"}]},"url":{"value":"https://doi.org/10.1016/j.ic.2021.104744"},"type":"journal-article","publication-date":{"year":{"value":"2022"},"month":{"value":"10"},"day":null},"journal-title":{"value":"Information and
Computation"},"visibility":"public","path":"/0000-0002-5022-178X/work/91156252","display-index":"0"}]},{"last-modified-date":{"value":1681765187292},"external-ids":{"external-id":[{"external-id-type":"wosuid","external-id-value":"WOS:000876366600017","external-id-normalized":{"value":"wos:000876366600017","transient":true},"external-id-normalized-error":null,"external-id-url":{"value":"https://www.webofscience.com/api/gateway?GWVersion=2&SrcApp=Publons&SrcAuth=Publons_CEL&KeyUT=WOS:000876366600017&DestLinkType=FullRecord&DestApp=WOS_CPL"},"external-id-relationship":"self"},{"external-id-type":"doi","external-id-value":"10.1007/978-3-031-07469-1_17","external-id-normalized":{"value":"10.1007/978-3-031-07469-1_17","transient":true},"external-id-normalized-error":null,"external-id-url":null,"external-id-relationship":"self"}]},"work-summary":[{"put-code":124851098,"created-date":{"value":1671265575610},"last-modified-date":{"value":1671265575610},"source":{"source-orcid":null,"source-client-id":{"uri":"https://orcid.org/client/0000-0002-3054-1567","path":"0000-0002-3054-1567","host":"orcid.org"},"source-name":{"value":"Crossref Metadata
Search"},"assertion-origin-orcid":{"uri":"https://orcid.org/0000-0002-5022-178X","path":"0000-0002-5022-178X","host":"orcid.org"},"assertion-origin-client-id":null,"assertion-origin-name":{"value":"Mehmet Utkan
Gezer"}},"title":{"title":{"value":"Real-Time, Constant-Space, Constant-Randomness
Verifiers"},"subtitle":null,"translated-title":null},"external-ids":{"external-id":[{"external-id-type":"doi","external-id-value":"10.1007/978-3-031-07469-1_17","external-id-normalized":{"value":"10.1007/978-3-031-07469-1_17","transient":true},"external-id-normalized-error":null,"external-id-url":null,"external-id-relationship":"self"},{"external-id-type":"issn","external-id-value":"0302-9743","external-id-normalized":{"value":"0302-9743","transient":true},"external-id-normalized-error":null,"external-id-url":null,"external-id-relationship":"part-of"}]},"url":null,"type":"other","publication-date":{"year":{"value":"2022"},"month":{"value":"05"},"day":null},"journal-title":{"value":"Implementation and Application of
Automata"},"visibility":"public","path":"/0000-0002-5022-178X/work/124851098","display-index":"0"},{"put-code":

133273113,"created-date":{"value":1681765187292},"last-modified-date":{"value":1681765187292},"source":{"so
urce-orcid":null,"source-client-id":{"uri":"https://orcid.org/client/APP-1DSJAQCZ1PW1VN5X","path":"APP-1DSJA
QCZ1PW1VN5X","host":"orcid.org"},"source-name":{"value":"Web of
Science"},"assertion-origin-orcid":null,"assertion-origin-client-id":null,"assertion-origin-name":null},"title":{"title":{
"value":"Real-Time, Constant-Space, Constant-Randomness
Verifiers"},"subtitle":null,"translated-title":null},"external-ids":{"external-id":[{"external-id-type":"doi","external-id-va
lue":"10.1007/978-3-031-07469-1_17","external-id-normalized":{"value":"10.1007/978-3-031-07469-1_17","tran
sient":true},"external-id-normalized-error":null,"external-id-url":{"value":"https://doi.org/10.1007/978-3-031-074
69-1_17"},"external-id-relationship":"self"},{"external-id-type":"wosuid","external-id-value":"WOS:0008763666000
17","external-id-normalized":{"value":"wos:000876366600017","transient":true},"external-id-normalized-error":nu
ll,"external-id-url":{"value":"https://www.webofscience.com/api/gateway?GWVersion=2&SrcApp=Publons&SrcAu
th=Publons_CEL&KeyUT=WOS:000876366600017&DestLinkType=FullRecord&DestApp=WOS_CPL"},"external-id
-relationship":"self"}]},"url":{"value":"https://publons.com/wos-op/publon/56938462/"},"type":"journal-article","pub
lication-date":{"year":{"value":"2022"},"month":null,"day":null},"journal-title":{"value":"Lecture Notes in Computer
Science"},"visibility":"public","path":"/0000-0002-5022-178X/work/133273113","display-index":"0"}]},{"last-modifi
ed-date":{"value":1681765186865},"external-ids":{"external-id":[{"external-id-type":"doi","external-id-value":"10.1
007/978-3-030-40608-0_12","external-id-normalized":{"value":"10.1007/978-3-030-40608-0_12","transient":true
},"external-id-normalized-error":null,"external-id-url":{"value":"https://doi.org/10.1007/978-3-030-40608-0_12"},"
external-id-relationship":"self"},{"external-id-type":"wosuid","external-id-value":"INSPEC:19435550","external-id-
normalized":{"value":"inspec:19435550","transient":true},"external-id-normalized-error":null,"external-id-url":{"val
ue":"https://www.webofscience.com/api/gateway?GWVersion=2&SrcApp=Publons&SrcAuth=Publons_CEL&Key
UT=INSPEC:19435550&DestLinkType=FullRecord&DestApp=WOS_CPL"},"external-id-relationship":"self"}]},"work
-summary":[{"put-code":69672838,"created-date":{"value":1582783745054},"last-modified-date":{"value":165386
7245134},"source":{"source-orcid":null,"source-client-id":{"uri":"https://orcid.org/client/0000-0001-9884-1913","p
ath":"0000-0001-9884-1913","host":"orcid.org"},"source-name":{"value":"Crossref"},"assertion-origin-orcid":null,"a
ssertion-origin-client-id":null,"assertion-origin-name":null},"title":{"title":{"value":"Windable Heads and
Recognizing NL with Constant
Randomness"},"subtitle":null,"translated-title":null},"external-ids":{"external-id":[{"external-id-type":"doi","external-i
d-value":"10.1007/978-3-030-40608-0_12","external-id-normalized":{"value":"10.1007/978-3-030-40608-0_12","
transient":true},"external-id-normalized-error":null,"external-id-url":{"value":"https://doi.org/10.1007/978-3-030-
40608-0_12"},"external-id-relationship":"self"}]},"url":{"value":"https://doi.org/10.1007/978-3-030-40608-0_12"},"t
ype":"book-chapter","publication-date":{"year":{"value":"2020"},"month":null,"day":null},"journal-title":null,"visibility"
:"public","path":"/0000-0002-5022-178X/work/69672838","display-index":"0"},{"put-code":133273112,"created-da
te":{"value":1681765186865},"last-modified-date":{"value":1681765186865},"source":{"source-orcid":null,"source
-client-id":{"uri":"https://orcid.org/client/APP-1DSJAQCZ1PW1VN5X","path":"APP-1DSJAQCZ1PW1VN5X","host":
"orcid.org"},"source-name":{"value":"Web of
Science"},"assertion-origin-orcid":null,"assertion-origin-client-id":null,"assertion-origin-name":null},"title":{"title":{
"value":"Windable Heads and Recognizing NL with Constant
Randomness"},"subtitle":null,"translated-title":null},"external-ids":{"external-id":[{"external-id-type":"doi","external-i
d-value":"10.1007/978-3-030-40608-0_12","external-id-normalized":{"value":"10.1007/978-3-030-40608-0_12","
transient":true},"external-id-normalized-error":null,"external-id-url":{"value":"https://doi.org/10.1007/978-3-030-
40608-0_12"},"external-id-relationship":"self"},{"external-id-type":"wosuid","external-id-value":"INSPEC:19435550
","external-id-normalized":{"value":"inspec:19435550","transient":true},"external-id-normalized-error":null,"extern
al-id-url":{"value":"https://www.webofscience.com/api/gateway?GWVersion=2&SrcApp=Publons&SrcAuth=Publ
ons_CEL&KeyUT=INSPEC:19435550&DestLinkType=FullRecord&DestApp=WOS_CPL"},"external-id-relationship":
"self"}]},"url":{"value":"https://publons.com/wos-op/publon/57078338/"},"type":"journal-article","publication-date":
{"year":{"value":"2020"},"month":null,"day":null},"journal-title":{"value":"Language and Automata Theory and
Applications. International Conference, LATA. Proceedings. Lecture Notes in Computer Science (LNCS
12038)"},"visibility":"public","path":"/0000-0002-5022-178X/work/133273112","display-index":"0"}]}],"path":"/0000
-0002-5022-178X/works"},"path":"/0000-0002-5022-178X/activities"},"path":"/0000-0002-5022-178X"}

ORCID sample call:

Request: http://localhost:8000/api/orcid_api/?user_id=0000-0002-5022-178X

Response:

status: 200 (user is created beforehand)

{ “user_id”: “0000-0002-5022-178X”, “name”: “Mehmet Utkan”, “surname”:”Gezer”}

Sample LOGIN Call:

Request:

url: http://localhost:8000/api/log-in/

headers: { “username”: “admin”, “password”: ”123” *}*

Response:

status: `200`

`{"status":"User logged in."}`

Sample LOGOUT Call:

Request:

url: http://localhost:8000/api/log-out/

Response:

status: `200`

`{"status":"User logged out."}`

`Sample REGISTER Call:`

`url:`http://localhost:8000/api/user-registration/

headers: {“username”: “0000-0002-5022-178X”,”password”:”123”}

Response:

status: 200

`{"status":"User created"}`

- Any other significant work related to this release (such as Docker, AWS, DB, Framework)

I contributed to the design and implementation of Sign Up, Sign In, List Content, Search Paper and Search User Pages.

- Describe any challenges you met during implementation.

I was not very experienced with front-end development with django. It was very challenged for me at first. But I have learned lots of things about front-end development during implementation.

I have never utilized and created any API functions therefore the concept was new for me. Having to learn so many new topics made the implementation process pretty slow.

Oğuz Pançuk

- Links to important issues (git issues) and wiki documentation related to practice application
- The name, route, description of utilized third party URIs
- The name, route, description of the created API functions
- Description of unit tests. You can include code fragments to illustrate.
- Sample calls: Show a transcript of a request and response for the API functions you used and that you provide. The transcript should include the request and the detailed response (headers, status code, json etc.)
- Any other significant work related to this release (such as Docker, AWS, DB, Framework)
- Describe any challenges you met during implementation.

Ahmed Bera Pay

- Third Party API Research (ERIC API) - [Issue #94](https://github.com/bounswe/bounswe2023group9/issues/94)
- ERIC API - Ahmed Bera Pay - [Issue #97](https://github.com/bounswe/bounswe2023group9/issues/97)
- Implementation of GET Methods - [Issue #110](https://github.com/bounswe/bounswe2023group9/issues/110)
- ERIC API GET Method - [Issue #114](https://github.com/bounswe/bounswe2023group9/issues/114)
- Models should be created - [Issue #117](https://github.com/bounswe/bounswe2023group9/issues/117)
- Implementation of POST Methods - [Issue #140](https://github.com/bounswe/bounswe2023group9/issues/140)
- POST Method to Save Paper Lists - [Issue #137](https://github.com/bounswe/bounswe2023group9/issues/137)
- Design and Implementation of Saved Paper Lists Page - [Issue #165](https://github.com/bounswe/bounswe2023group9/issues/165)
- Meeting #12 Notes will be added - [Issue #176](https://github.com/bounswe/bounswe2023group9/issues/176)

- Documentation of GET method which utilizes the ERIC API - [Issue #190](https://github.com/bounswe/bounswe2023group9/issues/190)
- Documentation of POST method for saving paper lists - [Issue #191](https://github.com/bounswe/bounswe2023group9/issues/191)
- Member Contribution - Ahmed Bera Pay - [Issue #193](https://github.com/bounswe/bounswe2023group9/issues/193)
- Bug in the following lists page - [Issue #214](https://github.com/bounswe/bounswe2023group9/issues/214)

- **The name, route, description of utilized third party URI**

I utilized the ERIC API. ERIC stands for “Education Resources Information Center” which provides a comprehensive online bibliographic and full-text database of education research and information. The website is offered free for public use and no membership is required to access the content. It only supports GET calls to the database and I used this function to make queries about the papers. The search query can be in several fields such as the following core fields: title, author, source, subject, and description. I only used the title field to search. Also the response format could be specified and I chose jSON format since it was the expected one. The desired fields can also be specified in the response format, and I chose the ones agreed upon as a team to have a standard response. In addition to these, with “rows” parameter the count of the returned papers can be specified.

It has the following route: https://api.ies.ed.gov/eric/

An example: https://api.ies.ed.gov/eric/?search=title:GPU&rows=2

- The name, route, description of the created API functions

GET Method

I implemented a GET endpoint that utilizes the third party ERIC API. This API function searches for papers with the given title using the ERIC API and retrieves the requested number of them. Since the ERIC API allows searching in many fields, the method restricts only to searching in titles.

If no paper is found with the provided title a 404 error response is returned.

Endpoint: /api/eric/

Example: http://localhost:8000/api/eric/?title=nanotechnology&rows=3

Parameters:

title: Specifies the title of the papers to be searched. It must be provided.

rows: Specifies the count of the papers to be returned. It is optional.

Response Format and Fields

A call to this endpoint returns a JSON response of the papers. Each paper record contains the following fields:

id: The id of the paper in the ERIC database.

title: The title of the paper.

author: Authors of the papers formatted as surname, name.

source: The source of the paper which is ERIC in this case.

date: The publication year of the paper.

abstract: The abstract section of the paper.

position: Order of the paper in the resulting paper list.

POST Method

This endpoint saves the current logged in user to the saver array of the paper list whose id is provided with the POST method. A status message is returned according to the success or failure type of the request.

The method that is run by the endpoint first checks if the user is logged in or not. If the user is not logged in but some credential information is provided in the headers, then it tries to authenticate the user. If the credentials are not valid or provided at all, an error message is returned. After the authentication process of the user the method tries to access the paper list with the given id. If there exists no paper list with the given id a 404 error is returned. If the given id of the paper list is valid, then the current logged in user is added to the saver array of the paper list.

Request Headers

If provided request headers are as follows:

username: Username of the user

password: Password of the user

Parameters:

POST parameter: (provided in Body form-data)

paper_list_id: The id of the paper list which the current logged in user saves/follows.

Response Format and Fields:

A call to this endpoint returns a JSON response which includes the status of the operation with the following field:

status: Specifies the success or failure status of the operation.

- Description of unit tests. You can include code fragments to illustrate.

I wrote unit tests for the GET and the POST methods.

Unit tests for the GET method works as follows.

First it sets up a client object upon which calls the requests. I thought about four different cases in general. First it tests for the case where the provided title is invalid. It can be empty or not given at all. An error JSON response is expected for those cases. I also regarded the case where an invalid row count is provided such as a non numeric value. This case also expects an error. Finally I tested for the cases where a valid title is provided but no row count provided and a valid row count is provided. These cases expect a success code and the required fields in the response.

Here is a snippet:

```
def test_valid_rows(self):

       # test when valid title and rows are provided

       field_count = 8

       response = self.client.get('/api/eric/?title=education&rows=10')

       self.assertEqual(response.status_code, 200)

       self.assertEqual(len(response.json()['papers'][0]), field_count)

       self.assertContains(response, 'id')

       self.assertContains(response, 'title')

       self.assertContains(response, 'author')

       self.assertContains(response, 'abstract')

       self.assertContains(response, 'source')

       self.assertContains(response, 'date')
```

```
        self.assertContains(response, 'url')

        self.assertContains(response, 'position')
```

Unit tests for the POST method works as follows:

First it sets up a client object for requests. Also it creates a user and a paper list object since the objective of the post method is adding the user to a given paper list object. For this method, I mainly considered four different cases. The first case is where the user is authenticated,logged in and the provided paper list id is valid. Here a successful response is expected. I also considered the cases where the user is not authenticated because of either empty or invalid credential information. These cases expect an error message since the current user is not a logged in authenticated user. Finally I tested the case where the user is authenticated but the provided paper list id is invalid. It can be due to a reference to a non-existing paper list object, empty id or a non-numeric id. Here also the expected response is an error response.

Here is an example snippet from the test code:

```
def test_save_paper_list_authenticated(self):

        # Testing for the successful case with a valid paper list id and valid
credentials

        url = '/api/save-paper-list/'

        headers = {

            'HTTP_USERNAME': 'testuser',

            'HTTP_PASSWORD': 'testpass',

        }

        response = self.client.post(url, {'paper_list_id': self.paper_list.id},
**headers)

        self.assertEqual(response.status_code, 200)

        self.paper_list.refresh_from_db()

        self.assertIn(self.user, self.paper_list.saver.all())

        self.assertEqual(

            response.json()['status'],

            'Paper list is saved successfully!'

        )
```

- Sample calls

Example GET request to ERIC API:

Request URL: https://api.ies.ed.gov/eric/?search=title%3Ananotechnology&rows=1

Response

Code: 200

Body: {

```
  "response":{"numFound":132,"start":0,"numFoundExact":true,"docs":[
    {
      "id":"EJ837785",
      "title":"Nanotechnology Slips into Schools",
      "author":["Cavanagh, Sean"],
      "description":"Some schools are crafting lessons with help from local universities
and companies that work in nanoscience. That's the case at Ballston Spa High, located
in an area of eastern New York known as Tech Valley, home to many technology firms
and top-flight research institutions. By delving into nanotechnology, a specialized subject
that is more commonly taught at the university level, teenagers in the 4,500-student
Ballston Spa district not only gain an understanding of a rapidly advancing area of
science, but also pick up skills coveted by local employers.",
      "subject":["Higher Education", "Technology",  "High Schools", "Students", "Scientific
Research",
        "Science Education",
        "Science Instruction",
        "Adolescents",
        "Engineering",
        "Engineering Education"],
      "publicationtype":["Journal Articles",
        "Reports - Descriptive"],
      "publicationdateyear":2009,
      "language":["English"],
      "issn":["ISSN-0277-4232"],
```

"publisher":"Editorial Projects in Education. 6935 Arlington Road Suite 100, Bethesda, MD 20814-5233. Tel: 800-346-1834; Tel: 301-280-3100; e-mail: customercare@epe.org; Web site: http://www.edweek.org/info/about/",

"peerreviewed":"F"}]

}}

Headers:

content-length: 1482

content-type: application/json;charset=utf-8

Example GET request to our API: http://localhost:8000/api/eric/?title=nanotechnology&rows=1

Parameters:

title: nanotechnology

rows: 1

Response:

```
HTTP/1.1 200 OK

Date: Fri, 12 May 2023 18:13:57 GMT

Server: WSGIServer/0.2 CPython/3.10.11

Content-Type: application/json

X-Frame-Options: DENY

Content-Length: 805

X-Content-Type-Options: nosniff

Referrer-Policy: same-origin

Cross-Origin-Opener-Policy: same-origin
```

{"results": [{"id": "EJ837785", "title": "Nanotechnology Slips into Schools", "author": [{"name": "Cavanagh, Sean"}], "source": "eric-api", "url": "http://www.edweek.org/ew/toc/2009/04/01/index.html", "date": 2009, "abstract": "Some schools are crafting lessons with help from local universities and companies that work in nanoscience. That's the case at Ballston Spa High, located in an area of eastern New York known as Tech Valley, home to many technology firms and top-flight research institutions. By delving into nanotechnology, a specialized subject that is more commonly taught at the university level, teenagers in the 4,500-student Ballston Spa district not only gain an understanding of a rapidly advancing area of science, but also pick up skills coveted by local employers.", "position": 0}]}

Example POST request to out API:

http://localhost:8000/api/save-paper-list/

Parameters:

headers = {“username”: “admin”, “password”: “123”}

body = {“paper_list_id”=”1”}

Response:
```
HTTP/1.1 200 OK
Date: Thu, 11 May 2023 19:12:49 GMT
Server: WSGIServer/0.2 CPython/3.10.11
Content-Type: application/json
X-Frame-Options: DENY
Content-Length: 47
Vary: Cookie
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin

{"status": "Paper list is saved successfully!"}
```

- Any other significant work related to this release (such as Docker, AWS, DB, Framework)

In addition to implementation of API functions, I implemented the front-end of the page which displays the following paper lists of the logged in user. Also I implemented all the models used in the database of our application.

- Describe any challenges you met during implementation.

Like most of us, I was completely new to Django so I needed to learn it. Also running and testing the application without errors were also challenging to me. The most challenging part was documentation and reporting.

Ahmet Abdullah Susuz

- Links to important issues (git issues) and wiki documentation related to practice application

- Third Party API Research for Semantic Scholar- Issue#94, Issue#95
- Implementation and testing of GET Method for Semantic Scholar API - Issue#110, Issue#126
- Implementation and testing of POST Method to Like Paper- Issue#140, Issue#203

- Documentation of GET and POST methods that I implemented - Issue#185, Issue#204, Issue#205
- Milestone report - Issue#192, Issue#226

- The name, route, description of utilized third party URIs

Semantic Scholar is an online platform designed to facilitate access to academic research papers and provide tools for exploring scholarly literature. Semantic Scholar provides a range of APIs that allow users to access academic paper metadata, citation information, and other related data. Details about paper API is a get api which returns papers with their title, id, date and other fields. It also provides to choose fields of returned paper results.

- The name, route, description of the created API functions

GET METHOD

I implemented a GET method which calls the search for papers by keyword method of the Semantic Scholar API for getting papers. It requires two parameters which are title and rows. “title” matches exactly with “query” of the Semantic ScholarAPI, while “rows” indicates the number of results in the response and it matches with the limit parameter. These parameters are mandatory, meaning that the GET method that I implemented returns a json file with 4xx status code when any of the parameters is missing. Response has the following format:

```
response = {
        “status_code”: 200,
        results = [
                {
                        “id”: “ID OF THE PAPER”,
                        “source”: “DOAJ”,
                        “position”: <POSITION IN RESULTS>,
                        “authors”: [ LIST OF AUTHORS ],
                        “date”: <DATE OF THE PAPER>,
                        “abstract”: “ABSTRACT OF THE PAPER”,
                        “title”: “TITLE OF THE PAPER”,
                        “url”: “URL OF THE PAPER”
                }
```

```
            ], ...
    }
```

POST METHOD

I also implemented post method which adds entry to Like table. It consists user and paper. If specified paper exists and user is not already liked the paper, it increments paper like count and add entry to table. It gets only paper_id field in body with username and password in headers.

Here are response status codes:

- *Status: 200 OK - If the paper is successfully liked.*
- *Status: 400 Bad Request - If the paper_id is missing or empty in the request.*
- *Status: 401 Unauthorized - If the user credentials are incorrect (for anonymous users).*
- *Status: 404 Not Found - If the paper with the provided paper_id does not exist.*
- *Status: 407 Proxy Authentication Required - If the username or password fields are empty (for anonymous users).*
- *Status: 409 Conflict - If the user has already liked the paper.*

- Description of unit tests.

GET METHOD

I wrote some test cases for semantic scholar get papers API. It is implemented in tests.py file and includes these methods:

setUp(): The setUp() method is a setup function that runs before each test case. It initializes the test client (self.client) for making HTTP requests.

tearDown(): The tearDown() method is a cleanup function that runs after each test case. In this case, it simply prints a message indicating the completion of the GET tests.

test_404_responses(): This test method checks for 404 responses from various URL endpoints of the Semantic Scholar API. It makes GET requests to different URL patterns and asserts that the response status codes should be 404 (Not Found).

test_results(): This test method verifies the results returned by the Semantic Scholar API. It performs the following steps: It sends a GET request to the Semantic Scholar API with a specific title (covid) and limit (rows=3). It asserts that the response status code should be 200 (OK). It extracts the response content as JSON and asserts that the length of the results array should be 3. It iterates through each result and asserts the presence of certain keys (source, authors, id, abstract, url, date, title) in the result dictionary. It also verifies the position of each result, which should match the count value.

Note: The test case also includes commented code that suggests a comparison of results between the Semantic Scholar API and the test API. However, due to the dynamic nature of the Semantic Scholar API, the commented assertions may not always work as the API can return different results for the same query. Overall, this code performs unit tests to ensure that the Semantic Scholar API responds correctly with the expected status codes and result structures.

POST METHOD

I also wrote some test cases for like-paper API. It is implemented in tests.py file and includes these methods:

setUp(): The setUp() method is a setup function that runs before each test case. It initializes the test client (self.client) for making HTTP requests and sets up necessary objects for testing, including a user, and two paper objects.

tearDown(): The tearDown() method is a cleanup function that runs after each test case. In this case, it simply prints a message indicating the completion of the tests.

test_Missing_Credentials(): This test method verifies the behavior of the API when credentials are missing. It sends a POST request to the API endpoint without providing the required username and password headers. It asserts that the response status code should be 407 (Proxy Authentication Required) and the response content should contain an appropriate error message.

test_Wrong_Credentials(): This test method checks the response of the API when wrong credentials are provided. It sends a POST request to the API endpoint with incorrect username and password headers. It asserts that the response status code should be 401 (Unauthorized) and the response content should contain an error message indicating incorrect user credentials.

test_like_paper(): This test method validates the functionality of liking a paper. It sends a POST request to the API endpoint with a valid paper ID and user credentials. It asserts that the response status code should be 200 (OK) and the response content should indicate that the paper has been liked successfully.

test_already_liked(): This test method simulates the scenario where a paper is already liked by the user. It sends multiple POST requests to the API endpoint with the same paper ID and user credentials. It asserts that the response status code should be 409 (Conflict) and the response content should indicate that the paper has already been liked.

test_empty_paper(): This test method tests the case where the paper ID is empty. It sends a POST request to the API endpoint with an empty paper ID and valid user credentials. It asserts that the response status code should be 400 (Bad Request) and the response content should indicate that a valid paper ID should be provided.

test_wrong_paper_id(): This test method checks the behavior of the API when an invalid paper ID is provided. It sends a POST request to the API endpoint with a non-existent paper ID and valid user credentials. It asserts that the response status code should be 404 (Not Found) and the response content should indicate that the paper ID is invalid

- Sample calls: Show a transcript of a request and response for the API functions you used and that you provide. The transcript should include the request and the detailed response (headers, status code, json etc.)

GET METHOD

Example request : http://127.0.0.1:8000/api/semantic-scholar/?title=sad&rows=5

RESPONSE

```
{
  "results": [
    {
      "source": "semantic_scholar",
      "authors": [
        {
          "name": "A. Baldassarre"
        },
        {
          "name": "G. Giorgi"
        },
        {
          "name": "Federico Alessio"
        },
        {

          "name": "L. Lulli"

        },

        {

          "name": "G. Arcangeli"

        },

        {

          "name": "N. Mucci"

        }

      ],

      "id": "8ce7b456be392edd7eed0b690feb361da88d65c5",

      "abstract": "Infectious disease control is a crucial public health issue. Although it is important to urgently perform public health measures in order to reduce the risk of spread, it could end up stigmatizing entire groups of people rather than offering control measures based on sound scientific principles. This “us” versus “them” dynamic is common in stigmatization, in general, and indicates a way in which disease stigma can be viewed as a proxy for other types of fears, especially xenophobia and general fear of outsiders. The pandemic risk associated with SARS-CoV-2 infection led us to consider, among other related issues, how stigma and discrimination remain serious barriers to care for people suspected of being infected, even more if they are assisting professions, such as health workers, employed in emergency response. The purpose of this review is to evaluate and promote the importance of psychological aspects of the stigma and social discrimination (SAD) in pandemic realities and, more specifically, nowadays, in the context of SARS-CoV-2/COVID-19. Just as it happened with HIV, HCV, tuberculosis, and Zika, stigma and discrimination undermine the social fabric compromising the ethics and principles of civilization to which each individual in entitled. Recognizing disease stigma history can give us insight into how, exactly, stigmatizing attitudes are formed, and how they are disbanded. Instead of simply blaming the ignorance of people espousing stigmatizing attitudes about certain diseases, we should try to understand precisely how these attitudes are formed so that we can intervene in their dissemination. We should also look at history to see what sorts of interventions against stigma may have worked in the past. Ongoing research into stigma should evaluate what has worked in the past, as above-mentioned, providing us with some clues as to what might work in the current pandemic emergency, to reduce devastating discrimination that keeps people from getting the care they need. We propose a systematic and historical review, in order to create a scientific and solid base for the following SAD analysis. The aim is to propose a coping strategy to face stigma and discrimination (SAD) related to SARS-CoV-2/COVID-19 pandemic outbreak, borrowing coping strategy tools and solutions from other common contagious diseases. Furthermore, our study observes how knowledge, education level, and socioeconomic status (SES) can influence perception of SARS-CoV-2/ COVID-19 risk in a digital world, based on previous research, best practices, and evidence-based research.",
        "title": "Stigma and Discrimination (SAD) at the Time of the SARS-CoV-2 Pandemic",
        "url": "https://www.semanticscholar.org/paper/8ce7b456be392edd7eed0b690feb361da88d65c5",
        "date": 2020,
        "position": 0
    },
    {
        "source": "semantic_scholar",
        "authors": [
            {
                "name": "Sunkyung Yoon"
            },
            {
                "name": "E. Verona"
            },
            {
                "name": "R. Schlauch"
            },
            {
                "name": "Sandra Schneider"
            },
            {
                "name": "J. Rottenberg"
            }
        ],
        "id": "e4a6dd8ddcbe98880164d4ab041d1edebd0b5718",
```

"abstract": "One of the cardinal symptoms of major depressive disorder (MDD) is persistent sadness. Do people with MDD actually prefer sad stimuli, potentially perpetuating their depression? Millgram, Joormann, Huppert, and Tamir (2015) observed such preferences and interpreted them as reflecting a maladaptive emotion regulatory goal to upregulate sad feelings. We assessed emotional music choice among both those with MDD and healthy controls (HC), and assessed the reasons for music preferences in these two groups. Seventy-six female participants (38 per group) completed two tasks: (1) Millgram et al.'s (2015) music task wherein participants listened to happy, neutral, and sad music excerpts and chose the one they wanted to listen to most, and (2) a novel Emotional Music Selection Task (EMST) wherein participants chose preferred music clips, varying in emotion and energy level, in paired-choice trials. In the replication music task, MDD people were more likely to choose sad music. However, inconsistent with any motivation to upregulate sadness, people with MDD reported that they chose sad music because it was low in energy levels (e.g., relaxing). EMST results revealed that MDD people had a stronger preference for both low energy and sad music, relative to HC. The strong appeal of sad music to people with MDD may be related to its calming effects rather than any desire to increase or maintain sad feelings. (PsycINFO Database Record (c) 2019 APA, all rights reserved).",

"title": "Why do depressed people prefer sad music?",

"url": "https://www.semanticscholar.org/paper/e4a6dd8ddcbe98880164d4ab041d1edebd0b5718",

"date": 2020,

"position": 1

},

{

"source": "semantic_scholar",

"authors": [

{

"name": "L. Steenbergen"

},

{

"name": "R. Sellaro"

},

{

"name": "S. V. Hemert"

},

```
            {
                "name": "J. Bosch"
            },
            {
                "name": "L. Colzato"
            }
        ],
        "id": "f98dd54d37081cddcccad5cbbd022c22bf53081b",
        "abstract": null,
        "title": "A randomized controlled trial to test the effect of multispecies probiotics on cognitive reactivity to
sad mood",
        "url": "https://www.semanticscholar.org/paper/f98dd54d37081cddcccad5cbbd022c22bf53081b",
        "date": 2015,
        "position": 2
    },
    {
        "source": "semantic_scholar",
        "authors": [
            {
                "name": "G. Lovink"
            }
        ],
        "id": "25612ebc744401575ffc1448f8cdc0b7bb5f7a0b",
        "abstract": "Sadness is now a design problem. The highs and lows of melancholy are coded into social
media platforms. After all the clicking, browsing, swiping and liking, all we are left with is the flat and empty
aftermath of time lost to the app. Sad by Design offers a critical analysis of the growing social media
controversies such as fake news, toxic viral memes and online addiction. The failed search for a grand design has
```

resulted in depoliticised internet studies unable to generate either radical critique or a search for alternatives. Geert Lovink calls for us to embrace the engineered intimacy of social media, messenger apps and selfies, because boredom is the first stage of overcoming ‘platform nihilism’. Then, after the haze, we can organise to disrupt the data extraction industries at their core.",

    "title": "Sad by Design",

    "url": "https://www.semanticscholar.org/paper/25612ebc744401575ffc1448f8cdc0b7bb5f7a0b",

    "date": 2019,

    "position": 3

  },

  {

    "source": "semantic_scholar",

    "authors": [

      {

        "name": "Tom Ter Bogt"

      },

      {

        "name": "Natale Canale"

      },

      {

        "name": "Michela Lenzi"

      },

      {

        "name": "A. Vieno"

      },

      {

        "name": "R. van den Eijnden"

```
            }
        ],
        "id": "e02d90192ede6efa9ef78e3a1a006d231a5b78da",
        "abstract": "This research explored both social context and personal characteristics in relation to being saddened by sad music when in a sad mood. Overall, 1686 respondents (aged 12–16 years; 44% female; 68% vocational training) answered questions about their background, social context (family climate, bullying issues), personal characteristics (depressive mood, self-esteem, social comparison style), and the degree to which they were saddened by listening to sad music. About 17% of the participants reported saddened mood as a consequence of listening to sad music when sad. Multivariate linear regression results revealed that female respondents and those who reported elevated levels of depressive mood and negative social comparison to peers were more likely to feel saddened. These young people further showed strained peer relations, as indicated by being bullied on social network sites and the Internet. Harsh family climate and low self-esteem correlated with the saddened mood-inducing effect of sad music, although they were not significant in multivariate analysis. This pattern of person and context relations with saddened mood was identical for girls and boys. Adolescents generally seek music that is mood-congruent; however, this research questioned whether adolescents with problems should dwell on music that reflects their unhappy state, as this may worsen their mood.",
        "title": "Sad music depresses sad adolescents: A listener's profile",
        "url": "https://www.semanticscholar.org/paper/e02d90192ede6efa9ef78e3a1a006d231a5b78da",
        "date": 2019,
        "position": 4
      }
   ]
}
```

POST METHOD

Example request : http://127.0.0.1:8000/api/like-paper/

RESPONSE

```
{
   "status": "Paper liked."
}
```

After i send same request

```
{
  "status": "You are already liked this paper."
}
```

I send it with wrong paper id(4)

```
{
  "status": "Paper id is invalid."
}
```

- Describe any challenges you met during implementation.

It was hard for me to work with a team and change the codes in the common repo and file. Also, trying to learn a new framework and understanding the unit test logic was one of the factors that slowed me down. In addition, deciding which APIs should be written in the meetings and discussing how they could be combined with the frontend were also important issues.

Mehmet Süzer

- ***Links to important issues (git issues) and wiki documentation related to practice application***

- Third Party API Research for DOAJ - Issue#94, Issue#96
- Implementation and testing of GET Method for DOAJ API - Issue#110, Issue#121
- Implementation and testing of POST Method to Add a Paper to Paper List - Issue#140, Issue#146
- Documentation of GET and POST methods that I implemented - Issue#185, Issue#211, Issue#212
- Creating Templates for User Interface of the Practice App (Sign-in, My Lists, Search Paper, Search User, Follow Requests, Following Lists) - Issue#134
- Design and Implementation of the Sign-in Page - Issue#136
- Implementation of the User Profile Page - - Issue#166
- Meeting Notes #10 - Issue#107
- Milestone report - Issue#182, Issue#206

- The name, route, description of utilized third party URIs

DOAJ is a platform where scientists publish their articles and journals. In order to access some features of the DOAJ API, users need to publish an article in the platform. A non-registered, ordinary user can only search articles and journals. I utilized the search article feature of the DOAJ API to be compatible with my project partners' work. The GET method for search article feature has parameters named search_query, page, pageSize, and sort. "search_query" is a type of query which allows us to find some specific words in an article. "page" refers the page that we want to see. "pageSize" indicates the number of results in the response, while "sort" describes in which order the results will be.

- ***The name, route, description of the created API functions***

I implemented a GET method which calls the GET method of the DOAJ API for articles. It requires two parameters which are title and rows. "title" matches exactly with "search_query" of the DOAJ API, while "rows" indicates the number of results in the response. These parameters are mandatory, meaning that the GET method that I implemented returns a json file with 4xx status code when any of the parameters is missing. Response has the following format:

```
response = {
        "status_code": 200,
        "count": <NUMBER OF RESULTS RETURNED>,
        results = [
                {
                        "id": "ID OF THE PAPER",
                        "source": "DOAJ",
                        "position": <POSITION IN RESULTS>,
                        "authors": [ LIST OF AUTHORS ],
                        "date": <DATE OF THE PAPER>,
                        "abstract": "ABSTRACT OF THE PAPER",
                        "title": "TITLE OF THE PAPER",
                        "url": "URL OF THE PAPER"
                }
        ], …
```

}

If any of the fields is missing in the DOAJ database, then the field is filled with a warning. For instance, If a paper doesn't have an "abstract" part in the DOAJ, then the "NO ABSTRACT" is written on my GET method's response.

I also implemented a POST method for our practice app. It adds a paper to a paper list of a user. Only the registered users can access this POST method. Every user can add only his/her own list. It requires a header which contains the username and password of the user and 2 parameters which are list_id and paper_id. In order to avoid conflicts, I needed two values which are unique to lists and papers.

If a user does not give either of the parameters or gives non-numeric values, list_id and paper_id become strings which cannot be converted to integers. In such cases, the POST method returns 400. If nothing goes wrong, the paper is added the paper list successfully

- Description of unit tests. You can include code fragments to illustrate.

The tester that I implemented for the GET method does the followings:

1. It directly calls the DOAJ API with the following URL: `https://doaj.org/api/search/articles/einstein,relativity?page=1&pageSize=10`
2. Checks whether the status code of the response is 200 or not
3. Calls the GET method that I implemented by the following URL:

   `http://127.0.0.1:8000/api/doaj-api/?title=einstein,relativity&rows=10`

4. Checks the status code is 200 or not
5. Checks whether the necessary fields are included in the response of the GET method I implemented
6. Compares every field of the response of my GET method to the relevant fields of the response of the GET method of the DOAJ.
7. If there is no problem, the test ends successfully.

The tester for the POST method I implemented does the followings:

1. Creates a dummy user in the setUp function
2. Creates a dummy paper list belonging to the user that we defined
3. Creates 4 papers to add to the paper list
4. Adds the papers to the paper list by the following call of my POST method:

```
headers = {'HTTP_USERNAME': "1234-5678-9012-3456",
'HTTP_PASSWORD': "strongpassword"}
```

```
response = self.client.post("/api/add-paper-to-list/",
data={'list_id': paper_list.id,'paper_id' : paper1.paper_id},
**headers)
```

5. Checks whether the status code is 200 or not
6. Checks whether the papers are in the paper list
7. If nothing goes wrong, it returns successfully

- ***Sample calls: Show a transcript of a request and response for the API functions you used and that you provide. The transcript should include the request and the detailed response (headers, status code, json etc.)***

DOAJ API

Example request:

https://doaj.org/api/search/articles/marie%2Cboltzmann?page=1&pageSize=1

Response: `{`

```
  "total": 7,

  "page": 1,

  "pageSize": 1,

  "timestamp": "2023-05-12T17:24:02.055891Z",

  "query": "marie,boltzmann",

  "results": [

    {

      "last_updated": "2022-12-21T18:08:32Z",

      "bibjson": {

        "identifier": [

          {

            "id": "10.3390/ijms21239030",

            "type": "doi"

          },

          {

            "id": "1422-0067",

            "type": "eissn"
          },
          {
            "id": "1661-6596",
            "type": "pissn"
          }
        ],
        "journal": {
          "volume": "21",
          "number": "9030",
          "country": "CH",
          "issns": [
            "1422-0067",
            "1661-6596"
          ],
          "publisher": "MDPI AG",
          "language": [
            "EN"
          ],
          "title": "International Journal of Molecular Sciences"
        },
        "month": "11",
        "keywords": [
          "Mast cell activation syndrome",
          "Hereditary alpha tryptasemia",
          "Mastocytosis",
          "IgE"
        ],
        "year": "2020",
        "start_page": "9030",
        "subject": [
          {
                "code": "QH301-705.5",
                "scheme": "LCC",
                "term": "Biology (General)"
            },
            {
                "code": "QD1-999",
                "scheme": "LCC",
                "term": "Chemistry"
            }
        ],
        "author": [
            {
                "affiliation": "Department of Internal Medicine I, Division
of Hematology and Hemostaseology and Ludwig Boltzmann Institute for Hematology
and Oncology, Medical University of Vienna, 1090 Vienna, Austria",
                "name": "Peter Valent"
            },
            {
                "affiliation": "Division of Allergy and Clinical Immunology,
University of Michigan, Ann Arbor, MI 48106, USA",
                "name": "Cem Akin"
            },
            {
                "affiliation": "Department of Dermatology, Medical
University of Gdansk, 80-211 Gdansk, Poland",
                "name": "Boguslaw Nedoszytko"
            },
            {
                "affiliation": "Allergy Unit, Verona University Hospital,
37126 Verona, Italy",
                "name": "Patrizia Bonadonna"
            },
            {
                "affiliation": "Division of Allergy, University Hospital
Basel and University of Basel, 4031 Basel, Switzerland",

                "name": "Karin Hartmann"

            },

            {

                "affiliation": "Department of Allergology, Medical
University of Gdansk, 80-211 Gdansk, Poland",

                "name": "Marek Niedoszytko"

            },

            {

                "affiliation": "Department of Dermatology and Allergy
Biederstein, Technical University of Munich, D-80802 Munich, Germany",

                "name": "Knut Brockow"

            },

            {

                "affiliation": "Dermatological Allergology, Department of
Dermatology and Allergy, Charité—Universitätsmedizin Berlin, Corporate Member
of Freie Universität Berlin, Humboldt-Universität zu Berlin, and Berlin
Institute of Health, 10117 Berlin, Germany",

                "name": "Frank Siebenhaar"

            },

            {

                "affiliation": "Division of Allergy and Clinical Immunology,
University of Salerno, 84131 Salerno, Italy",

                "name": "Massimo Triggiani"

            },

            {

                "affiliation": "Department of Hematological Biology,
Pitié-Salpêtrière Hospital, Pierre et Marie Curie University (UPMC), 75005
Paris, France",

                "name": "Michel Arock"

            },

            {

                "affiliation": "Department of Allergology, Medical
University of Gdansk, 80-211 Gdansk, Poland",

                "name": "Jan Romantowski"
            },
            {
                "affiliation": "Department of Allergology, Medical
University of Gdansk, 80-211 Gdansk, Poland",
                "name": "Aleksandra  Górska Lawrence B. Schwartz"
            },
            {
                "affiliation": "Laboratory of Allergic Diseases, NIAID, NIH,
Bethesda, MD 20852, USA",
                "name": "Dean  D. Metcalfe"
            }
        ],
        "link": [
            {
                "content_type": "text/html",
                "type": "fulltext",
                "url": "https://www.mdpi.com/1422-0067/21/23/9030"
            }
        ],
        "abstract": "Mast cell activation (MCA) is seen in a variety of
clinical contexts and pathologies, including IgE-dependent allergic
inflammation, other immunologic and inflammatory reactions, primary mast cell
(MC) disorders, and hereditary alpha tryptasemia (HAT). MCA-related symptoms
range from mild to severe to life-threatening. The severity of MCA-related
symptoms depends on a number of factors, including genetic predisposition, the
number and releasability of MCs, organs affected, and the type and consequences
of comorbid conditions. In severe systemic reactions, MCA is demonstrable by a
substantial increase of basal serum tryptase levels above the individual´s
baseline. When, in addition, the symptoms are recurrent, involve more than one
organ system, and are responsive to therapy with MC-stabilizing or
mediator-targeting drugs, the consensus criteria for the diagnosis of MCA
syndrome (MCAS) are met. Based on the etiology of MCA, patients can further be
classified as having i) primary MCAS where <i>KIT</i>-mutated, clonal, MCs are
detected; ii) secondary MCAS where an underlying IgE-dependent allergy or other
reactive MCA-triggering pathology is found; or iii) idiopathic MCAS, where
neither a triggering reactive state nor <i>KIT</i>-mutated MCs are identified.
Most severe MCA events occur in combined forms of MCAS, where
<i>KIT</i>-mutated MCs, IgE-dependent allergies and sometimes HAT are detected.
These patients may suffer from life-threatening anaphylaxis and are candidates
for combined treatment with various types of drugs, including IgE-blocking
antibodies, anti-mediator-type drugs and MC-targeting therapy. In conclusion,
detailed knowledge about the etiology, underlying pathologies and
co-morbidities is important to establish the diagnosis and develop an optimal
management plan for MCAS, following the principles of personalized medicine.",
          "title": "Diagnosis, Classification and Management of Mast Cell
Activation Syndromes (MCAS) in the Era of Personalized Medicine"
        },
        "admin": {
          "seal": true
        },
        "id": "0ee96c2cf07d4db3926fdc3928a4a09b",
        "created_date": "2020-11-28T00:06:37Z"
      }
    ],
    "next":
"https://doaj.org/api/v3/search/articles/marie%2Cboltzmann?page=2&pageSize=1",
    "last":
"https://doaj.org/api/v3/search/articles/marie%2Cboltzmann?page=7&pageSize=1"
}
```

GET METHOD

Example request:

http://127.0.0.1:8000/api/doaj-api/?title=gravitation&rows=1

Response: {"status_code": 200, "count": 1, "results": [{"id": "0016778428be48f887367f19bcd07384", "source": "DOAJ", "position": 0, "authors": ["Alexander B. Balakin", "Vladimir V. Bochkarev", "Albina F. Nizamieva"], "date": 2021, "abstract": "We consider the nonlinearly extended Einstein\u2013Maxwell-axion theory, which is based on the account for two symmetries: first, the discrete symmetry associated with the properties of the axion field, and second, the Jackson\u2019s symmetry, prescribing to the electrodynamics to be invariant with respect to the rotation in the plane coordinated by the electric and magnetic fields. We derive the master equations of the nonlinearly extended theory and apply them to the Bianchi-I model with magnetic field. The main result, describing the behavior of the nonlinearly coupled axion, electromagnetic, and gravitational fields is the anomalous growth of the axionically induced electric field in the early magnetized Universe. The character of behavior of this anomalous electric field can be indicated by the term flare. We expect, that these electric flares can produce the electron\u2013positron pair creation, significant acceleration of the born charged particles, and the emission of the electromagnetic waves by these accelerated particles.", "title": "Nonlinear Axion Electrodynamics: Axionically Induced Electric Flares in the Early Magnetized Universe", "url": "https://www.mdpi.com/2073-8994/13/11/2038"}]}

POST METHOD

Example request: http://127.0.0.1:8000/api/add-paper-to-list/

headers = {“username”: “mehmet”, “password”: “123”}

body = {“list_id”=1, “paper_id”=1}

Response:

If the user is not authorized:

{ “status”: “user credentials are incorrect” **}** status_code=401

If there is no paper with paper_id=1:

{“status”: No such Paper”} status_code=400

If there is no paper list with list_id=1:

{“status”: No such List”} status_code=400

If it completes successfully:

{“status”: “Paper Has Been Added To The List”} status_code = 200

- Any other significant work related to this release (such as Docker, AWS, DB, Framework)

I designed a template for the UI. I created search paper, search user, my lists, follow requests, following lists, and sign-in pages. These pages include buttons, single-lined text fields, tables, and hyperlinks to the other page pages of the web application we have developed. Frontend development continued on top of my implementation. It uses bootstrap to increase the quality of the pages. Additionally, I designed the profile page of users which includes username, first name, last name, date joined, and interests.

- Describe any challenges you met during implementation.

I knew nothing about Django and Web Application Development before this project. I learned the details of the structure of Django, API development, frontend development, and Django databases. Implementation of the GET method was exceptionally difficult for me since I don’t know how to parse the parameters from the URL. Additionally, some papers in the DOAJ database don't have some fields such as “title” and “abstract”. Handling such things took considerable time.

Ömer Şükrü Uyduran

- Links to important issues (git issues) and wiki documentation related to practice application
Some assigned issues:
Meeting #9 Notes will be added
CORE.AC.UK API Research - Ömer Şükrü Uyduran
CORE.AC.UK API GET Implementation
Design and Implementation of Sign-in Page
Design and Implementation of My Paper Lists Page
POST method to Create Paper List
Documentation of the GET method which utilizes CORE.AC.UK
Documentation of the POST method for creating paper lists
Member Contribution - Ömer Şükrü Uyduran

Some opened issues:
Meeting #10 Notes will be added
Front-end of the practice application will be implemented
Design and Implementation of Sign-up Page
Dockerization and deployment of practice app
Meeting #12 Notes will be added
Documentation of APIs
Subsection of team members in the milestone 2 report

Some documentation in wiki:
Meeting #9 | 19.04.2023
I also documented the APIs I implemented in the postman workspace.

- The name, route, description of utilized third party URIs
I have utilized CORE.AC.UK API for this practice app. The URL is:
https://api.core.ac.uk/v3/search/works?q=<keyword>&limit=<result_number>
This request looks for papers in CORE.AC.UK database that is related to the specified keyword. It returns the first <result_number> results.
Also authorization is needed to use this API. An API key, which can be easily gotten from core.ac.uk website, must be provided in the headers in this form: “Authorization”: “Bearer <api_key>”.
- The name, route, description of the created API functions
I have implemented 2 API functions for the practice app: A GET method which uses the CORE.AC.UK API in itself and a POST method to create paper lists.
GET Method:
/api/core/?title=<keyword>&rows=<result_num>
This end point is for allowing users to search papers on CORE.AC.UK database via our servers.

title param must be provided and used searching for specific keywords. Also the row param can be specified to limit the returning results (optional). Default is 3 for row when not specified.
The method takes the request, fetches the params, and sends a request to the CORE.AC.UK API.
The request returns 400 if the title parameter is missing or the row parameter is non-numeric.
It returns 404 if there is no matching paper with the specified keyword.
It returns 204 if the rate-limit of the CORE.AC.UK API is 0.
It returns 500 if an internal server error exists or the server couldn't get the info from CORE.AC.UK.
It returns 200 for success.
No authentication needed to use this GET method.

POST Method:

This end point is for allowing users to create new paper lists in the database.
Authentication needed: username and password should be supplied in the header
list_title should be supplied in the body form-data.
The method takes the request; checks the list title, username, and password; inserts the new data to the database; and returns the respnose.
The request returns 407 if the suername and the password did not supplied.
It returns 401 if the credentials are invalid.
It returns 400 if the list_title did not provided.
It returns 200 for success.

- Description of unit tests. You can include code fragments to illustrate.
Tests for the GET Method:
I have created test cases for my GET method under the core_api_test_cases(TestCase) class in tests.py file. I wrote 2 functions: the first one tests the unexpected results and the second one tests the expected results.
In the first one, I tested the missing title param cases and non-numeric rows param cases. I also tested the case which is correct however there is no data related to the specified keyword in the CORE.AC.UK database.
In the second one, I tested the successful requests with and without row param. I checked some of the fields of the results also.
Tests for the POST Method:
I have created test cases for my POST method under the create_paper_list_test_cases(TestCase) class in tests.py file. I created a user in the setUp function to test the API. I tested the cases: the case that no credentials supplied, the case that has invalid credentials, the case that the list_title is missing, and the success case.
The source code can be found in the practice_app/api/tests.py file.

- Sample calls: Show a transcript of a request and response for the API functions you used

and that you provide. The transcript should include the request and the detailed response (headers, status code, json etc.)
Utilized API:

Request to URL:
https://api.core.ac.uk/v3/search/works?q=hardware%20accelerators&limit=1

with headers: {“Authorization”: “Bearer <api_key>”}

Responds with status code 200:

```
{
   "totalHits": 26994,
   "limit": "1",
   "offset": 0,
   "scrollId": null,
   "results": [
      {
         "acceptedDate": null,
         "arxivId": null,
         "authors": [
            {
               "name": "Colle, Didier"
            },
            {
               "name": "Pickavet, Mario"
            },
            {
               "name": "Sharma, Gourav Prateek"
            },
            {
               "name": "Tavernier, Wouter"
            }
         ],
         "citationCount": null,
         "contributors": [
            "Gourav Prateek"
         ],
         "outputs": [
            "https://api.core.ac.uk/v3/outputs/472912667",
            "https://api.core.ac.uk/v3/outputs/323232513"
         ],
         "createdDate": "2020-05-27T16:13:06",
         "dataProviders": [
        {
          "id": 4786,
          "name": "",
          "url": "https://api.core.ac.uk/v3/data-providers/4786",
          "logo": "https://api.core.ac.uk/data-providers/4786/logo"
        },
        {
          "id": 1493,
          "name": "",
          "url": "https://api.core.ac.uk/v3/data-providers/1493",
          "logo": "https://api.core.ac.uk/data-providers/1493/logo"
        }
      ],
      "depositedDate": null,
      "abstract": "Hardware-accelerators in Network Function Virtualization (NFV)
environments have aided telecommunications companies (telcos) to reduce their expenditures
by offloading compute-intensive VNFs to hardware-accelerators. To fully utilize the benefits
of hardware-accelerators, VNF-chain recovery models need to be adapted. In this paper, we
present an ILP model for optimizing prioritized recovery of VNF-chains in heterogeneous
NFV environments following node failures. We also propose an accelerator-aware heuristic
for solving prioritized VNF-chain recovery problems of large-size in a reasonable time.
Evaluation results show that the performance of heuristic matches with that of ILP in regard
to restoration of high and medium priority VNF-chains and a small penalty occurs only for
low-priority VNF-chains",
      "documentType": "research",
      "doi": "10.1109/drcn48652.2020.1570604216",
      "downloadUrl": "https://core.ac.uk/download/323232513.pdf",
      "fieldOfStudy": null,
      "fullText": <too_long_to_paste_here>,
      "id": 85951287,
      "identifiers": [
        {
          "identifier": "oai:archive.ugent.be:8662296",
          "type": "OAI_ID"
        },
        {
          "identifier": "10.1109/drcn48652.2020.1570604216",
          "type": "DOI"
        },
        {
          "identifier": "472912667",
          "type": "CORE_ID"
        },
        {
            "identifier": "323232513",
            "type": "CORE_ID"
        }
    ],
    "title": "Hardware-accelerator aware VNF-chain recovery",
    "language": {
        "code": "en",
        "name": "English"
    },
    "magId": null,
    "oaiIds": [
        "oai:archive.ugent.be:8662296"
    ],
    "publishedDate": "2020-01-01T00:00:00",
    "publisher": "'Institute of Electrical and Electronics Engineers (IEEE)'",
    "pubmedId": null,
    "references": [],
    "sourceFulltextUrls": [
        "https://biblio.ugent.be/publication/8662296/file/8662298.pdf"
    ],
    "updatedDate": "2021-08-10T22:03:31",
    "yearPublished": 2020,
    "journals": [],
    "links": [
        {
            "type": "download",
            "url": "https://core.ac.uk/download/323232513.pdf"
        },
        {
            "type": "reader",
            "url": "https://core.ac.uk/reader/323232513"
        },
        {
            "type": "thumbnail_m",
            "url": "https://core.ac.uk/image/323232513/medium"
        },
        {
            "type": "thumbnail_l",
            "url": "https://core.ac.uk/image/323232513/large"
        },
        {
            "type": "display",
            "url": "https://core.ac.uk/works/85951287"
        }
```

```
            ]
        }
    ],
    "tooks": null,
    "esTook": null
}
```

with headers:

```
{
  "Date":"Fri, 12 May 2023 17:24:39 GMT",
  "Content-Type":"application/json",
  "Content-Length":"10581",
  "Connection":"keep-alive",

"Set-Cookie":"AWSALBTG=TNStXpJwm5D4a2VlwuFdoslc/g2me5EUyJGHQ0PdtsPoAQL
e/WZfxYd3PdJGhteijCVrThPBQOoAp8b41volpbusrWXSCGvMMauqk3jZeCK0KhSjxN3x
Tu8T7QuI6j3+2CWYxEKs+6cEDLKpY5eYDS6F2hHzI9RVTNnpZNP0u7GnR8fp/8Q=;
Expires=Fri, 19 May 2023 17:24:39 GMT; Path=/,
AWSALBTGCORS=TNStXpJwm5D4a2VlwuFdoslc/g2me5EUyJGHQ0PdtsPoAQLe/WZfx
Yd3PdJGhteijCVrThPBQOoAp8b41volpbusrWXSCGvMMauqk3jZeCK0KhSjxN3xTu8T7Q
uI6j3+2CWYxEKs+6cEDLKpY5eYDS6F2hHzI9RVTNnpZNP0u7GnR8fp/8Q=;
Expires=Fri, 19 May 2023 17:24:39 GMT; Path=/; SameSite=None; Secure,
AWSALB=D4tSp7H+BexlWq/59JgxHssI0qS1p+ylB5kC/doKhGvCrsCE6PRlBCHs8nmC5a
2JR/AJqbuAdA/VaXjXl1A7sdDGlJF3Ec/LYFs2M78YLkZvFod57LWGt9+GrR5r;
Expires=Fri, 19 May 2023 17:24:39 GMT; Path=/,
AWSALBCORS=D4tSp7H+BexlWq/59JgxHssI0qS1p+ylB5kC/doKhGvCrsCE6PRlBCHs8n
mC5a2JR/AJqbuAdA/VaXjXl1A7sdDGlJF3Ec/LYFs2M78YLkZvFod57LWGt9+GrR5r;
Expires=Fri, 19 May 2023 17:24:39 GMT; Path=/; SameSite=None; Secure",
  "Server":"Apache",
  "Cache-Control":"no-cache, private",
  "Symfony-Session-NoAutoCacheControl":"true",
  "X-RateLimit-Remaining":"8",
  "X-RateLimit-Retry-After":"2023-05-12T17:24:39+0000",
  "X-RateLimit-Limit":"10",
  "Strict-Transport-Security":"max-age=15560640; includeSubDomains; preload",
  "X-Frame-Options":"SAMEORIGIN",
  "Vary":"User-Agent,Referer,Authorization,Accept-Encoding",
  "Content-Encoding":"gzip",
  "X-Clacks-Overhead":"GNU Terry Pratchett, GNU Ashley Sanders"
}
```

My GET Method:

Request to URL: http://localhost:8000/api/core/?title=vision%20transformers&rows=1

Responds with status code 200:

```
{
   "status_code": 200,
   "results": [
      {
         "title": "LaCViT: A Label-aware Contrastive Training Framework for Vision\n Transformers",
         "authors": [
            "Camarasa, Gerardo Aragon",
            "Long, Zijun",
            "McCreadie, Richard",
            "Meng, Zaiqiao"
         ],
         "source": "core.ac.uk",
         "id": 142114723,
         "date": "2023-03-31T01:00:00",
         "url": "http://arxiv.org/abs/2303.18013",
         "position": 0,
         "abstract": "Vision Transformers have been incredibly effective when tackling computer\nvision tasks due to their ability to model long feature dependencies. By using\nlarge-scale training data and various self-supervised signals (e.g., masked\nrandom patches), vision transformers provide state-of-the-art performance on\nseveral benchmarking datasets, such as ImageNet-1k and CIFAR-10. However, these\nvision transformers pretrained over general large-scale image corpora could\nonly produce an anisotropic representation space, limiting their\ngeneralizability and transferability to the target downstream tasks. In this\npaper, we propose a simple and effective Label-aware Contrastive Training\nframework LaCViT, which improves the isotropy of the pretrained representation\nspace for vision transformers, thereby enabling more effective transfer\nlearning amongst a wide range of image classification tasks. Through\nexperimentation over five standard image classification datasets, we\ndemonstrate that LaCViT-trained models outperform the original pretrained\nbaselines by around 9% absolute Accuracy@1, and consistent improvements can be\nobserved when applying LaCViT to our three evaluated vision transformers"
      }
   ]
}
```

My POST Method:

Request to URL: http://localhost:8000/api/create-paper-list/
With headers: {“username”: “<a username from database>”, “password”: “<the password of the user with the same username>”}

With body: {“list_title”: “my new list”}

Responds with status code 200:
```
{
   "status": "Paper list created successfully!"
}
```

- Any other significant work related to this release (such as Docker, AWS, DB, Framework)
I have done research on how to use the Django framework in the front-end and found out some of the necessary techniques. I also implemented some of the front-end pages. I also reviewed and tested a lot of pull requests and fix some of the problems in them.

- Describe any challenges you met during implementation.
At the very beginning, it was so hard to work on the same project with such a big team. Everything was a mess for me since everyone was doing something and everything went fast. However, I got very comfortable and better as we proceed. At the beginning, for me, it was hard to use django properly and to ask for help from my teammates for anything. As we work on it together, these got easier too.

Leyla Yayladere
important issues:
- Third Party API Search - Elsevier ScienceDirect #104
- Implementation of GET Methods - X-ELS API #124
- Third Party API Search - STI Repository OpenAPI #161
- Implementation of GET Methods - NASA STI OpenApi #162
- Implementation of POST Methods - Endpoint to Send Follow Request #150
- Front-end - Design and Implementation of Follow Request Page #160
- Updating API URLs to use hyphens for word separation #157

The name, route, description of utilized third party URIs
Since I encountered issues while working with my previous API (X-ELS API for Elsevier ScienceDirect journal), I opted to switch to another one. I have decided to utilize the NASA STI OpenAPI to search for articles in NASA Scientific and Technical Information (STI) Repository. I picked it because I had limited time and it doesn't require any authentication model and its features are aligned with our requirements, i.e it has REST architectural style, supports JSON response format, and useful for our practice-app in terms of enhancing the database where papers are being searched.
Their documentation:
https://www.sti.nasa.gov/docs/20221122_OpenAPI_Documentation.pdf
Their website: https://ntrs.nasa.gov/api/openapi/
Route to API: https://ntrs.nasa.gov/api/openapi/search
Example: https://ntrs.nasa.gov/api/citations/search?abstract=space&page.size=5

The name, route, description of the created API functions
- **GET method : NASA-STI:**

This endpoint allows users to retrieve the details of a specific paper in the NASA-STI database. This function takes 2 parameters called ‘title’ and ‘rows’. Users can also specify the number of results by setting 'rows' parameter. Default number of results returned is 3.
When a search is successful, it returns a JSON response with information such as id, title, url to original paper, authors, abstract, published date, source of found papers.
If ‘title' param is empty or null, it throws a 400 error with the message '*Title to search must be given'*. If 'rows' param is empty, null, or nonnumeric, it returns 200 status code. It throws a 404 error with the message '*Unsuccessful Search*.' when NASA-STI API returns a 404 error. It returns 503 '*An internal server error has occurred. Please try again.*' when there is an internal server error.
No authentication is needed to access this endpoint.
route: /api/nasa-sti/

- **POST method : Send a follow request:**
  This function allows users to follow another user . It takes 1 parameter named followed_username in the request body as form-data.
  Authentication is needed to access this endpoint. User should be registered or authenticated through its username and password in the headers.
  If there is a conflict such as the user to be sent a follow request is already followed or the request was sent previously, it throws a 409 error with the message accordingly.
  If 'followed_username' param is invalid, it throws a 404 error. If the 'followed_username' param is empty or null, it throws 400 errors. If the user is anonymous and the headers don't contain authentication info (username and password), it throws a 407 error. Else it contains authentication info but incorrect credentials, then it throws 401 error.
  route: /api/follow-user/

Description of unit tests. You can include code fragments to illustrate.

Unit tests for the GET method works as follows.

First, it sets up a client object which is used to make requests. I considered the different cases that can occur during an API request. Firstly, I set incorrect URLs to test whether invalid cases were correctly caught. An error JSON response is expected in such cases. I also tested valid cases and checked if the results matched expectations. I used assertions to verify the status code, message, or response fields.

Here is a snippet:

```
class NasaStiTestCase(TestCase):

   def setUp(self):

       self.client = Client()

   def tearDown(self):

       print('GET Tests of Nasa STI Completed Successfully')

   def test_4xx_responses(self):
       self.assertEquals(self.client.get("/api/nasa-sti/?title=").status_code, 400)

       self.assertEquals(self.client.get("/api/nasa-sti/?").status_code, 400)

       self.assertEquals(self.client.get("/api/nasa-sti/").status_code, 400)

       self.assertEquals(self.client.get("/api/nasa-sti/?rows=9").status_code, 400)

       self.assertEquals(self.client.get("/api/nasa-sti/title=space").status_code,
404)

       self.assertEquals(self.client.get("/api/nasa-sti/?title=&").status_code,
400)

   def test_valid_title_valid_rows(self):

       # test when valid title and rows are provided

       field_count = 8

       rows = 10

       response = self.client.get('/api/nasa-sti/?title=space&rows='+str(rows))

       self.assertEqual(response.status_code, 200)

       self.assertEqual(len(response.json()['results'][0]), field_count)

       self.assertEqual(len(response.json()['results']), rows)

       self.assertContains(response, 'id')

       self.assertContains(response, 'title')

       self.assertContains(response, 'authors')

       self.assertContains(response, 'abstract')

       self.assertContains(response, 'source')

       self.assertContains(response, 'date')

       self.assertContains(response, 'url')

       self.assertContains(response, 'position')
```

Unit tests for the POST method works as follows:

First it sets up a client object for requests. Also it creates three different user and two followed objects since the objective of the post method is following another user. For this method, I mainly considered four different cases. The first case is where the user is authenticated/logged in and the provided followed_username is valid. User sends following request two times to the same user. Here first a successful response, then unsuccessful is expected. Another case is trying to send a follow request to already followed user which should throw a 409 conflict error. I also considered other invalid cases such as the user is not authenticated because of either empty or invalid credential information, but they're not in the code snippet.

Here is a snippet:

```
class FollowUserTestCase(TestCase):

   def setUp(self):

       self.c = Client()

       user_1 = User.objects.create_user(username="0009-0005-5924-1831",
password="strongpassword", first_name = "follower", last_name = "follower")

       user_2 = User.objects.create_user(username="0009-0005-5924-1832",
password="strongpassword", first_name = "followed_pending", last_name =
"followed_pending")

       user_3 = User.objects.create_user(username="0009-0005-5924-1833",
password="strongpassword", first_name = "followed_approved", last_name =
"followed_approved")

       user_1_follower = models.Follower.objects.create(user=user_1)

       user_1_follower.followed.add(user_3)

       user_3_follower = models.Follower.objects.create(user=user_3)

       user_3_follower.follower.add(user_1)

   def tearDown(self):

       print('Tests for POST requests using follow_user completed!')

def test_valid_and_invalid_follow_cases(self):

       # follow request is sent for the first time

       Headers = {'username': "0009-0005-5924-1831", "password": "strongpassword"}

       Body = {'followed_username':'0009-0005-5924-1832'}

       response = self.c.post("/api/follow-user/", headers = Headers, data = Body)

       self.assertEqual(response.status_code, 200)

       self.assertEqual(response.json()['status'], "User followed.")

       # follow request is sent for the second time

       response = self.c.post("/api/follow-user/", headers = Headers, data = Body)

       self.assertEqual(response.status_code, 409)

       self.assertEqual(response.json()['status'], "You have already sent a
following request this user.")

   def test_already_follower_cases(self):

       # follower_user is already following followed_user

       Headers = {'username': "0009-0005-5924-1831", "password": "strongpassword"}
```

```
        Body = {'followed_username':'0009-0005-5924-1833'}

        response = self.c.post("/api/follow-user/", headers = Headers, data = Body)

        self.assertEqual(response.status_code, 409)

        self.assertEqual(response.json()['status'], "You are already following this
user.")
```

Sample calls:

API I Utilized:

- request: https://ntrs.nasa.gov/api/citations/search?abstract=space&page.size=2
- response:
{"stats":{"took":105,"total":90683,"estimate":false,"maxScore":5.110316},"results":[{"_meta":{"score":5.110316},"legacyMeta":{"__type":"LegacyMetaIndex, StrivesApi.ServiceModel","accessionNumber":"86A17301"},"copyright":{"licenseType":"NO","determinationType":"OTHER","thirdPartyContentCondition":"NOT_SET"},"subjectCategories":["Astronautics (General)"],"exportControl":{"isExportControl":"NO","ear":"NO","itar":"NO"},"created":"2013-08-12T19:03:00.0000000+00:00","distributionDate":"2019-08-27T08:39:11.5930000+00:00","center":{"code":"CDMS","name":"Legacy CDMS","id":"092d6e0881874968859b972d39a888dc"},"onlyAbstract":false,"sensitiveInformation":2,"abstract":"Topics discussed include space infrastructures and early Space Station and platform planning. Consideration is given to the supportive role of the Space Station and platform in future astronomy, earth observation, planetary, and communication space missions. Papers are presented on the history of the Space Station and space platform concepts, potential designs of space stations and space platforms, and long-range plans for space research.","title":"Space stations and space platforms - Concepts, design, infrastructure, and uses","stiType":"OTHER","distribution":"PUBLIC","submittedDate":"2013-08-12T19:03:00.0000000+00:00","isLessonsLearned":false,"authorAffiliations":[{"sequence":0,"submissionId":19860032563,"meta":{"author":{"name":"Bekey, I."},"organization":{"name":"NASA Headquarters","location":"Washington, DC United States"}},"id":"9aca9a8f57e74ab68821c934485cb61d"},{"sequence":1,"submissionId":19860032563,"meta":{"author":{"name":"Herman, D."},"organization":{"name":"NASA Headquarters","location":"Washington, DC United States"}},"id":"c08514e77f964e3f8529a6eb14218e1c"}],"disseminated":"METADATA_ONLY","stiTypeDetails":"Other - Collected Works","technicalReviewType":"TECHNICAL_REVIEW_TYPE_NONE","modified":"2022-11-19T07:45:26.3393150+00:00","id":19860032563,"publications":[{"submissionId":19860032563,"id":"2d14e7388bd8495793dca4bd992e9d9d","publicationDate":"1985-01-01T00:00:00.0000000+00:00"}],"status":"CURATED","related":[],"downloads":[],"downloadsAvailable":false,"index":"submissions-2023-05-12-05-39"},{"_meta":{"score":5.100472},"copyright":{"licenseType":"NO","determinationType":"NO_PERMISSION","thirdPartyContentCondition":"NOT_SET"},"subjectCategories":["Astronautics (General)"],"keywords":["Space life support systems","Aerospace engineering","Construction engineering","Space colonies","Space structures","Space exploration","Engineering education","Moon"],"exportControl":{"isExportControl":"NO","ear":"NO","itar":"NO"},"distributionDate":"2019-08-27T08:38:19.5000000+00:00","fundingNumbers":[],"title":"Engineering, Construction, and Operations in Space III","stiType":"CONFERENCE_PROCEEDINGS","distribution":"PUBLIC","submittedDate":"2013-08-16T01:40:00.0000000+00:00","authorAffiliations":[{"organizationId":"c88218f17b635833817172a563399015","sequence":0,"submissionId":19930057979,"meta":{"author":{"name":"Willy Z. Sadeh"},"organization":{"name":"Colorado State University","location":"Fort Collins, Colorado, United States"}},"id":"13b651b8e6094684a904a795d7ef7ec4","userType":"OTHER"},{"organizationId":"8f7454239bfc5883a27b8bc72d648415","sequence":1,"submissionId":19930057979,"meta":{"author":{"orcidId":"","name":"Stein Sture"},"organization":{"name":"University of Colorado Boulder","location":"Boulder, Colorado, United States"}},"id":"c45ea7f0c21e4a1b8dd8fc2cd4a0c4be","userType":"OTHER","userId":"7f6923a40cee45a097

1342544edaad96"},{"organizationId":"54c72687f6765050b5d52bca86beb2e8","sequence":2,"submissionId":19930057979,"meta":{"author":{"name":"Russell J. Miller"},"organization":{"name":"Colorado School of Mines","location":"Golden, Colorado, United States"}},"id":"7b64d06e10d042bcae7a860c02ede5f6","userType":"OTHER"}],"stiTypeDetails":"Conference Proceedings","technicalReviewType":"TECHNICAL_REVIEW_TYPE_NONE","modified":"2020-10-21T16:15:57.2228540+00:00","id":19930057979,"sourceIdentifiers":[],"legacyMeta":{"__type":"LegacyMetaIndex, StrivesApi.ServiceModel","accessionNumber":"93A41976"},"created":"2013-08-16T01:40:00.0000000+00:00","center":{"code":"JSC","name":"Johnson Space Center","id":"dd46994b91a94cb2a2722617cbbadc55"},"onlyAbstract":true,"sensitiveInformation":2,"abstract":"The present volume on engineering, construction, and operations in space discusses surface structures on the moon and Mars, surface equipment, construction, and transportation on the moon and Mars, in situ materials use and processing, and space energy. Attention is given to such orbital structures as LEO and the space station, space mining and excavation, space materials, space automation and robotics, and space life support systems. Topics addressed include lunar-based astronomy, space systems integration, terrestrial support for space functions, and space education. Also discussed are space plans, policy, and history, space science and engineering, geoengineering and space exploration, and the construction and development of a human habitat on Mars.","isLessonsLearned":false,"disseminated":"METADATA_ONLY","meetings":[{"country":"US","submissionId":19930057979,"endDate":"1992-06-04T04:00:00.0000000+00:00","sponsors":[{"organizationId":"6457a6ab23c95056a4fd2df12a5908a5","meta":{"organization":{"name":"American Society of Civil Engineers","location":"Reston, Virginia, United States"}},"meetingId":"e92f9bbe917145e09999d182e4461ede","id":"d2b0ae96d11749f98b1fd861bd69d368"}],"name":"3rd International Conference (Space '92)","location":"Denver, CO","id":"e92f9bbe917145e09999d182e4461ede","startDate":"1992-05-31T04:00:00.0000000+00:00"}],"publications":[{"volume":"1 & 2","submissionId":19930057979,"isbn":"978-0872628687","publisher":"American Society of Civil Engineers","id":"e872b3e722f345ad95bc79b0523621c4","publicationName":"Proceedings of the 3rd International Conference (Space '92)","publicationDate":"1992-01-01T05:00:00.0000000+00:00"}],"status":"CURATED","related":[{"disseminated":"METADATA_ONLY","id":19930057983,"type":"ANALYTIC_SUBSIDIARY","title":"Human exploration of Mars - The role of a Mars outpost laboratory","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930057985,"type":"ANALYTIC_SUBSIDIARY","title":"An analysis of an inflatable module for planetary surfaces","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930057986,"type":"ANALYTIC_SUBSIDIARY","title":"Construction and development of a human base on Mars","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930057990,"type":"ANALYTIC_SUBSIDIARY","title":"A horizontal inflatable habitat for SEI","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930057995,"type":"ANALYTIC_SUBSIDIARY","title":"Experimental, physical and numerical modeling of lunar regolith and lunar regolith structures","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058000,"type":"ANALYTIC_SUBSIDIARY","title":"Design and technology assessment of three lunar habitat concepts","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058004,"type":"ANALYTIC_SUBSIDIARY","title":"Structural characterization of an articulated-truss joint","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058006,"type":"ANALYTIC_SUBSIDIARY","title":"Analysis of space crane articulated-truss joints","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058009,"type":"ANALYTIC_SUBSIDIARY","title":"The challenge of constraining mass for planetary construction","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058010,"type":"ANALYTIC_SUBSIDIARY","title":"Directions for lunar construction - A derivation of requirements from a construction scenario analysis","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058014,"type":"ANALYTIC_SUBSIDIARY","title":"Dust control research for SEI","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058020,"type":"ANALYTIC_SUBSIDIARY","title":"The application of Open System Architecture to planetary surface systems","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058025,"type":"ANALYTIC_SUBSIDIARY","title":"Structural materials from lunar simulants through thermal liquefaction","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058026,"type":"ANALYTIC_SUBSIDIARY","title":"In-situ release of solar wind gases from lunar soil","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058032,"type":"ANALYTIC_SUBSIDIARY","title":"Environmental aspects of lunar helium-3 mining","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058034,"type":"ANALYTIC_SUBSIDIARY","title":"Lunar oxygen - The reduction of glass by hydrogen","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058035,"type":"ANALYTIC_SUBSIDIARY","title":"A modified sulfate process to lunar oxygen","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058036,"type":"ANALYTIC_SUBSIDIARY","title":"Steady state composition with low Fe(2+) concentrations for efficient O2 production by 'magma' electrolysis of lunar soils","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058040,"type":"ANALYTIC_SUBSIDIARY","title":"Materials and structure synergistic with in-space materials utilization","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058043,"type":"ANALYTIC_SUBSIDIARY","title":"The feasibility of processes for the production of oxygen on the moon","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058046,"type":"ANALYTIC_SUBSIDIARY","title":"A Mars 1 Watt vortex wind energy machine","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058054,"type":"ANALYTIC_SUBSIDIARY","title":"Structural considerations in the design of a Mars mission aerobrake","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058056,"type":"ANALYTIC_SUBSIDIARY","title":"Tethers and their role in the Space Exploration Initiative","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058058,"type":"ANALYTIC_SUBSIDIARY","title":"Structural studies of two aerobrake heatshield panel concepts","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058059,"type":"ANALYTIC_SUBSIDIARY","title":"Modeling and analysis of doubly curved aerobrake truss structures","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058063,"type":"ANALYTIC_SUBSIDIARY","title":"Evolution of the Space Station Freedom module pattern","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058070,"type":"ANALYTIC_SUBSIDIARY","title":"Beneficiation of lunar rocks and regolith - Concepts and difficulties","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058071,"type":"ANALYTIC_SUBSIDIARY","title":"Mobile continuous lunar excavation","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058078,"type":"ANALYTIC_SUBSIDIARY","title":"Transfer of terrestrial technology for lunar mining","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058081,"type":"ANALYTIC_SUBSIDIARY","title":"Design criteria for an underground lunar mine","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058084,"type":"ANALYTIC_SUBSIDIARY","title":"Vacuum melting and mechanical testing of simulated lunar glasses","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058085,"type":"ANALYTIC_SUBSIDIARY","title":"Strength and fracture of glass in the lunar environment","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058086,"type":"ANALYTIC_SUBSIDIARY","title":"Mechanical properties of compacted lunar simulant using new vacuum triaxial equipment","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058088,"type":"ANALYTIC_SUBSIDIARY","title":"The effect of multiple compliant layers at the fiber-matrix interface on residual thermal stresses in metal matrix composites","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058091,"type":"ANALYTIC_SUBSIDIARY","title":"Composite materials for structures on planetary surfaces","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058092,"type":"ANALYTIC_SUBSIDIARY","title":"Comparison of micromechanical models for elastic properties","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058099,"type":"ANALYTIC_SUBSIDIARY","title":"Combustion synthesis of advanced materials","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058103,"type":"ANALYTIC_SUBSIDIARY","title":"Telerobotic field geologist - Preliminary results of a feasibility study","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058108,"type":"ANALYTIC_SUBSIDIARY","title":"The initial exploration of Mars - Rationale for a return mission to Chryse Planitia and the Viking 1 Lander","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058109,"type":"ANALYTIC_SUBSIDIARY","title":"Operations planning for Space Station Freedom - And beyond","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058111,"type":"ANALYTIC_SUBSIDIARY","title":"The Virtual Mission - A step-wise approach to large space missions","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058114,"type":"ANALYTIC_SUBSIDIARY","title":"Advanced construction management for lunar base construction - Surface operations planner","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058116,"type":"ANALYTIC_SUBSIDIARY","title":"Architectures for mission control at the Jet Propulsion Laboratory","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058118,"type":"ANALYTIC_SUBSIDIARY","title":"Operations analysis for a large lunar telescope","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058121,"type":"ANALYTIC_SUBSIDIARY","title":"A facility for training Space Station astronauts","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058122,"type":"ANALYTIC_SUBSIDIARY","title":"EVA operational guidelines and considerations for use during the Space Station Freedom design review process","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058123,"type":"ANALYTIC_SUBSIDIARY","title":"Assessment of a SSF servicing facility","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058127,"type":"ANALYTIC_SUBSIDIARY","title":"Utilization of on-site resources for Regenerative Life Support Systems at a lunar outpost","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058131,"type":"ANALYTIC_SUBSIDIARY","title":"Regenerative life support technology challenges for the Space Exploration Initiative","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058134,"type":"ANALYTIC_SUBSIDIARY","title":"NASA's future plans for space astronomy and astrophysics","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058135,"type":"ANALYTIC_SUBSIDIARY","title":"The proposed NASA lunar-based astronomical observatories","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058136,"type":"ANALYTIC_SUBSIDIARY","title":"System concepts for a series of lunar optical telescopes","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058137,"type":"ANALYTIC_SUBSIDIARY","title":"Mitigation of adverse environmental effects on lunar-based astronomical instruments","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058138,"type":"ANALYTIC_SUBSIDIARY","title":"Thermal investigation of a large lunar telescope","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058139,"type":"ANALYTIC_SUBSIDIARY","title":"Developing technologies for lunar-based astronomy","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058140,"type":"ANALYTIC_SUBSIDIARY","title":"The Lunar Transit Telescope (LTT) - An early lunar-based science and engineering mission","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058141,"type":"ANALYTIC_SUBSIDIARY","title":"Lunar transit telescope lander design","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058143,"type":"ANALYTIC_SUBSIDIARY","title":"SALSA - A lunar submillimeter-wavelength array","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058144,"type":"ANALYTIC_SUBSIDIARY","title":"Low frequency astronomy from lunar orbit","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058146,"type":"ANALYTIC_SUBSIDIARY","title":"Concept for a lunar array for very low frequency radio astronomy","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058147,"type":"ANALYTIC_SUBSIDIARY","title":"Laboratory evaluation of footings for lunar telescopes","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058148,"type":"ANALYTIC_SUBSIDIARY","title":"Design of a support and foundation for a large lunar optical telescope","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058150,"type":"ANALYTIC_SUBSIDIARY","title":"Systems integration of lunar Campsite vehicles","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058151,"type":"ANALYTIC_SUBSIDIARY","title":"Spaceborne construction and operations planning - Decision rules for selecting EVA, telerobot, and combined work-systems","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058159,"type":"ANALYTIC_SUBSIDIARY","title":"Global change - Geoengineering and space exploration","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058160,"type":"ANALYTIC_SUBSIDIARY","title":"Plot-scale field experiment of surface hydrologic processes with EOS implications","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058164,"type":"ANALYTIC_SUBSIDIARY","title":"Space Civil Engineering option - A progress report","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058168,"type":"ANALYTIC_SUBSIDIARY","title":"A vision for planetary exploration","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058176,"type":"ANALYTIC_SUBSIDIARY","title":"An analysis of human performance in simulated partial-gravity environments","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058188,"type":"ANALYTIC_SUBSIDIARY","title":"Apollo 11 ilmenite revisited","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"},{"disseminated":"METADATA_ONLY","id":19930058083,"type":"ANALYTIC_SUBSIDIARY","title":"Sintering of Lunar Glass and Basalt","stiType":"CONFERENCE_PAPER","distribution":"PUBLIC","status":"CURATED"}],"downloads":[],"downloadsAvailable":false,"index":"submissions-2023-05-12-05-39"}],"aggregations":{"created":{"buckets":[{"key_as_string":"2013","key":1356998400000,"doc_count":74712},{"key_as_string":"2015","key":1420070400000,"doc_count":2002},{"key_as_string":"2014","key":1388534400000,"doc_count":1911},{"key_as_string":"2017","key":1483228800000,"doc_count":1801},{"key_as_string":"2016","key":1451606400000,"doc_count":1708},{"key_as_string":"2019","key":1546300800000,"doc_count":1678},{"key_as_string":"2022","key":1640995200000,"doc_count":1489},{"key_as_string":"2020","key":1577836800000,"doc_count":1484},{"key_as_string":"2018","key":1514764800000,"doc_count":1446},{"key_as_string":"2021","key":1609459200000,"doc_count":1388},{"key_as_string":"2023","key":1672531200000,"doc_count":407},{"key_as_string":"1999","key":915148800000,"doc_count":98},{"key_as_string":"1998","key":883612800000,"doc_count":63},{"key_as_string":"1993","key":725846400000,"doc_count":61},{"key_as_string":"1996","key":820454400000,"doc_count":57},{"key_as_string":"2000","key":946684800000,"doc_count":56},{"key_as_string":"1994","key":757382400000,"doc_count":47},{"key_as_string":"1997","key":852076800000,"doc_count":47},{"key_as_string":"1995","key":788918400000,"doc_count":46},{"key_as_string":"2004","key":1072915200000,"doc_count":39},{"key_as_string":"2002","key":1009843200000,"doc_count":37},{"key_as_string":"2001","key":978307200000,"doc_count":31},{"key_as_string":"2003","key":1041379200000,"doc_count":22},{"key_as_string":"2009","key":1230768000000,"doc_count":10},{"key_as_string":"2008","key":1199145600000,"doc_count":9},{"key_as_string":"2005","key":1104537600000,"doc_count":7},{"key_as_string":"2012","key":1325376000000,"doc_count":7},{"key_as_string":"1992","key":694224000000,"doc_count":6},{"key_as_string":"2006","key":1136073600000,"doc_count":6},{"key_as_string":"2010","key":1262304000000,"doc_count":3},{"key_as_string":"2011","key":1293840000000,"doc_count":3},{"key_as_string":"0202","key":-55792713600000,"doc_count":1},{"key_as_string":"2007","key":1167609600000,"doc_count":1}]},"author":{"doc_count_error_upper_bound":54,"sum_other_doc_count":264102,"buckets":[{"key":"Cucinotta, Francis A.","doc_count":142},{"key":"Wilson, J. W.","doc_count":126},{"key":"Cucinotta, F. A.","doc_count":115},{"key":"Stahl, H. Philip","doc_count":104},{"key":"Banks, Bruce A.","doc_count":100},{"key":"Johnson, Les","doc_count":99},{"key":"Wielicki, Bruce A.","doc_count":97},{"key":"Wilson, John W.","doc_count":79},{"key":"Smith, Scott M.","doc_count":77},{"key":"Minow, Joseph I.","doc_count":76},{"key":"Bloomberg, J. J.","doc_count":74},{"key":"Hammoud, Ahmad","doc_count":70},{"key":"Steinetz, Bruce M.","doc_count":63},{"key":"Landis, Geoffrey A.","doc_count":58},{"key":"Juang, Jer-Nan","doc_count":55},{"key":"Katnik, Gregory N.","doc_count":55},{"key":"Whitaker, Ann F.","doc_count":53},{"key":"Renzetti, N. A.","doc_count":51},{"key":"Wheeler, Raymond M.","doc_count":50},{"key":"Keller, L. P.","doc_count":49}]},"reportNumber":{"doc_count_error_upper_bound":8,"sum_other_doc_count":72534,"buckets":[{"key":"DRL-184-58","doc_count":16},{"key":"LPI-Contrib-1197","doc_count":9},{"key":"2018-561-NEPP","doc_count":7},{"key":"QR-2","doc_count":7},{"key":"QR-3","doc_count":6},{"key":"QR-4","doc_count":6},{"key":"DR-4","doc_count":4},{"key":"JPL-TDA-PR-42-103","doc_count":4},{"key":"LC-92-4468","doc_count":4},{"key":"QR-1","doc_count":4},{"key":"MDC-H1343A","doc_count":3},{"key":"ME-4182","doc_count":3},{"key":"PR-1","doc_count":3},{"key":"QPR-2","doc_count":3},{"key":"REPT-3","doc_count":3},{"key":"SAR-1","doc_count":3},{"key":"12180-60","doc_count":2},{"key":"2015-545-NESC","doc_count":2},{"key":"AAS PAPER 66-150","doc_count":2},{"key":"AAS PAPER 67-37","doc_count":2}]},"center":{"doc_count_error_upper_bound":0,"sum_other_doc_count":0,"buckets":[{"key":"CDMS","doc_count":45510},{"key":"GSFC","doc_count":8527},{"key":"JSC","doc_count":7692},{"key":"MSFC","doc_count":7108},{"key":"JPL","doc_count":4893},{"key":"GRC","doc_count":3871},{"key":"ARC","doc_count":3551},{"key":"HQ","doc_count":3205},{"key":"LaRC","doc_count":2993},{"key":"KSC","doc_count":2639},{"key":"SSC","doc_count":346},{"key":"AFRC","doc_count":276},{"key":"2230","doc_count":48},{"key":"WFF","doc_count":19},{"key":"WSTF","doc_count":5}]},"index":{"doc_count_error_upper_bound":0,"sum_other_doc_count":0,"buckets":[{"key":"submissions-2023-05-12-05-39","doc_count":90683}]},"fundingNumber":{"doc_co

unt_error_upper_bound":61,"sum_other_doc_count":47665,"buckets":[{"key":"NAS7-100","doc_count":950},
{"key":"NAS5-26555","doc_count":320},{"key":"NNA14AB82C","doc_count":268},{"key":"NAS9-13247","doc_
count":207},{"key":"NNJ13HA01C","doc_count":205},{"key":"NNJ15HK11B","doc_count":200},{"key":"NNC1
2BA01B","doc_count":181},{"key":"NAS7-918","doc_count":179},{"key":"NNK11EA08C","doc_count":164},{"k
ey":"NNM12AA41C","doc_count":154},{"key":"NNX13AJ45A","doc_count":152},{"key":"NASW-4435","doc_co
unt":136},{"key":"NNG11HP16A","doc_count":133},{"key":"NNH15CO48B","doc_count":127},{"key":"80MSFC
18C0011","doc_count":114},{"key":"NNG06EO90A","doc_count":106},{"key":"NNG13CR48C","doc_count":10
3},{"key":"NNA16BD14C","doc_count":102},{"key":"NAS3-25266","doc_count":100},{"key":"NNA14AA60C","d
oc_count":96}]},"published":{"buckets":[{"key_as_string":"1992","key":694224000000,"doc_count":3154},{"k
ey_as_string":"1991","key":662688000000,"doc_count":3080},{"key_as_string":"1990","key":631152000000,
"doc_count":2790},{"key_as_string":"1993","key":725846400000,"doc_count":2767},{"key_as_string":"1989"
,"key":599616000000,"doc_count":2378},{"key_as_string":"1994","key":757382400000,"doc_count":2228},{"
key_as_string":"1988","key":567993600000,"doc_count":2134},{"key_as_string":"2004","key":10729152000
00,"doc_count":2101},{"key_as_string":"1987","key":536457600000,"doc_count":2008},{"key_as_string":"20
11","key":1293840000000,"doc_count":1962},{"key_as_string":"1999","key":915148800000,"doc_count":19
36},{"key_as_string":"1985","key":473385600000,"doc_count":1934},{"key_as_string":"2010","key":1262304
000000,"doc_count":1880},{"key_as_string":"2002","key":1009843200000,"doc_count":1836},{"key_as_strin
g":"1986","key":504921600000,"doc_count":1825},{"key_as_string":"2001","key":978307200000,"doc_count
":1817},{"key_as_string":"2016","key":1451606400000,"doc_count":1815},{"key_as_string":"2003","key":104
1379200000,"doc_count":1813},{"key_as_string":"2012","key":1325376000000,"doc_count":1810},{"key_as
_string":"2000","key":946684800000,"doc_count":1804},{"key_as_string":"2005","key":1104537600000,"doc
_count":1724},{"key_as_string":"1995","key":788918400000,"doc_count":1678},{"key_as_string":"2017","key
":1483228800000,"doc_count":1652},{"key_as_string":"2014","key":1388534400000,"doc_count":1605},{"k
ey_as_string":"2015","key":1420070400000,"doc_count":1584},{"key_as_string":"2018","key":15147648000
00,"doc_count":1536},{"key_as_string":"2006","key":1136073600000,"doc_count":1533},{"key_as_string":"2
007","key":1167609600000,"doc_count":1510},{"key_as_string":"2008","key":1199145600000,"doc_count":
1461},{"key_as_string":"1998","key":883612800000,"doc_count":1452},{"key_as_string":"1984","key":44176
3200000,"doc_count":1386},{"key_as_string":"2013","key":1356998400000,"doc_count":1350},{"key_as_stri
ng":"1972","key":63072000000,"doc_count":1289},{"key_as_string":"2009","key":1230768000000,"doc_cou
nt":1284},{"key_as_string":"2019","key":1546300800000,"doc_count":1229},{"key_as_string":"1983","key":4
10227200000,"doc_count":1207},{"key_as_string":"1996","key":820454400000,"doc_count":1192},{"key_as
_string":"1982","key":378691200000,"doc_count":1184},{"key_as_string":"1997","key":852076800000,"doc_
count":1176},{"key_as_string":"1975","key":157766400000,"doc_count":1144},{"key_as_string":"1971","key"
:31536000000,"doc_count":1120},{"key_as_string":"1974","key":126230400000,"doc_count":1090},{"key_a
s_string":"1981","key":347155200000,"doc_count":1045},{"key_as_string":"1980","key":315532800000,"doc
_count":1028},{"key_as_string":"1973","key":94694400000,"doc_count":1017},{"key_as_string":"1977","key":
220924800000,"doc_count":959},{"key_as_string":"1978","key":252460800000,"doc_count":954},{"key_as_
string":"1979","key":283996800000,"doc_count":952},{"key_as_string":"1976","key":189302400000,"doc_co
unt":941},{"key_as_string":"1970","key":0,"doc_count":880},{"key_as_string":"1966","key":-126230400000,"d
oc_count":788},{"key_as_string":"1965","key":-157766400000,"doc_count":737},{"key_as_string":"1967","ke
y":-94694400000,"doc_count":735},{"key_as_string":"2020","key":1577836800000,"doc_count":686},{"key_
as_string":"1968","key":-63158400000,"doc_count":684},{"key_as_string":"1969","key":-31536000000,"doc_
count":628},{"key_as_string":"2021","key":1609459200000,"doc_count":525},{"key_as_string":"1964","key":-
189388800000,"doc_count":437},{"key_as_string":"1963","key":-220924800000,"doc_count":429},{"key_as_
string":"2022","key":1640995200000,"doc_count":420},{"key_as_string":"1962","key":-252460800000,"doc_
count":306},{"key_as_string":"1961","key":-283996800000,"doc_count":95},{"key_as_string":"2023","key":16
72531200000,"doc_count":94},{"key_as_string":"1960","key":-315619200000,"doc_count":41},{"key_as_stri
ng":"1959","key":-347155200000,"doc_count":26},{"key_as_string":"1958","key":-378691200000,"doc_count
":7},{"key_as_string":"1955","key":-473385600000,"doc_count":5},{"key_as_string":"1937","key":-104137920
0000,"doc_count":3},{"key_as_string":"1956","key":-441849600000,"doc_count":3},{"key_as_string":"1922","
key":-1514764800000,"doc_count":2},{"key_as_string":"1923","key":-1483228800000,"doc_count":2},{"key_
as_string":"1935","key":-1104537600000,"doc_count":2},{"key_as_string":"1941","key":-915148800000,"doc
_count":2},{"key_as_string":"1945","key":-788918400000,"doc_count":2},{"key_as_string":"1948","key":-6943
10400000,"doc_count":2},{"key_as_string":"1950","key":-631152000000,"doc_count":2},{"key_as_string":"1
952","key":-568080000000,"doc_count":2},{"key_as_string":"0202","key":-55792713600000,"doc_count":1},{
"key_as_string":"1921","key":-1546300800000,"doc_count":1},{"key_as_string":"1926","key":-138853440000

0,"doc_count":1},{"key_as_string":"1929","key":-1293840000000,"doc_count":1},{"key_as_string":"1930","key":-1262304000000,"doc_count":1},{"key_as_string":"1932","key":-1199232000000,"doc_count":1},{"key_as_string":"1933","key":-1167609600000,"doc_count":1},{"key_as_string":"1934","key":-1136073600000,"doc_count":1},{"key_as_string":"1939","key":-978307200000,"doc_count":1},{"key_as_string":"1940","key":-946771200000,"doc_count":1},{"key_as_string":"1942","key":-883612800000,"doc_count":1},{"key_as_string":"1943","key":-852076800000,"doc_count":1},{"key_as_string":"1944","key":-820540800000,"doc_count":1},{"key_as_string":"1946","key":-757382400000,"doc_count":1},{"key_as_string":"1947","key":-725846400000,"doc_count":1},{"key_as_string":"1951","key":-599616000000,"doc_count":1},{"key_as_string":"1953","key":-536457600000,"doc_count":1},{"key_as_string":"1954","key":-504921600000,"doc_count":1},{"key_as_string":"1957","key":-410227200000,"doc_count":1}]},"distribution":{"doc_count_error_upper_bound":0,"sum_other_doc_count":0,"buckets":[{"key":"PUBLIC","doc_count":90683}]},"stiType":{"doc_count_error_upper_bound":0,"sum_other_doc_count":26,"buckets":[{"key":"CONFERENCE_PAPER","doc_count":37031},{"key":"OTHER","doc_count":11572},{"key":"REPRINT","doc_count":10725},{"key":"CONTRACTOR_REPORT","doc_count":9435},{"key":"PREPRINT","doc_count":5329},{"key":"TECHNICAL_MEMORANDUM","doc_count":5031},{"key":"PRESENTATION","doc_count":4388},{"key":"CONFERENCE_PROCEEDINGS","doc_count":2738},{"key":"VIDEO","doc_count":1195},{"key":"CONTRACTOR_OR_GRANTEE_REPORT","doc_count":677},{"key":"TECHNICAL_PUBLICATION","doc_count":626},{"key":"ACCEPTED_MANUSCRIPT","doc_count":482},{"key":"POSTER","doc_count":386},{"key":"SPECIAL_PUBLICATION","doc_count":277},{"key":"BOOK","doc_count":238},{"key":"THESIS_DISSERTATION","doc_count":227},{"key":"BOOK_CHAPTER","doc_count":138},{"key":"WHITE_PAPER","doc_count":100},{"key":"TECHNICAL_TRANSLATION","doc_count":35},{"key":"CONTRIBUTION_TO_LARGER_WORK","doc_count":27}]},"subjectCategory":{"doc_count_error_upper_bound":754,"sum_other_doc_count":44324,"buckets":[{"key":"Spacecraft Design, Testing And Performance","doc_count":5685},{"key":"Man/System Technology And Life Support","doc_count":4462},{"key":"Spacecraft Propulsion And Power","doc_count":4362},{"key":"Aerospace Medicine","doc_count":3040},{"key":"Astronautics (General)","doc_count":2656},{"key":"Life Sciences (General)","doc_count":2519},{"key":"Astronomy","doc_count":2485},{"key":"Lunar And Planetary Science And Exploration","doc_count":2457},{"key":"Earth Resources And Remote Sensing","doc_count":2218},{"key":"Astrophysics","doc_count":2203},{"key":"Instrumentation And Photography","doc_count":2124},{"key":"Meteorology And Climatology","doc_count":2077},{"key":"Communications And Radar","doc_count":1875},{"key":"Computer Programming And Software","doc_count":1873},{"key":"Geophysics","doc_count":1752},{"key":"Space Transportation And Safety","doc_count":1678},{"key":"Space Communications, Spacecraft Communications, Command And Tracking","doc_count":1666},{"key":"Space Transportation","doc_count":1588},{"key":"Electronics And Electrical Engineering","doc_count":1537},{"key":"Space Vehicles","doc_count":1499}]},"disseminated":{"doc_count_error_upper_bound":0,"sum_other_doc_count":0,"buckets":[{"key":"DOCUMENT_AND_METADATA","doc_count":53665},{"key":"METADATA_ONLY","doc_count":37018}]},"stiTypeDetails":{"doc_count_error_upper_bound":9,"sum_other_doc_count":1447,"buckets":[{"key":"Conference Paper","doc_count":37031},{"key":"Reprint (Version printed in journal)","doc_count":10725},{"key":"Contractor Report (CR)","doc_count":9435},{"key":"Other","doc_count":7198},{"key":"Preprint (Draft being sent to journal)","doc_count":5329},{"key":"Technical Memorandum (TM)","doc_count":5031},{"key":"Presentation","doc_count":4388},{"key":"Conference Proceedings","doc_count":2738},{"key":"Other - NASA Tech Brief","doc_count":2126},{"key":"Video","doc_count":1195},{"key":"Contractor or Grantee Report","doc_count":677},{"key":"Technical Publication (TP)","doc_count":626},{"key":"Accepted Manuscript (Version with final changes)","doc_count":482},{"key":"Other - Patent","doc_count":463},{"key":"Other - NASA Technical Note (TN)","doc_count":398},{"key":"Poster","doc_count":386},{"key":"Special Publication (SP)","doc_count":277},{"key":"Other - Other","doc_count":257},{"key":"Book","doc_count":238},{"key":"Other - Collected Works","doc_count":236}]},"organization":{"doc_count_error_upper_bound":395,"sum_other_doc_count":84094,"buckets":[{"key":"NASA Goddard Space Flight Center","doc_count":9132},{"key":"NASA Marshall Space Flight Center","doc_count":8242},{"key":"NASA Johnson Space Center","doc_count":6949},{"key":"Jet Propulsion Lab., California Inst. of

Tech.","doc_count":5023},{"key":"NASA Langley Research Center","doc_count":4868},{"key":"NASA Ames Research Center","doc_count":4176},{"key":"NASA Glenn Research Center","doc_count":3399},{"key":"NASA Lewis Research Center","doc_count":3126},{"key":"NASA Headquarters","doc_count":2131},{"key":"NASA Kennedy Space Center","doc_count":1560},{"key":"NASA Lyndon B. Johnson Space Center","doc_count":1461},{"key":"Universities Space Research Association","doc_count":1264},{"key":"California Univ.","doc_count":1229},{"key":"Alabama Univ.","doc_count":1183},{"key":"Goddard Space Flight Center","doc_count":1017},{"key":"California Inst. of Tech.","doc_count":780},{"key":"Johnson Space Center","doc_count":778},{"key":"Rockwell International Corp.","doc_count":756},{"key":"Langley Research Center","doc_count":719},{"key":"Maryland Univ.","doc_count":668}]},"modified":{"buckets":[{"key_as_string":"2022","key":1640995200000,"doc_count":71383},{"key_as_string":"2011","key":1293840000000,"doc_count":6162},{"key_as_string":"2013","key":1356998400000,"doc_count":5769},{"key_as_string":"2023","key":1672531200000,"doc_count":2972},{"key_as_string":"2021","key":1609459200000,"doc_count":584},{"key_as_string":"2012","key":1325376000000,"doc_count":256},{"key_as_string":"2020","key":1577836800000,"doc_count":233},{"key_as_string":"2018","key":1514764800000,"doc_count":191},{"key_as_string":"2014","key":1388534400000,"doc_count":149},{"key_as_string":"2016","key":1451606400000,"doc_count":121},{"key_as_string":"2019","key":1546300800000,"doc_count":118},{"key_as_string":"2017","key":1483228800000,"doc_count":82},{"key_as_string":"2015","key":1420070400000,"doc_count":71}]},"keyword":{"doc_count_error_upper_bound":69,"sum_other_doc_count":45278,"buckets":[{"key":"Non-NASA Center","doc_count":612},{"key":"Flight Experiment","doc_count":416},{"key":"manned","doc_count":381},{"key":"short duration","doc_count":350},{"key":"MANNED SPACE FLIGHT","doc_count":187},{"key":"Review","doc_count":184},{"key":"NASA Center JSC","doc_count":173},{"key":"Review, Tutorial","doc_count":164},{"key":"NASA Discipline Radiation Health","doc_count":162},{"key":"SPACE SCIENCE","doc_count":162},{"key":"SPACE ENVIRONMENT","doc_count":161},{"key":"NASA PROGRAM","doc_count":147},{"key":"long duration","doc_count":136},{"key":"SPACE PROGRAM","doc_count":132},{"key":"SPACE VEHICLE","doc_count":126},{"key":"SPACE FLIGHT","doc_count":124},{"key":"STS Shuttle Project","doc_count":123},{"key":"Space","doc_count":116},{"key":"Mars","doc_count":114},{"key":"NASA Discipline Life Support Systems","doc_count":109}]}}}

GET Method I Provided:

- request: http://localhost:8000/api/nasa-sti/?title=space&rows=2
- response: {"results": [{"id": 19860032563, "title": "Space stations and space platforms - Concepts, design, infrastructure, and uses", "url": "/", "authors": [{"name": "Bekey, I."}, {"name": "Herman, D."}], "abstract": "Topics discussed include space infrastructures and early Space Station and platform planning. Consideration is given to the supportive role of the Space Station and platform in future astronomy, earth observation, planetary, and communication space missions. Papers are presented on the history of the Space Station and space platform concepts, potential designs of space stations and space platforms, and long-range plans for space research.", "date": 2013, "position": 0, "source": "Nasa STI"}, {"id": 19930057979, "title": "Engineering, Construction, and Operations in Space III", "url": "/", "authors": [{"name": "Willy Z. Sadeh"}, {"name": "Stein Sture"}, {"name": "Russell J. Miller"}], "abstract": "The present volume on engineering, construction, and operations in space discusses surface structures on the moon and Mars, surface equipment, construction, and transportation on the moon and Mars, in situ materials use and processing, and space energy. Attention is given to such orbital structures as LEO and the space station, space mining and excavation, space materials, space automation and robotics, and space life support systems. Topics addressed include lunar-based astronomy, space systems integration, terrestrial support for space functions, and space education. Also discussed are space plans, policy, and history, space science and engineering, geoengineering and space exploration, and the construction and development of a human habitat on Mars.", "date": 2013, "position": 1, "source": "Nasa STI"}]}

POST Method I Provided:

- request to local= http://localhost:8000/api/follow-user/
- request body :
  followed_username = 0009-0005-5925-1831 (there is a user with this username)
- header:
  username:admin

password:123 (there is a user with these credentials)
- response : (status code = 200) (json)
  { "status": "User followed.” }

Rafet Oğuz Pançuk

Important Issues:
- Third Party API research #105
- Implemented Pubchem GET method #210
- Implemented Accept Follow Request POST method #208
- Implemented Reject Follow Request POST method #209
- Documented Pubchem GET method #232
- Documented Accept Follow Request POST method #233
- Documented Reject Follow Request POST method #234

The name, route, description of utilized third party URIs

I utilized a third party API called **Pubchem API** which we use for obtaining data on chemical structures, properties, identifiers, bioactivity, and more. The PubChem API allows developers to access and retrieve data from the database programmatically.
Their website: **https://pubchem.ncbi.nlm.nih.gov/**
Route to API: **https://pubchem.ncbi.nlm.nih.gov/rest/pug/compound/cid/2244/json**

The name, route, description of the created API functions
- **GET method : Pubchem:**
  This method allows users to send search queries to Pubchem. This API method utilizes PubchemAPI. Function takes 1 parameter called ‘component_id”. Component_id is the id of the chemical component in the system.
  If the search is successful a response with status code 200 is returned. If the component_id is missing 400 is returned. If component_id does not exist in Pubchem database, 404 is returned.
  route: /api/pubchem-get/

- **POST method : Accept follow request:**
  This endpoint takes sender_id and receiver_id as parameters and adds user with id=sender_id to followers list of the user with id=receiver_id.
  This endpoint returns JsonResponse with status code 401 if user credentials are empty. This endpoint returns JsonResponse with status code 407 if user credentials are incorrect
  This endpoint returns JsonResponse with status code 400 and with status 'User/users not found' if user with sender_id or user with receiver_id does not exist in database

This endpoint returns JsonResponse with status code 400 and with status 'There is no such follow request' if a request with sender_id and receiver_id does not exist in database.
This endpoint returns JsonResponse with status code 400 and with status 'Follow request is already answered' if a request with sender_id and receiver_id is already answered.
If all of these conditions are false, this endpoint returns status code 200 and makes necessary arrangements in the database.

route: /api/accept-follow-request/

- **POST method : Reject follow request:**
  This endpoint takes sender_id and receiver_id as parameters and adds user with id=sender_id to followers list of the user with id=receiver_id.
  This endpoint returns JsonResponse with status code 401 if user credentials are empty. This endpoint returns JsonResponse with status code 407 if user credentials are incorrect
  This endpoint returns JsonResponse with status code 400 and with status 'User/users not found' if user with sender_id or user with receiver_id does not exist in database
  This endpoint returns JsonResponse with status code 400 and with status 'There is no such follow request' if a request with sender_id and receiver_id does not exist in database.
  This endpoint returns JsonResponse with status code 400 and with status 'Follow request is already answered' if a request with sender_id and receiver_id is already answered.
  If all of these conditions are false, this endpoint returns status code 200 and makes necessary arrangements in the database.

  route: /api/reject-follow-request/

Detailed information about the route, parameters and responses of the api functions can be found in the swagger page of our app which can be accessed in /api/swagger-ui/

Description of unit tests

Unit tests of Pubmet Get Method

test_empty_compound_id: This test checks the behavior when an empty compound ID is provided. It makes a GET request to the "/api/pubchem-api/" endpoint with an empty compound_id parameter. The test asserts that the response status code should be 400 (Bad Request) and that the JSON response should contain a "status" field with the value "Compound ID can't be empty". This test aims to verify that the API correctly handles empty compound IDs.

test_invalid_compound_id: This test checks the behavior when an invalid compound ID is provided. It makes a GET request to the "/api/pubchem-api/" endpoint with a 'dummy' compound ID. The test expects the API to respond with a 404 status code (Not Found), indicating that there are no compounds with the requested compound ID. It also verifies that the JSON response contains a "status" field with the value "There isn't any compounds with the requested compound ID". This test ensures that the API correctly handles invalid compound IDs.
test_valid_compound_id: This test checks the behavior when a valid compound ID is provided. It makes a GET request to the "/api/pubchem-api/" endpoint with a compound ID of '1'. The test expects a successful response with a 200 status code, indicating a successful request. It also verifies that the JSON response contains a "status" field with the value "Compound found". This test ensures that the API can correctly handle and retrieve data for a valid compound ID.

Unit tests of Accept Follow Request POST Method

test_unauthorized_receiver: This test checks the behavior when an unauthorized receiver tries to accept a follow request. It performs HTTP POST requests to the "/api/accept-follow-request/" endpoint with different scenarios. It verifies that if no headers are provided, or empty/invalid credentials are provided, the response should have a 407 status code (Proxy Authentication Required) or a 401 status code (Unauthorized). The JSON response is expected to contain specific error messages indicating the incorrect credentials.
test_nonexisting_follow_request: This test checks the behavior when a non-existing follow request is provided. It makes an HTTP POST request to the "/api/accept-follow-request/" endpoint with headers containing valid receiver credentials but with wrong sender or receiver IDs. The test expects a 400 status code (Bad Request) and a JSON response indicating that the user/users are not found.
test_answered_follow_request: This test checks the behavior when an already answered follow request is provided. It makes an HTTP POST request to the "/api/accept-follow-request/" endpoint with headers containing valid receiver credentials and a follow request that has already been answered. The test expects a 400 status code (Bad Request) and a JSON response indicating that the follow request is already answered.
test_unanswered_follow_request: This test checks the behavior when an unanswered follow request is provided. It makes an HTTP POST request to the "/api/accept-follow-request/" endpoint with headers containing valid receiver credentials and an unanswered follow request. The test expects a 200 status code (OK) and a JSON response indicating that the follow request has been accepted.

Unit tests of Reject Follow Request POST Method

test_unauthorized_receiver: This test checks the behavior when an unauthorized receiver tries to reject a follow request. It performs HTTP POST requests to the

"/api/reject-follow-request/" endpoint with different scenarios. It verifies that if no headers are provided or invalid credentials are provided, the response should have a 407 status code (Proxy Authentication Required) or a 401 status code (Unauthorized). The JSON response is expected to contain specific error messages indicating the incorrect credentials.
test_nonexisting_follow_request: This test checks the behavior when a non-existing follow request is provided. It makes an HTTP POST request to the "/api/reject-follow-request/" endpoint with headers containing valid receiver credentials but with wrong sender or receiver IDs. The test expects a 400 status code (Bad Request) and a JSON response indicating that the user/users are not found.
test_answered_follow_request: This test checks the behavior when an already answered follow request is provided. It makes an HTTP POST request to the "/api/reject-follow-request/" endpoint with headers containing valid receiver credentials and a follow request that has already been answered. The test expects a 400 status code (Bad Request) and a JSON response indicating that the follow request is already answered.
test_unanswered_follow_request: This test checks the behavior when an unanswered follow request is provided. It makes an HTTP POST request to the "/api/reject-follow-request/" endpoint with headers containing valid receiver credentials and an unanswered follow request. The test expects a 200 status code (OK) and a JSON response indicating that the follow request has been rejected.

Sample calls:

Pubmed GET

- **request:** https://pubchem.ncbi.nlm.nih.gov/rest/pug/compound/cid/2244/json

- **response :**

```
{
  "PC_Compounds": [
    {
      "id": {
        "id": 2244,
        "idtype": "CID"
      },
      "atoms": {
        "aid": [
          {
            "number": 1,
            "element": "C",
            "x": 7.6871,
            "y": -1.3482,
            "z": -0.0806
          },
          {
            "number": 2,
            "element": "C",
            "x": 6.2796,
            "y": -1.3799,
            "z": 0.381
          },
          ...
        ],
        ...
      },
      "bonds": {
        "aid1": [
          1,
          2,
          ...
        ],
        ...
      },
      ...
    }
  ]
}
```

Accept Follow Request POST

- request to local= http://localhost:8000/api/accept-follow-request/
- request body :
  sender_id = 0009-0005-5924-2031
  receiver_id = 0009-0005-5924-2032
- response : (status code = 200) (json)
  { "status": "Follow request accepted" }

Reject Follow Request POST

- request to local= http://localhost:8000/api/reject-follow-request/
- request body :
  sender_id =
  receiver_id = dummy'
- response : (status code = 400) (json)
  { "status": "User/users not found." }