

I. IDENTIFICATION DATA

Title:	User interface development for advanced searching in the MBDB
Author's name:	Kryštof Krejčí
Type of assignment:	Bachelor Project
Faculty:	Faculty of Nuclear Sciences and Physical Engineering (FNSPE)
Department:	Department of Mathematics
Reviewer:	RNDr. Martin Mokrejš, Ph.D.
Reviewer's affiliation:	Institute of Biotechnology of the Czech Academy of Sciences

II. ASSESSMENT OF CRITERIA

Work assignment	easy
<i>Assess how demanding the work topic is.</i>	
The student created a simple python scripts used to convert YAML to JSON with a few data checks on top aiming to be provided (in an unspecified way) to Elasticsearch.	

Fulfilling the assignment	fulfilled with reservations
<i>Consider whether the work submitted meets the assignment. If necessary, give your comments on items of the assignment not fully answered, or judge whether the scope of the assignment has been broadened. If student failed to fully treat the assignment, try to assess the importance, impact and/or the reasons for the failings.</i>	
Student should familiarize himself with the MBDB project, its schema, interfaces. However, a schema of the database is not provided in the thesis and at the best, one can trace in the github repository trase MST.yaml with 3249 lines of YAML lines. No overall figure is available, not even mentioning the other three methods to be supported. It is very hard to infer an overall picture of the project and of the part contributed by the student.	

Chosen approach to solution	appropriate with reservations
<i>Assess whether student applied a correct approach or method of solution.</i>	
The is no relevant introduction to existing tools used to render web-based query forms (e.g. web templating technologies) and why they cannot be used for this task. There is no discussion why existing YAML to JSON technologies could not be used and instead, why yet another YAML to JSON was created. It is unclear why no existing syntactic checks could not be employed too. The thesis is quiet which YAML schema standard it supports (although in the end version 1.2 is hidden on page 15 so that probably answers my question).	

Professional standard	below average
<i>Assess the professional standard of the work, application of course knowledge, references, and data from practice.</i>	
The python-based code is not a pythonic package and therefore cannot be installed as usual an imported. It also contains hardcoded paths to its components. The additional research methods cannot be easily added unless the hardcoded paths are taken out. Not even cmdline arguments nor environment variables were employed. The hardcoded relative paths are even visible in the thesis, see e.g. Listing 3.1 on page 25. https://github.com/Molecular-Biophysics-Database/mbdb-search/blob/main/src/MST.yaml https://github.com/Molecular-Biophysics-Database/mbdb-search/blob/da895e341ca65cc97f2fdca171f93a434f817cf9/src/convert_yaml_to_json.py#L79 https://github.com/Molecular-Biophysics-Database/mbdb-search/blob/da895e341ca65cc97f2fdca171f93a434f817cf9/src/luqui_converter.py#L195 https://github.com/Molecular-Biophysics-Database/mbdb-search/blob/da895e341ca65cc97f2fdca171f93a434f817cf9/src/main.jsx#L3	

The README.md is almost empty and definitely not helpful in any way. Provided the python code does not employ any doc strings one is left to study the code itself or rely on a few comment lines.

The git commits have only a typically inscriptive subject line but not explanatory text describing what is the particular change about. There were 96 commit in total. The code spans the following numbers of lines (including empty lines, comments, etc.):

147 src/App.css

567 src/App.jsx

102 src/convert_yaml_to_json.py

69 src/index.css

231 src/luql_convertor.py

10 src/main.jsx

3248 src/MST.yaml

2488 src/mst-1.0.0.json

35 src/output_for_presentation.json [is this some forgotten file in the codebase?]

8123 src/output.json [this is actually a file generated by convert_yaml_to_json.py]

64 src/prettier_names.py

327 src/SearchCriteria.jsx

89 src/simplified_model.yaml

It is great some continuous-integration tests are in place, per <https://github.com/Molecular-Biophysics-Database/mbdb-search/actions> they time to time catch a breaking commit. However, it is not clear what the student has setup from the repository on his own.

Interestingly, the software outputs a JSON string but upon producing it it immediately re-opens it to parse it and mange it (replace underscores with spaces, capitalize first characters of every word and alter names so they are unique). This could have been implemented using a single pass, provided this is a student learning project.

Naming a software App is asking for a trouble due to name collisions.

Level of formality and of the language used

below average

Assess the use of scientific formalism, the typography and language of the work.

There are a few typos in the thesis, but really a few. One broke the first sentence of chapter 3.4 into two, mistakenly. More difficult to spot is probably in the first sentence of paragraph 3.3.2 where author speaks of about 500 searchable items in the MBD database but probably has meant searchable fields of the data model instead.

The Figure legends are just unhelpful. The legends should clearly describe what is one supposed to infer from a Figure. Moreover, there are no references to the Figures from the main text, so can actually miss the figures completely. Sadly, the Figure 3.2 is the main result of this whole student project

On a similar note, the individual paragraphs on pages 18 to 20 are probably aiming to speak of the contents in Figure 3.2 but it is not stated anywhere and the formulations of the sentences would need to be revised in some places. Preferably, most of this text should be melted into the Figure legend.

Personally I would prefer more logical structure of the thesis: Introduction, Methods, Results, Discussion, Supplements.

The Chapter 1 is too general and unhelpful in understanding the work to be done.
The Chapter 2 does not really lay down clearly the existing environment, into which new functionalities should be implemented, nor does it describe what approaches were considered.
The Chapter 3 is probably Methods+Results+Discussion mixed together.

References to somewhat relevant software tools are added after each paragraph although the paragraph itself refers to multiple. The referencing approach should be largely improved.

Arguably the second main output of the thesis is presented on page 25, where is a sketch of App.jsx code without a pointer to the full source code on file. One is left to find

Choice of references, citation correctness

below average

Assess student's effort in finding and using study sources for completing their work. Give characteristics of the references chosen. Assess whether student made use of all the relevant sources. Verify whether all items used are properly distinguished from the results obtained by student and their deliberations, whether there are no violations of citation ethics, and whether the bibliography presented is complete and complies with the citation usage and standards.

References to somewhat relevant software tools are added after each paragraph although the paragraph itself refers to multiple. The referencing approach should be largely improved. Notably, there is no brief introduction to existing software solution, for example there are plenty YAML to JSON converters. Why is there need for yet another one, not even describing which YAML specification it respects. Figure legends are almost non-existent. Also, the only more complex figure was actually taken over from Invenio website, without proper reference and without explanation. There is not even a pointer from the main text to this particular Figure 3.2.

The Figure 2.2 was simply taken over from <https://narodni-repozitar.github.io/developer-docs/docs/technology/invenio/architecture/> although it is not mentioned in a Figure legend. What should one infer from this schema? Where is the code developed during this software project represented in the figure?

Further comments and assessment

Give your opinion on the quality of the main results obtained in the work, e.g. the theoretical results, or the applicability of the engineering or programming solutions obtained, publication outputs, experimental skills, and the like.

I would scratch out the BLI.yaml, SPR.yaml and ITC.yaml from the page 16 because these files are not present (yet?) in the source code tree and it is unclear how they would be called from the python code. See the GitHub links spotted above my review under "Professional standard" on page 1 (bottom) pointing to just MST.yaml and mst-1.0.0.json.

The user is, per-description the thesis, allowed to save search queries for future re-use. Does the developed software facilitate batch queries using those, saved queries? How can one execute them and how to collect the results. Examples of queries and returned out would be very helpful.

What is the “Exp” button in Figure 3.2 on page 18? There is no “Exp” mentioned in the thesis at all.

III. OVERALL ASSESSMENT, QUESTIONS TO BE ASKED DURING THE WORK DEFENCE, SUGGESTED GRADE

Summarize those aspects of the work that were significantly influential for your overall assessment. Suggest questions to be answered by student during the defence of the work before the examination board.

The presented thesis is a difficult read. The thesis should be full of implementation details, approaches considered/tried but eventually not implemented with explanation why some other approach was taken. That would make the text not only more readable but also show what the student has learned and actually done. It is a pity that from a quick glance the Invenio software bundle is not very documented, the docker image is not available for download (see <https://narodni-repozitar.github.io/developer-docs/docs/technology/invenio/architecture/>). It is unclear how the developed code should be implemented into the whole framework, namely in the context of apps running under <https://narodni-repozitar.github.io/developer-docs/docs/technology/invenio/ecosystem#required-ports> . I have heard about incompatibilities of the developed code with Invenio which currently requires React 16 but the App.jsx e.g. at its very end refers to React 18. What are the current plans to tackle the issue?

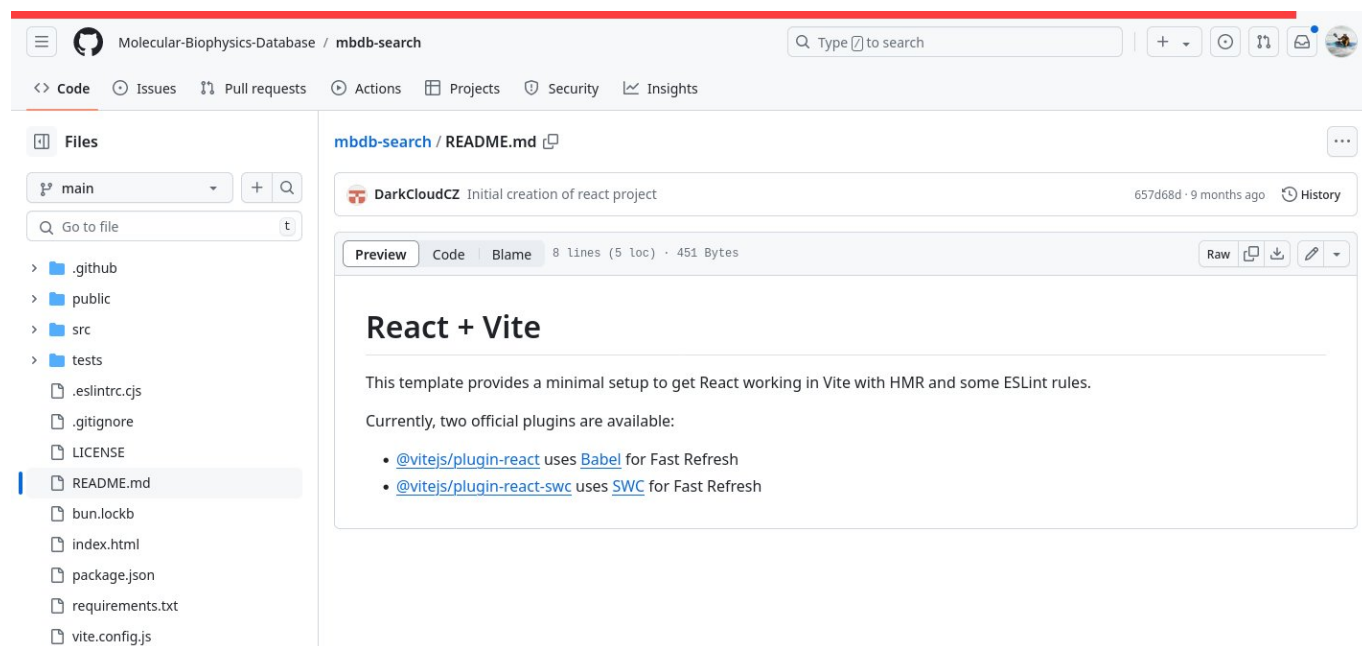


Figure 1: The main landing page of a project. Usually there is written and to install and run the tool.

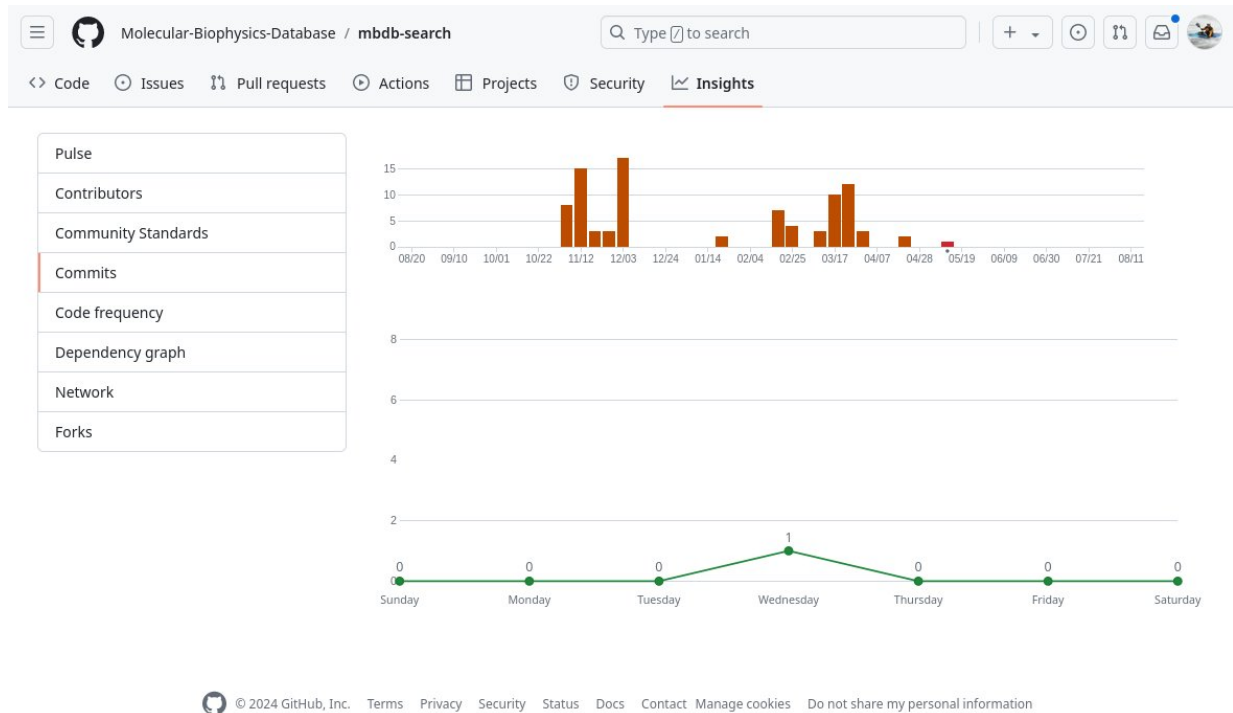


Figure 2: Project activity from <https://github.com/Molecular-Biophysics-Database/mbdb-search/graphs/commit-activity> was mainly in Nov2023 and 03/2024.

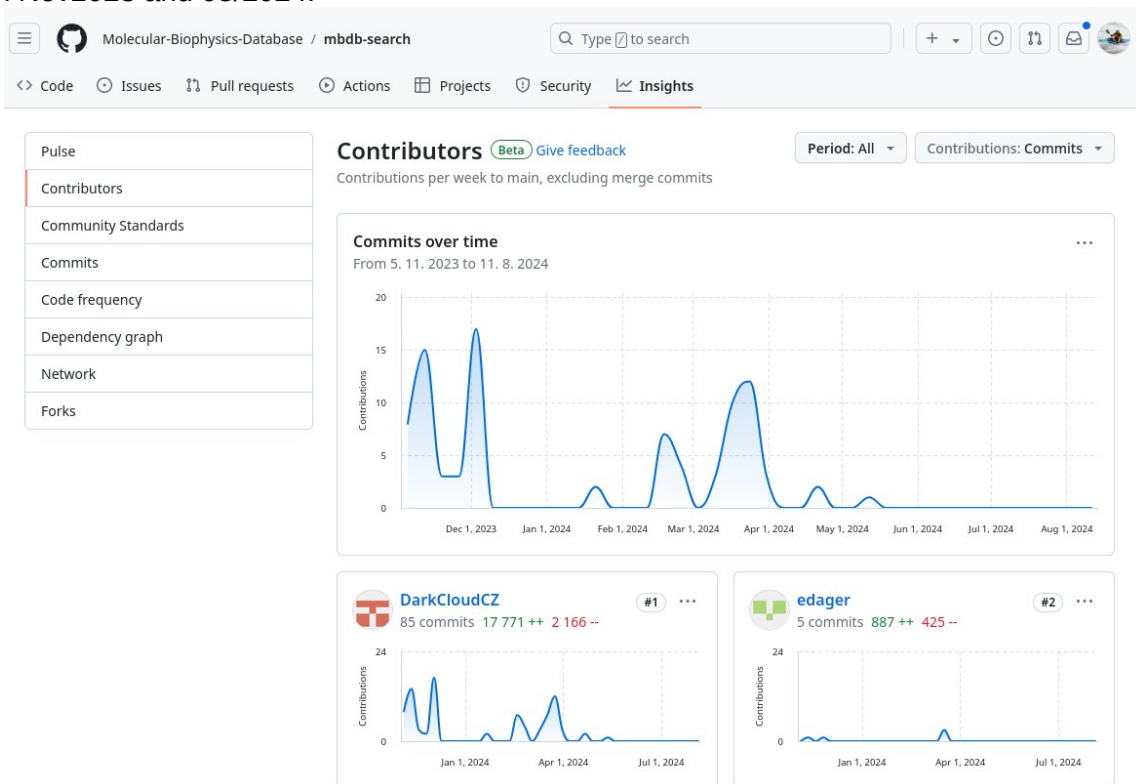
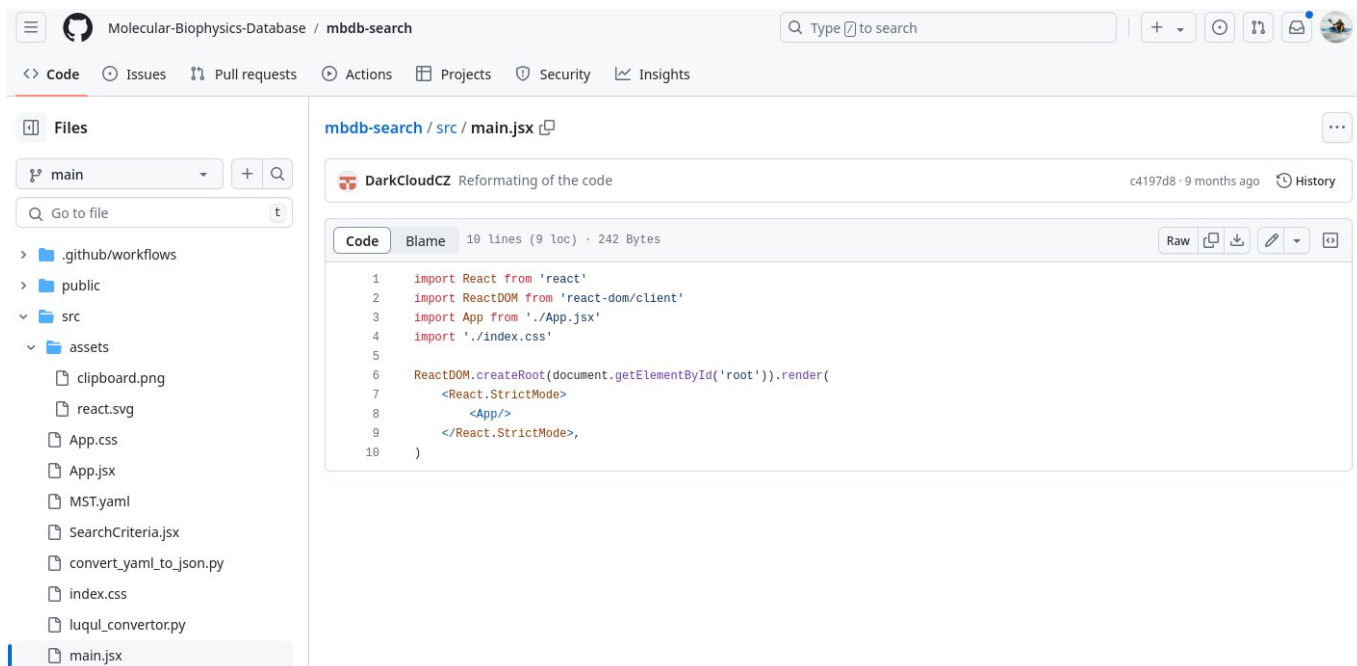


Figure 3: Activity in the project according to <https://github.com/Molecular-Biophysics-Database/mbdb-search/graphs/contributors>.



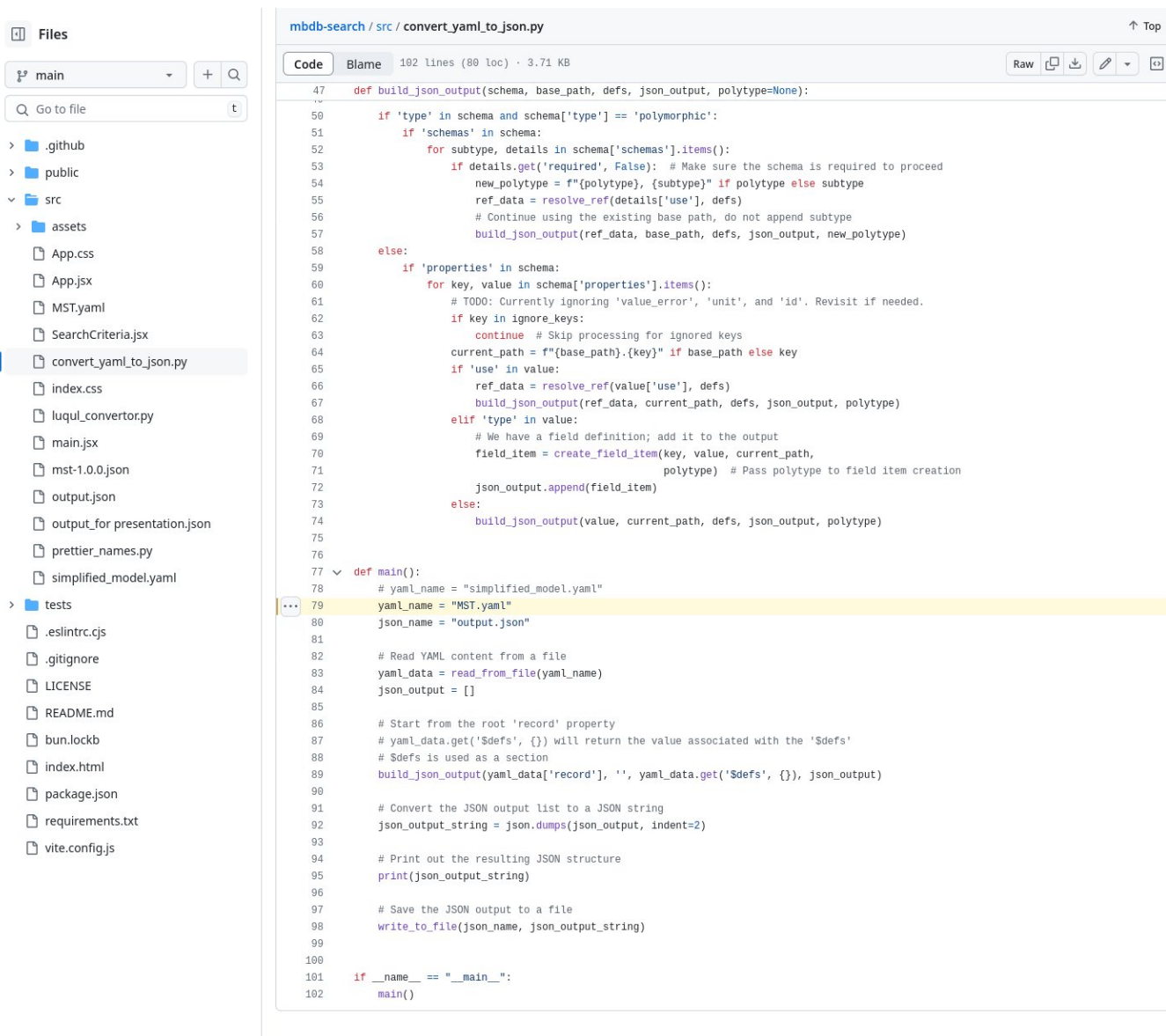
The screenshot shows a GitHub repository named 'Molecular-Biophysics-Database / mbdb-search'. The file 'main.jsx' is open, showing the following code:

```
1 import React from 'react'
2 import ReactDOM from 'react-dom/client'
3 import App from './App.jsx'
4 import './index.css'
5
6 ReactDOM.createRoot(document.getElementById('root')).render(
7   <React.StrictMode>
8     <App/>
9   </React.StrictMode>,
10 )
```

The code is highlighted in red, indicating a linting error. The error message is 'DarkCloudCZ Reformating of the code' with a commit hash 'c4197d8 - 9 months ago' and a 'History' link. The file is 10 lines long, 9 loc, and 242 Bytes. The file explorer on the left shows the directory structure: .github/workflows, public, src, and assets. The assets directory contains clipboard.png, react.svg, App.css, App.jsx, MST.yaml, SearchCriteria.jsx, convert_yaml_to_json.py, index.css, luqul_convertor.py, and main.jsx.

Figure 4: Hardcoded import of `./App.jsx`, please refer to

<https://github.com/Molecular-Biophysics-Database/mbdb-search/blob/da895e341ca65cc97f2fdca171f93a434f817cf9/src/main.jsx#L3> .



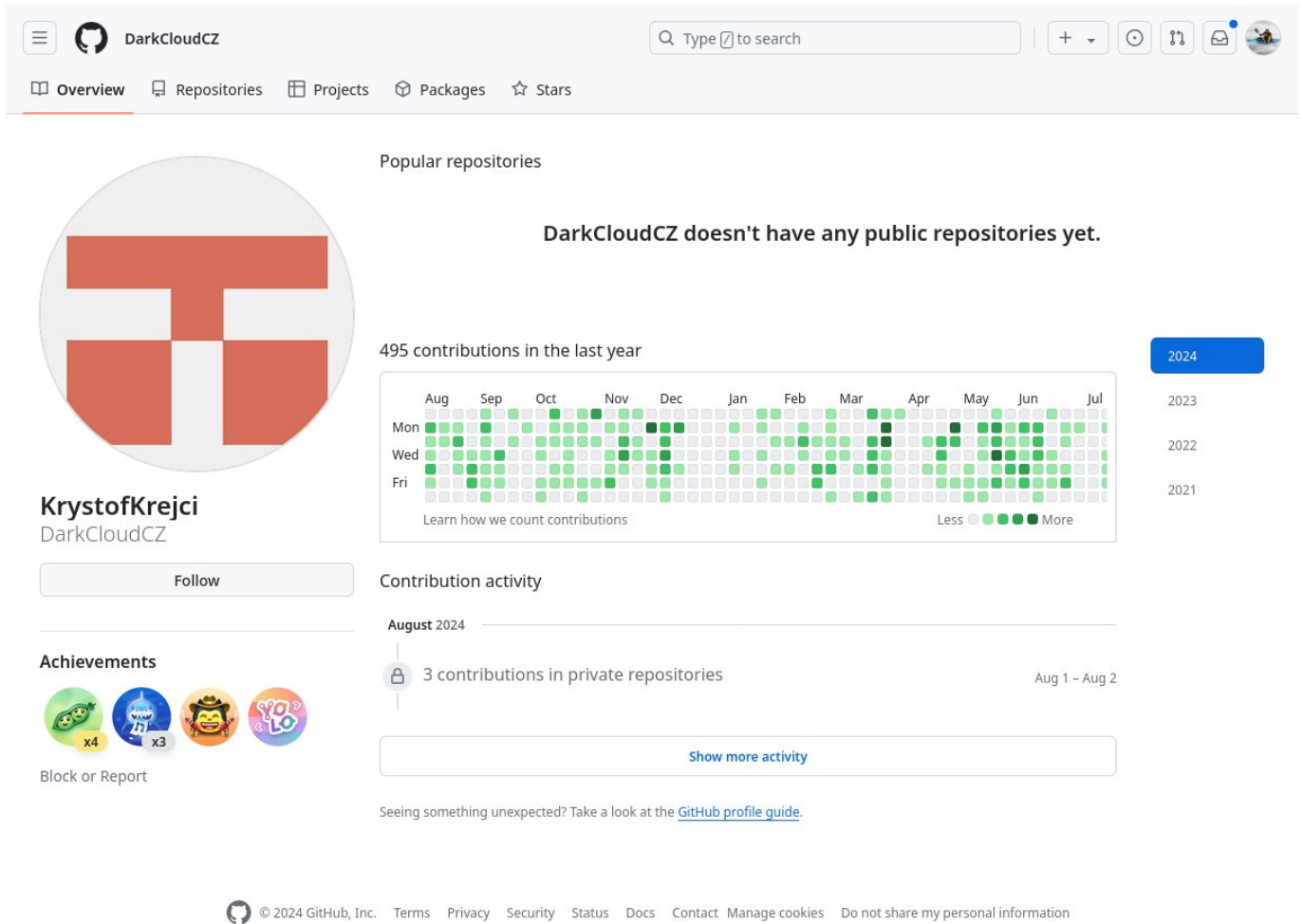
The screenshot shows a GitHub repository interface. On the left is a file explorer showing the directory structure: main, public, src, assets, and tests. The file 'convert_yaml_to_json.py' is selected in the 'src' directory. On the right is a code editor showing the content of 'convert_yaml_to_json.py'. The code is a Python script for converting YAML to JSON. Line 79 is highlighted in yellow, showing the hardcoded assignment: `yaml_name = "MST.yaml"`. The code includes a `def build_json_output` function and a `def main` function that reads the YAML file, processes it, and writes the JSON output to a file.

```

mbdb-search / src / convert_yaml_to_json.py
Code Blame 102 lines (80 loc) · 3.71 KB
47 def build_json_output(schema, base_path, defs, json_output, polytype=None):
48
49     if 'type' in schema and schema['type'] == 'polymorphic':
50         if 'schemas' in schema:
51             for subtype, details in schema['schemas'].items():
52                 if details.get('required', False): # Make sure the schema is required to proceed
53                     new_polytype = f"{polytype}, {subtype}" if polytype else subtype
54                     ref_data = resolve_ref(details['use'], defs)
55                     # Continue using the existing base path, do not append subtype
56                     build_json_output(ref_data, base_path, defs, json_output, new_polytype)
57
58     else:
59         if 'properties' in schema:
60             for key, value in schema['properties'].items():
61                 # TODO: Currently ignoring 'value_error', 'unit', and 'id'. Revisit if needed.
62                 if key in ignore_keys:
63                     continue # Skip processing for ignored keys
64                 current_path = f"{base_path}.{key}" if base_path else key
65                 if 'use' in value:
66                     ref_data = resolve_ref(value['use'], defs)
67                     build_json_output(ref_data, current_path, defs, json_output, polytype)
68             elif 'type' in value:
69                 # We have a field definition; add it to the output
70                 field_item = create_field_item(key, value, current_path,
71                                             polytype) # Pass polytype to field item creation
72                 json_output.append(field_item)
73             else:
74                 build_json_output(value, current_path, defs, json_output, polytype)
75
76
77 def main():
78     # yaml_name = "simplified_model.yaml"
79     yaml_name = "MST.yaml"
80     json_name = "output.json"
81
82     # Read YAML content from a file
83     yaml_data = read_from_file(yaml_name)
84     json_output = []
85
86     # Start from the root 'record' property
87     # yaml_data.get('$defs', {}) will return the value associated with the '$defs'
88     # $defs is used as a section
89     build_json_output(yaml_data['record'], '', yaml_data.get('$defs', {}), json_output)
90
91     # Convert the JSON output list to a JSON string
92     json_output_string = json.dumps(json_output, indent=2)
93
94     # Print out the resulting JSON structure
95     print(json_output_string)
96
97     # Save the JSON output to a file
98     write_to_file(json_name, json_output_string)
99
100
101 if __name__ == "__main__":
102     main()

```

Figure 5: Hardcoded import enabling only a single experimental method. Please refer to https://github.com/Molecular-Biophysics-Database/mbdb-search/blob/main/src/convert_yaml_to_json.py#L79



DarkCloudCZ

Overview Repositories Projects Packages Stars

Popular repositories

DarkCloudCZ doesn't have any public repositories yet.

495 contributions in the last year

2024
2023
2022
2021

KrystofKrejci
DarkCloudCZ

Follow

Achievements

Block or Report

Contribution activity

August 2024

3 contributions in private repositories Aug 1 - Aug 2

Show more activity

Seeing something unexpected? Take a look at the [GitHub profile guide](#).

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

Figure 6: Kryštof is evidently an active coder on github.com .

Despite the tothingng issues I believe the purpose of the student project was to provide basis for learning.

Suggested grade: D - satisfactory.

Date: 20/08/2024

Signature: