CSCI 6991
Data Engineering Capstone

Guided By :
Dr. Feng George Yu

Done By :
Harsha Vardhan Munari
Parveen Shaik
Buvana Yalam

DATA PROCESSING & VISUALIZATION USING MEAN

Structured Data Ingestion, Storage and Retrieval of NSF Awards (1960–2025)

AGENDA

- Introduction
- Background
- Problem Statement
- Methodology
- Experiments
- Discussion
- Conclusion

Introduction

Source : https://www.nsf.gov/awardsearch/download.jsp

Processing and parsing huge unstructured content from source url in to a structured content.

Hence for parsing the folder contents developed a backend system to procees the data.

Storing the nsf data in json to a relational data base mysql.

Joining tables based on award id key in awards table to inner join institutions, principleInvestigators, Performance Institution, Funds, Funding Obligations, Program Elements, Program Reference.

Presenting Data using in tables with accordion cards using Js framwork.

Background

- The U.S. National Science Foundation is an independent federal agency that supports science and engineering in all 50 states and U.S. territories.
- NSF was established in 1950 by Congress to:
- **Promote** the progress of science. **Advance** the national health, prosperity and welfare. **Secure** the national defense.

Hence every year awards will be presented to the persons/Institutions who excelled in science for their contributions started from 1960 – till now.

Where we have huge data sets provided in the url previously this data is available in XML format, XML is a widely-used data format for data exchange across different platforms and systems .

But In January 2025, NSF converted the downloadable format for all awards from XML to JSON stored in year wise zip folders to download.

Source : https://www.nsf.gov/awardsearch/download.jsp

Problem Statement

- **Format Incompatibility**

JSONs hierarchical structure doesn't naturally align with relational database schemas. Where db structure, data type, charset, ..etc parameters should be achieved to store the json in a relational db.

- **Parsing Complexity**

Efficiently converting and validating large JSON files poses technical difficulties.

As the provided source has huge content and in unsupported format to read and process, extract zip contents and read json files init to a sql insert query

- **Performance Concerns**

JSON processing can become a bottleneck in high-volume web applications, which requires high end fast processing system or backend to do and achieve upload files, format conversions, parsing, fetch the db data, process again to json, and visualize in Angular application, do search, pagination, ..etc

Methodology

- The data sets/files available in Nsf downloads are huge, should be processed and migrated in structured manner to a relational database preferably MYSQL
- Bulk JSON data sets(1960–2025) stored in unstructured ZIP folders should be extracted, processed and read every json files, extract contents from those files to store in a relational structured db with proper insertions in rows, ignore any unavailable data, segregate data into multiple tables and join the tables based on awardid as primary/foreign key for efficiency and to acheive ACID principles.
- Need for relational storage with indexing, rows and columns to enable efficient querying based on the structure and json schema defined in nsf source.
- Upload, retrieve, search the contents of db using backend apis with minimal delay as api processing time should be minimized while processing.
- Visualize the data and its contents in a clean UI in tabluar format, with accordions to show related data sets with search by award id, mail, pagination, and limit contents to 50 items per page for faster processing and accessing of files.

Experiments

- **For Data** – From source site download all **ZIP** files in a single folder, upload individually to backend system.
- **For Backend** – Setup a backend which is efficient and reliable in processing, extracting, reading and writing..etc features, one of the efficient programming language is c++, but we can achieve these in shorter time by utilising a framework which was developed on top of c++, which is google's v8 chrome engine called **Nodejs**, where it consits of lakhs of packages as modules to perform and achieve multiple real time workflows/tasks, In backend

(a)Express JS, Router – for setting up the server, and accessing api route paths

(b)NodeSequelize ORM – for database communication, generating db tables/schemas/structures and aggregations, faster updated queries, joins, includes..etc features available compatible with mysql2, postgres, any relational db.

(**c)os, workerthreads, fs, async await, promises** – effective, efficient inbuilt node packages/functions utilised, for uploading zip files using multer, for parsing, modeling, processing utilizing os, workerthreads, filesystem modules, and async await for asynchronous operations and returning response, promises to represent the completion of an asynchronous operation.

- **For Frontend** – utilising a javascript/typescript framework which is **Angular** to represent and visualize the data in structured appealing way with **bootstrap css** styles , making table collapsable with accordion effect to inbuilt sub contents of actual row data into collapsable cards.

Discussion

- **Workflow**

(a) Upload a zip file from client which calls /**upload** api, after some time the zip files are processed and json data sets are extracted, then parse them in to a db friendly values which are now inserted into db rows, like wise will perform for all the files, these data consists in 9 tables where the db is developed as per schema available in nsf, (Award, PI, Inst, PerfInst, PgmEle, PgmRef, AppFund, OblgFy, Por).

(b)Now we have data available in db, once client is loaded, an api called /**awards** will fetch the contents from db. As data is huge, to maintain low latency added the features like limit, pagination and included/innerjoin the sub content tables data (PI, Inst, PerfInst, PgmEle, PgmRef, AppFund, OblgFy, Por) based on foreign key award_id

(c) As data is huge unable to check the known awards in whole db, implemented search feature in an api call /**awards/search** based on search term along with pagination.

(d) The data values are shown in a visually appealing screen to user in tablular form with accordion collapsable cards in angular

- **System performance and evaluation**

In nodejs using os.cpus() to know the hardware configuration, hence utilizing the hardware cores to do tasks better by assigning tasks through nodejs worker threads and batch processing, which mitigates and eradicate the system crash and enhance performance.

- **Testing and reviews –** completed sanity testing and implemented post calls for all apis to validate the request and parsing, tested successfully in **postman,** done load test utilising most of the cores and memory

Conclusion

- **Project Status**

Achieved a fully functional end to end full stack application with database configuration, With upload, fetch and search features.

Commands : **npm start, pm2 start index**

Utilised most of the resources and modern tools at the time of developments and enhancements

- **Future Work**

Will enhance the application with most advanced security features like implementing jwt(jsonwebtoken), validation, authorization, authentication, cors, ..etc.

- **Learning Outcomes**

Enhanced skills in developing Full stack application using JS frameworks like Angular, NodeJS, Mysql.

References

- Data and db structure

NSF - https://www.nsf.gov/awardsearch/download.jsp

DataBase - https://www.w3schools.com/MySQL/default.asp

- Developments

Backend Nodejs

(1) https://nodejs.org/en

(2) https://www.npmjs.com/

(3) https://nodejs.org/en/learn/getting-started/introduction-to-nodejs

(4) https://docs.npmjs.com/cli/v8/using-npm/registry

- Frontend Angular

https://angular.dev/

Appendix

- **Source code - https://github.com/**
- **Video link - https://www.youtube.com/watch?v=Hgm2V_zqGqM**
- **Frontend client Url - http://localhost:4100/**
- **Backend server/Api Urls**
- http://localhost:3000/awards
- http://localhost:3000/awards/search
- http://localhost:3000/upload
- **Database Url – http://localhost:3306**
- **Test cases for doing curl**

```
curl --location 'localhost:3000/awards/search' \
--header 'Content-Type: application/json' \
--header 'x-file-type: js' \
--data '{
    "page": "1",
    "searchTerm":"Hyman H. Field",
    "limit": "50"
}‘
```

```
curl --location 'localhost:3000/awards' \
--header 'Content-Type: application/json' \
--header 'Content-Type: text/plain' \
--header 'x-file-type: js' \
--data '{
    "page": "1",
    "limit": "50"
}'
```

- **Sample queries**

```
SELECT COUNT(*) AS `count` FROM `awards` AS `awards`;
```

```
SELECT `awards`.*, `pis`.`id` AS `pis.id`, `pis`.`pi_role` AS `pis.pi_role`, `pis`.`pi_first_name` AS `pis.pi_first_name`, `pis`.`pi_last_name` AS `pis.pi_last_name`, `pis`.`pi_mid_init` AS `pis.pi_mid_init`, `pis`.`pi_sufx_name` AS `pis.pi_sufx_name`, `pis`.`pi_full_name` AS `pis.pi_full_name`, `pis`.`pi_email_addr` AS `pis.pi_email_addr`, `pis`.`nsf_id` AS `pis.nsf_id`, `pis`.`pi_start_date` AS `pis.pi_start_date`, `pis`.`pi_end_date` AS `pis.pi_end_date`, `pis`.`awardAwdId` AS `pis.awardAwdId`, `inst`.`id` AS `inst.id`, `inst`.`inst_name` AS `inst.inst_name`, `inst`.`inst_street_address` AS `inst.inst_street_address`, `inst`.`inst_street_address_2` AS `inst.inst_street_address_2`, `inst`.`inst_city_name` AS `inst.inst_city_name`, `inst`.`inst_state_code` AS `inst.inst_state_code`, `inst`.`inst_state_name` AS `inst.inst_state_name`, `inst`.`inst_phone_num` AS `inst.inst_phone_num`, `inst`.`inst_zip_code` AS `inst.inst_zip_code`, `inst`.`inst_country_name` AS `inst.inst_country_name`, `inst`.`cong_dist_code` AS `inst.cong_dist_code`, `inst`.`st_cong_dist_code` AS `inst.st_cong_dist_code`, `inst`.`org_lgl_bus_name` AS `inst.org_lgl_bus_name`, `inst`.`org_prnt_uei_num` AS `inst.org_prnt_uei_num`, `inst`.`org_uei_num` AS `inst.org_uei_num`, `inst`.`instAwdId` AS `inst.instAwdId`, `inst`.`awardAwdId` AS `inst.awardAwdId`, `perf_inst`.`id` AS `perf_inst.id`, `perf_inst`.`perf_inst_name` AS `perf_inst.perf_inst_name`, `perf_inst`.`perf_str_addr` AS `perf_inst.perf_str_addr`, `perf_inst`.`perf_city_name` AS `perf_inst.perf_city_name`, `perf_inst`.`perf_st_code` AS `perf_inst.perf_st_code`, `perf_inst`.`perf_st_name` AS `perf_inst.perf_st_name`, `perf_inst`.`perf_zip_code` AS `perf_inst.perf_zip_code`, `perf_inst`.`perf_ctry_code` AS `perf_inst.perf_ctry_code`, `perf_inst`.`perf_cong_dist` AS `perf_inst.perf_cong_dist`, `perf_inst`.`perf_st_cong_dist` AS `perf_inst.perf_st_cong_dist`, `perf_inst`.`perf_ctry_name` AS `perf_inst.perf_ctry_name`, `perf_inst`.`perf_ctry_flag` AS `perf_inst.perf_ctry_flag`, `perf_inst`.`perfInstAwdId` AS `perf_inst.perfInstAwdId`, `perf_inst`.`awardAwdId` AS `perf_inst.awardAwdId`, `pgm_eles`.`id` AS `pgm_eles.id`, `pgm_eles`.`pgm_ele_code` AS `pgm_eles.pgm_ele_code`, `pgm_eles`.`pgm_ele_name` AS `pgm_eles.pgm_ele_name`, `pgm_eles`.`awardAwdId` AS `pgm_eles.awardAwdId`, `pgm_refs`.`id` AS `pgm_refs.id`, `pgm_refs`.`pgm_ref_code` AS `pgm_refs.pgm_ref_code`, `pgm_refs`.`pgm_ref_txt` AS `pgm_refs.pgm_ref_txt`, `pgm_refs`.`awardAwdId` AS `pgm_refs.awardAwdId`, `app_funds`.`id` AS `app_funds.id`, `app_funds`.`app_code` AS `app_funds.app_code`, `app_funds`.`app_name` AS `app_funds.app_name`, `app_funds`.`app_symb_id` AS `app_funds.app_symb_id`, `app_funds`.`fund_code` AS `app_funds.fund_code`, `app_funds`.`fund_name` AS `app_funds.fund_name`, `app_funds`.`fund_symb_id` AS `app_funds.fund_symb_id`, `app_funds`.`awardAwdId` AS `app_funds.awardAwdId`, `oblg_fys`.`id` AS `oblg_fys.id`, `oblg_fys`.`fund_oblg_fiscal_yr` AS `oblg_fys.fund_oblg_fiscal_yr`, `oblg_fys`.`fund_oblg_amt` AS `oblg_fys.fund_oblg_amt`, `oblg_fys`.`awardAwdId` AS `oblg_fys.awardAwdId`, `por`.`id` AS `por.id`, `por`.`por_cntn` AS `por.por_cntn`, `por`.`por_txt_cntn` AS `por.por_txt_cntn`, `por`.`porAwdId` AS `por.porAwdId`, `por`.`awardAwdId` AS `por.awardAwdId` FROM (SELECT `awards`.`awd_id`, `awards`.`agcy_id`, `awards`.`tran_type`, `awards`.`awd_istr_txt`, `awards`.`awd_titl_txt`, `awards`.`cfda_num`, `awards`.`org_code`, `awards`.`po_phone`, `awards`.`po_email`, `awards`.`po_sign_block_name`, `awards`.`awd_eff_date`, `awards`.`awd_exp_date`, `awards`.`tot_intn_awd_amt`, `awards`.`awd_amount`, `awards`.`awd_min_amd_letter_date`, `awards`.`awd_max_amd_letter_date`, `awards`.`awd_abstract_narration`, `awards`.`awd_arra_amount`, `awards`.`dir_abbr`, `awards`.`org_dir_long_name`, `awards`.`div_abbr`, `awards`.`org_div_long_name`, `awards`.`awd_agcy_code`, `awards`.`fund_agcy_code` FROM `awards` AS `awards` LIMIT 0, 50) AS `awards` LEFT OUTER JOIN `pis` AS `pis` ON `awards`.`awd_id` = `pis`.`awardAwdId` LEFT OUTER JOIN `insts` AS `inst` ON `awards`.`awd_id` = `inst`.`instAwdId` LEFT OUTER JOIN `perf_insts` AS `perf_inst` ON `awards`.`awd_id` = `perf_inst`.`perfInstAwdId` LEFT OUTER JOIN `pgm_eles` AS `pgm_eles` ON `awards`.`awd_id` = `pgm_eles`.`awardAwdId` LEFT OUTER JOIN `pgm_refs` AS `pgm_refs` ON `awards`.`awd_id` = `pgm_refs`.`awardAwdId` LEFT OUTER JOIN `app_funds` AS `app_funds` ON `awards`.`awd_id` = `app_funds`.`awardAwdId` LEFT OUTER JOIN `oblg_fys` AS `oblg_fys` ON `awards`.`awd_id` = `oblg_fys`.`awardAwdId` LEFT OUTER JOIN `pors` AS `por` ON `awards`.`awd_id` = `por`.`porAwdId`;
```

```
SELECT `awards`.*, `pis`.`id` AS `pis.id`, `pis`.`pi_role` AS `pis.pi_role`, `pis`.`pi_first_name` AS `pis.pi_first_name`, `pis`.`pi_last_name` AS `pis.pi_last_name`, `pis`.`pi_mid_init` AS `pis.pi_mid_init`, `pis`.`pi_sufx_name` AS `pis.pi_sufx_name`, `pis`.`pi_full_name` AS `pis.pi_full_name`, `pis`.`pi_email_addr` AS `pis.pi_email_addr`, `pis`.`nsf_id` AS `pis.nsf_id`, `pis`.`pi_start_date` AS `pis.pi_start_date`, `pis`.`pi_end_date` AS `pis.pi_end_date`, `pis`.`awardAwdId` AS `pis.awardAwdId`, `inst`.`id` AS `inst.id`, `inst`.`inst_name` AS `inst.inst_name`, `inst`.`inst_street_address` AS `inst.inst_street_address`, `inst`.`inst_street_address_2` AS `inst.inst_street_address_2`, `inst`.`inst_city_name` AS `inst.inst_city_name`, `inst`.`inst_state_code` AS `inst.inst_state_code`, `inst`.`inst_state_name` AS `inst.inst_state_name`, `inst`.`inst_phone_num` AS `inst.inst_phone_num`, `inst`.`inst_zip_code` AS `inst.inst_zip_code`, `inst`.`inst_country_name` AS `inst.inst_country_name`, `inst`.`cong_dist_code` AS `inst.cong_dist_code`, `inst`.`st_cong_dist_code` AS `inst.st_cong_dist_code`, `inst`.`org_lgl_bus_name` AS `inst.org_lgl_bus_name`, `inst`.`org_prnt_uei_num` AS `inst.org_prnt_uei_num`, `inst`.`org_uei_num` AS `inst.org_uei_num`, `inst`.`instAwdId` AS `inst.instAwdId`, `inst`.`awardAwdId` AS `inst.awardAwdId`, `perf_inst`.`id` AS `perf_inst.id`, `perf_inst`.`perf_inst_name` AS `perf_inst.perf_inst_name`, `perf_inst`.`perf_str_addr` AS `perf_inst.perf_str_addr`, `perf_inst`.`perf_city_name` AS `perf_inst.perf_city_name`, `perf_inst`.`perf_st_code` AS `perf_inst.perf_st_code`, `perf_inst`.`perf_st_name` AS `perf_inst.perf_st_name`, `perf_inst`.`perf_zip_code` AS `perf_inst.perf_zip_code`, `perf_inst`.`perf_ctry_code` AS `perf_inst.perf_ctry_code`, `perf_inst`.`perf_cong_dist` AS `perf_inst.perf_cong_dist`, `perf_inst`.`perf_st_cong_dist` AS `perf_inst.perf_st_cong_dist`, `perf_inst`.`perf_ctry_name` AS `perf_inst.perf_ctry_name`, `perf_inst`.`perf_ctry_flag` AS `perf_inst.perf_ctry_flag`, `perf_inst`.`perfInstAwdId` AS `perf_inst.perfInstAwdId`, `perf_inst`.`awardAwdId` AS `perf_inst.awardAwdId`, `pgm_eles`.`id` AS `pgm_eles.id`, `pgm_eles`.`pgm_ele_code` AS `pgm_eles.pgm_ele_code`, `pgm_eles`.`pgm_ele_name` AS `pgm_eles.pgm_ele_name`, `pgm_eles`.`awardAwdId` AS `pgm_eles.awardAwdId`, `pgm_refs`.`id` AS `pgm_refs.id`, `pgm_refs`.`pgm_ref_code` AS `pgm_refs.pgm_ref_code`, `pgm_refs`.`pgm_ref_txt` AS `pgm_refs.pgm_ref_txt`, `pgm_refs`.`awardAwdId` AS `pgm_refs.awardAwdId`, `app_funds`.`id` AS `app_funds.id`, `app_funds`.`app_code` AS `app_funds.app_code`, `app_funds`.`app_name` AS `app_funds.app_name`, `app_funds`.`app_symb_id` AS `app_funds.app_symb_id`, `app_funds`.`fund_code` AS `app_funds.fund_code`, `app_funds`.`fund_name` AS `app_funds.fund_name`, `app_funds`.`fund_symb_id` AS `app_funds.fund_symb_id`, `app_funds`.`awardAwdId` AS `app_funds.awardAwdId`, `oblg_fys`.`id` AS `oblg_fys.id`, `oblg_fys`.`fund_oblg_fiscal_yr` AS `oblg_fys.fund_oblg_fiscal_yr`, `oblg_fys`.`fund_oblg_amt` AS `oblg_fys.fund_oblg_amt`, `oblg_fys`.`awardAwdId` AS `oblg_fys.awardAwdId`, `por`.`id` AS `por.id`, `por`.`por_cntn` AS `por.por_cntn`, `por`.`por_txt_cntn` AS `por.por_txt_cntn`, `por`.`porAwdId` AS `por.porAwdId`, `por`.`awardAwdId` AS `por.awardAwdId` FROM (SELECT `awards`.`awd_id`, `awards`.`agcy_id`, `awards`.`tran_type`, `awards`.`awd_istr_txt`, `awards`.`awd_titl_txt`, `awards`.`cfda_num`, `awards`.`org_code`, `awards`.`po_phone`, `awards`.`po_email`, `awards`.`po_sign_block_name`, `awards`.`awd_eff_date`, `awards`.`awd_exp_date`, `awards`.`tot_intn_awd_amt`, `awards`.`awd_amount`, `awards`.`awd_min_amd_letter_date`, `awards`.`awd_max_amd_letter_date`, `awards`.`awd_abstract_narration`, `awards`.`awd_arra_amount`, `awards`.`dir_abbr`, `awards`.`org_dir_long_name`, `awards`.`div_abbr`, `awards`.`org_div_long_name`, `awards`.`awd_agcy_code`, `awards`.`fund_agcy_code` FROM `awards` AS `awards` WHERE (`awards`.`awd_id` LIKE '%0002470%' OR `awards`.`po_email` LIKE '%0002470%' OR `awards`.`awd_titl_txt` LIKE '%0002470%' OR `awards`.`po_phone` LIKE '%0002470%' OR `awards`.`po_sign_block_name` LIKE '%0002470%' OR `awards`.`awd_id` LIKE '%0002470%' OR `awards`.`po_email` LIKE '%0002470%' OR `awards`.`awd_titl_txt` LIKE '%0002470%' OR `awards`.`po_phone` LIKE '%0002470%' OR `awards`.`po_sign_block_name` LIKE '%0002470%' OR `awards`.`awd_eff_date` LIKE '%0002470%') LIMIT 0, 50) AS `awards` LEFT OUTER JOIN `pis` AS `pis` ON `awards`.`awd_id` = `pis`.`awardAwdId` LEFT OUTER JOIN `insts` AS `inst` ON `awards`.`awd_id` = `inst`.`instAwdId` LEFT OUTER JOIN `perf_insts` AS `perf_inst` ON `awards`.`awd_id` = `perf_inst`.`perfInstAwdId` LEFT OUTER JOIN `pgm_eles` AS `pgm_eles` ON `awards`.`awd_id` = `pgm_eles`.`awardAwdId` LEFT OUTER JOIN `pgm_refs` AS `pgm_refs` ON `awards`.`awd_id` = `pgm_refs`.`awardAwdId` LEFT OUTER JOIN `app_funds` AS `app_funds` ON `awards`.`awd_id` = `app_funds`.`awardAwdId` LEFT OUTER JOIN `oblg_fys` AS `oblg_fys` ON `awards`.`awd_id` = `oblg_fys`.`awardAwdId` LEFT OUTER JOIN `pors` AS `por` ON `awards`.`awd_id` = `por`.`porAwdId`;
```