W4111 – Introduction to Databases
Section 002, Spring 2025
Introduction to Project Template

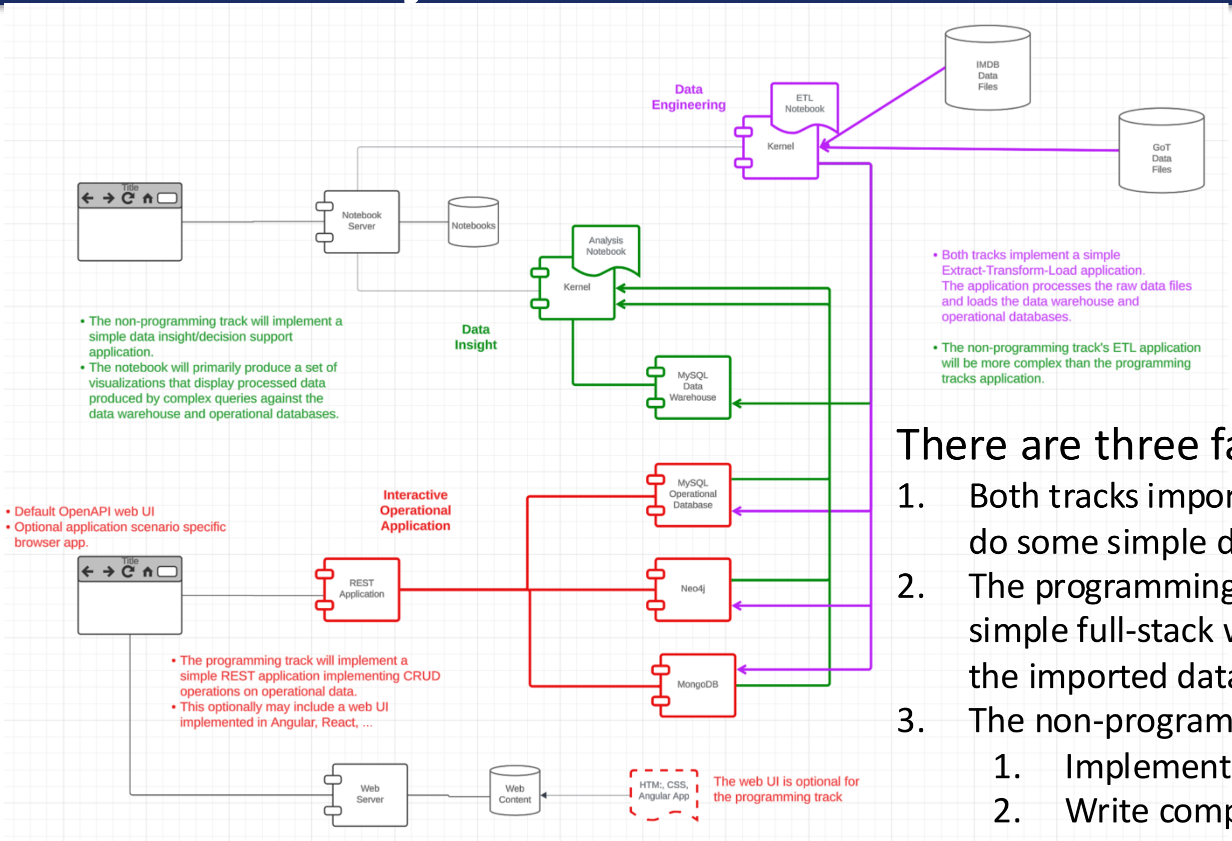
COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

Introduction

Two Common Database Applications

- Operational/Interactive:
  - Users and roles can create, retrieve, update, search and delete “records.”
  - Examples: SSOL, ATMs, … …
- Business Intelligence, Decision Support, …:
  - Users can perform complex queries and analyze a lot of data to generate a report, make a decision, … …
  - Examples: Build AI/ML training data, dashboards, reports, … …
- Some of our major datasets this semester will be:
  - IMDB: https://developer.imdb.com/non-commercial-datasets/
  - Game of Thrones: https://developer.imdb.com/non-commercial-datasets/
  - Lahman’s Baseball Dataset: http://seanlahman.com/
  - … …
- We will build a simple web application and do some data engineering.

Overall Project Structure



- Both tracks implement a simple Extract-Transform-Load application. The application processes the raw data files and loads the data warehouse and operational databases.
- The non-programming track's ETL application will be more complex than the programming tracks application.

- The non-programming track will implement a simple data insight/decision support application.
- The notebook will primarily produce a set of visualizations that display processed data produced by complex queries against the data warehouse and operational databases.

- Default OpenAPI web UI
- Optional application scenario specific browser app.

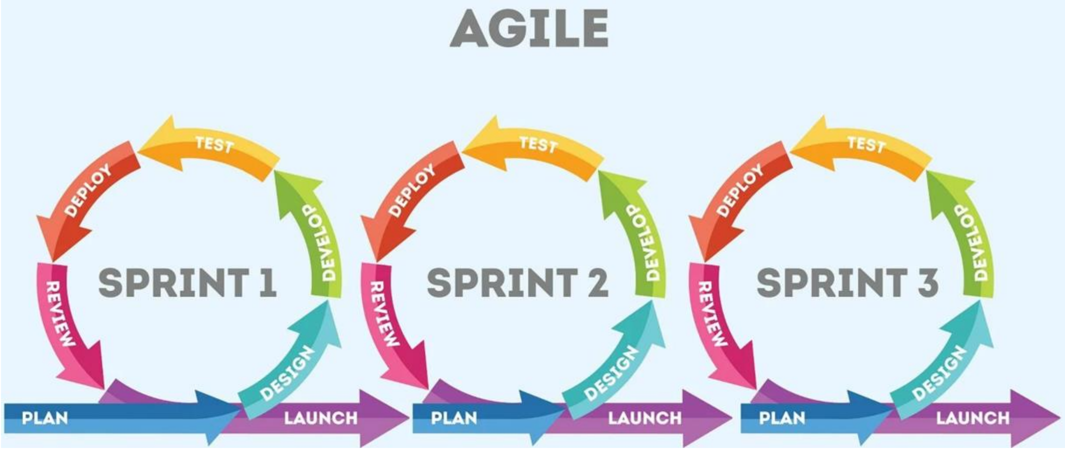
- The programming track will implement a simple REST application implementing CRUD operations on operational data.
- This optionally may include a web UI implemented in Angular, React, …

The web UI is optional for the programming track

There are three facets to the course project:

1. Both tracks import data from files and do some simple data engineering.
2. The programming track implements a simple full-stack web application that uses the imported data.
3. The non-programming track:
    1. Implements more challenging data engineering.
    2. Write complex queries that produce data used for visualization.

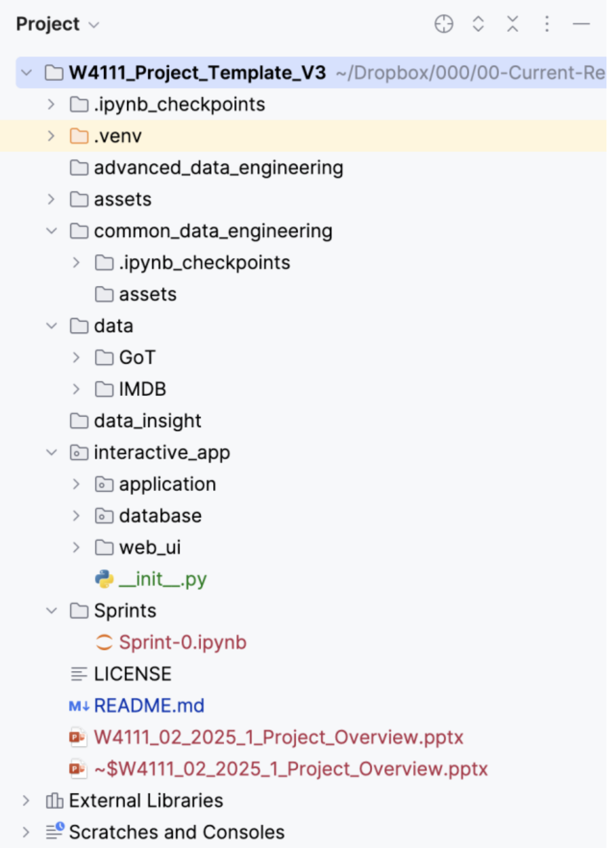
Agile Development



- You will do a very simple form of Agile Development.
  Review involves submitting your homework for grading.
- Sprint 1 is simply to: 1) Demonstrate that you can clone and execute the project template. 2) Submit your proposed dataset for approval.
- Sprint 2 will implement the tasks using SQL.
- Sprint 3 will add MongoDB and Neo4j.

Or, you can just use the sample dataset I use.

Project Structure

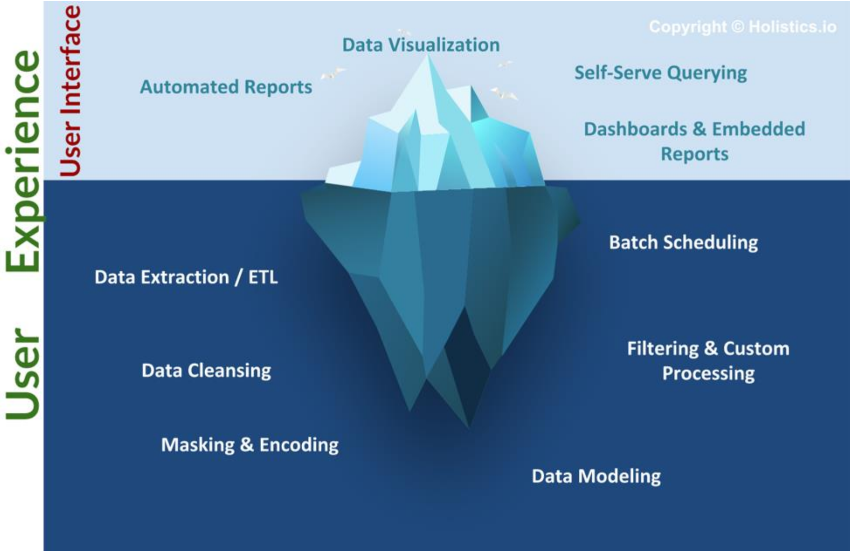


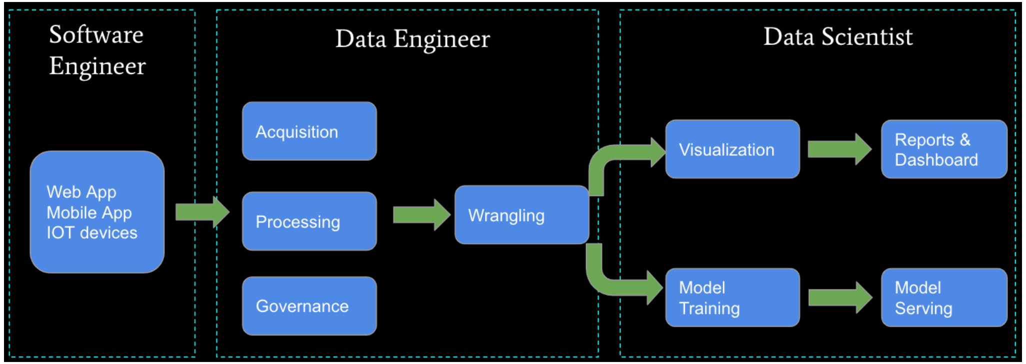
```
W4111_Project_Template_V3  ~/Dropbox/000/00-Current-Re
  .ipynb_checkpoints
  .venv
  advanced_data_engineering
  assets
  common_data_engineering
    .ipynb_checkpoints
    assets
  data
    GoT
    IMDB
  data_insight
  interactive_app
    application
    database
    web_ui
    __init__.py
  Sprints
    Sprint-0.ipynb
  LICENSE
  README.md
  W4111_02_2025_1_Project_Overview.pptx
  ~$W4111_02_2025_1_Project_Overview.pptx
External Libraries
Scratches and Consoles
```

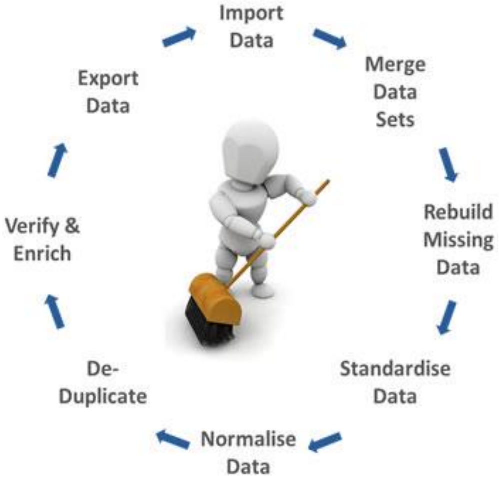
- Non-programming:
  - advanced_data_engineering will contain examples and templates of advanced data engineering tasks.
  - data_insight will contain examples and templates for complex queries and visualization.
- Programming is in interactive_app and there are examples and templates"
  - application is the middle-tier "business logic."
  - database holds DDL and other setup scripts.
  - web_UI is a simple, Angular web UI.
- Common
  - assets holds miscellaneous images, etc.
  - common_data_engineering will contain examples and templates for the data engineering.
  - data holds the datasets for Prof. Ferguson's examples.
  - Sprints will contain the templates for sprints.

Common Data Engineering

Business Intelligence, Insight, Analysis, … …





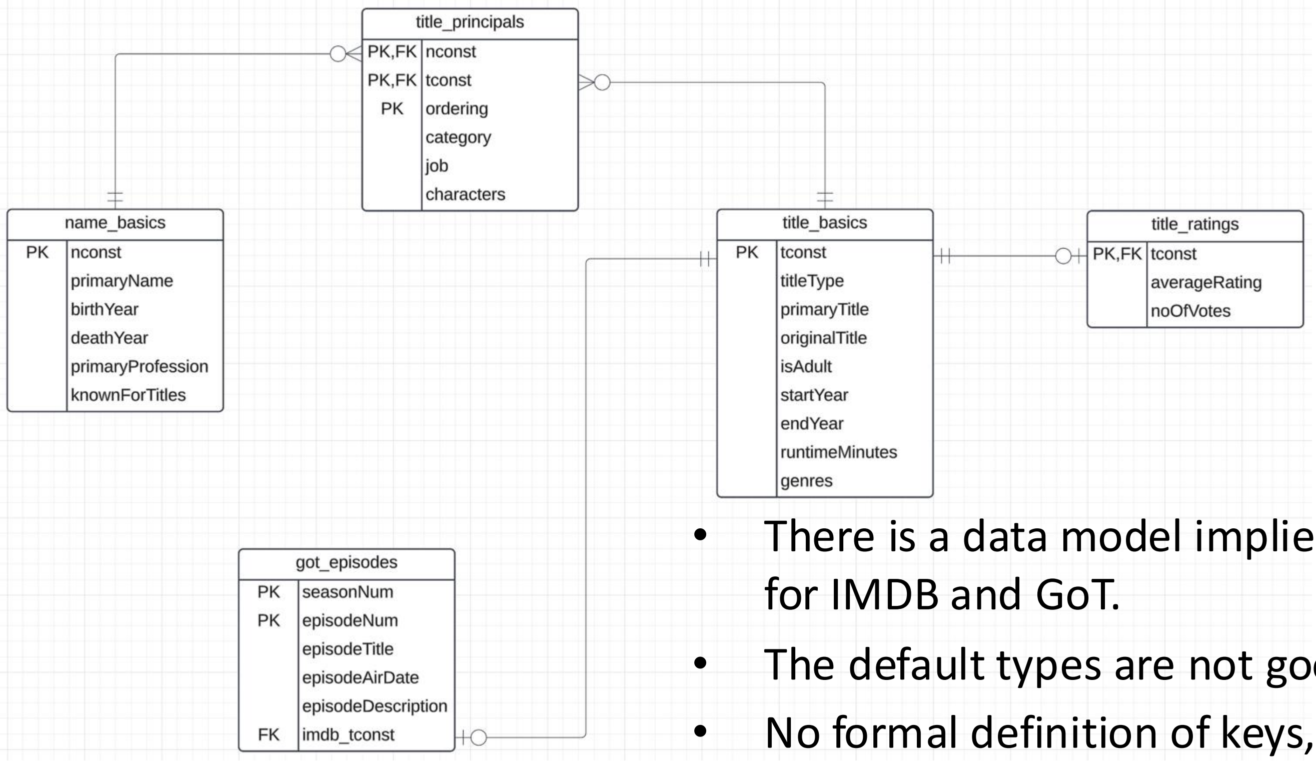


- The “fun” stuff in data science and AI/ML is the “tip of the iceberg.”
- Data engineering is a necessary condition for producing analyzable data. This is often more than 80% of the hard work.
- We will do some small data engineering projects in this course.

Common Data Engineering Tasks

- Input file formats will be CSV and JSON.
- Read and import data into a Jupyter notebook.
- Perform "shallow" export of data to MySQL, Neo4j, MongoDB
- Perform basic "hygiene" on imported data:
  - Data types, indexes, keys, … …
  - Fix data, e.g. split and link, fill in missing values, ignore bad data, … …
  - … …
- Implement and test views to simplify application development and complex queries.

Initial Data – Example



title_principals	
PK,FK	nconst
PK,FK	tconst
PK	ordering
	category
	job
	characters

name_basics	
PK	nconst
	primaryName
	birthYear
	deathYear
	primaryProfession
	knownForTitles

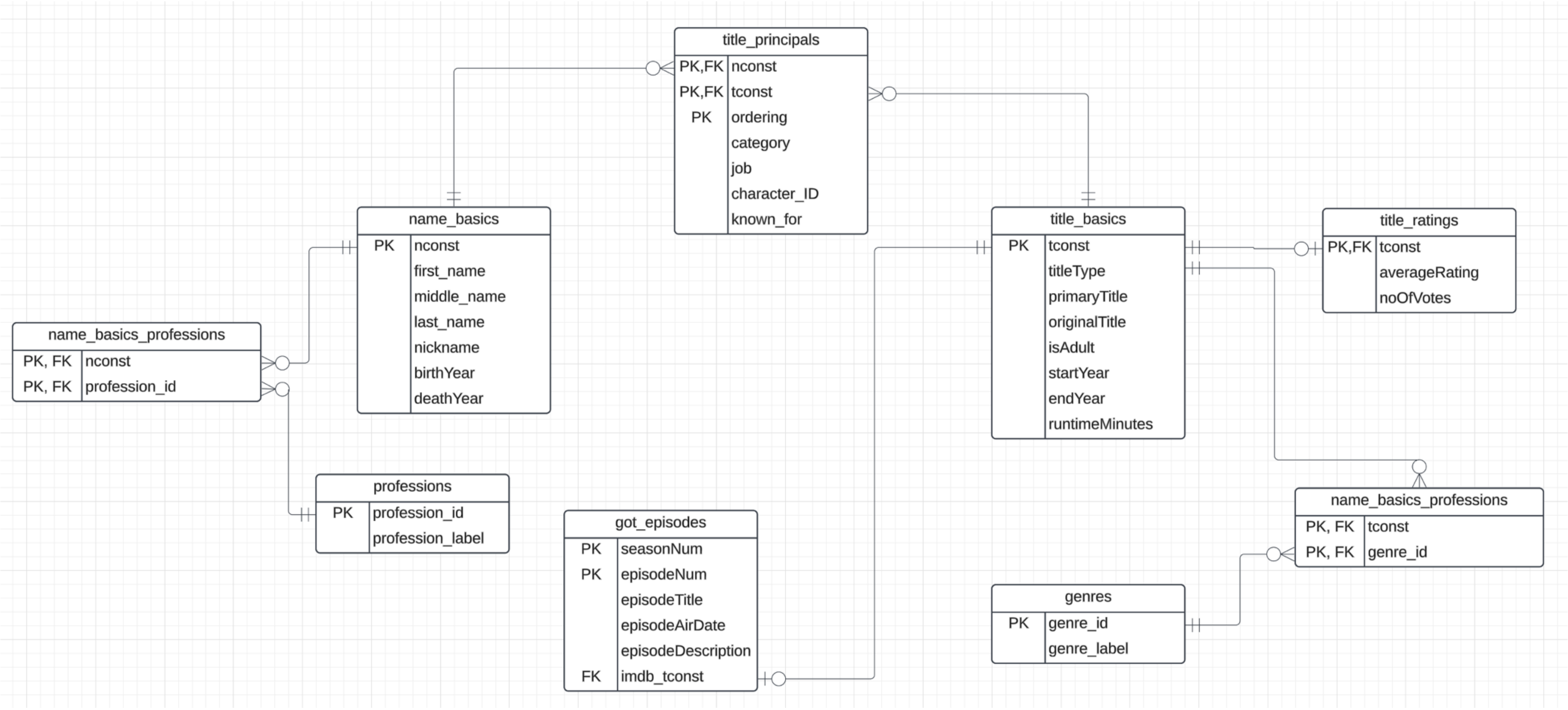
title_basics	
PK	tconst
	titleType
	primaryTitle
	originalTitle
	isAdult
	startYear
	endYear
	runtimeMinutes
	genres

title_ratings	
PK,FK	tconst
	averageRating
	noOfVotes

got_episodes	
PK	seasonNum
PK	episodeNum
	episodeTitle
	episodeAirDate
	episodeDescription
FK	imdb_tconst

- There is a data model implied by the raw input files for IMDB and GoT.
- The default types are not good, e.g. TEXT, DOUBLE.
- No formal definition of keys, foreign keys, … …
- Basic constraints missing.

Initial Cleanup – Example



	title_principals
PK,FK	nconst
PK,FK	tconst
PK	ordering
	category
	job
	character_ID
	known_for

	name_basics
PK	nconst
	first_name
	middle_name
	last_name
	nickname
	birthYear
	deathYear

	title_basics
PK	tconst
	titleType
	primaryTitle
	originalTitle
	isAdult
	startYear
	endYear
	runtimeMinutes

	title_ratings
PK,FK	tconst
	averageRating
	noOfVotes

	name_basics_professions
PK, FK	nconst
PK, FK	profession_id

	professions
PK	profession_id
	profession_label

	got_episodes
PK	seasonNum
PK	episodeNum
	episodeTitle
	episodeAirDate
	episodeDescription
FK	imdb_tconst

	name_basics_professions
PK, FK	tconst
PK, FK	genre_id

	genres
PK	genre_id
	genre_label

Advanced Data Engineering and Insight

Advanced Data Engineering and Insight

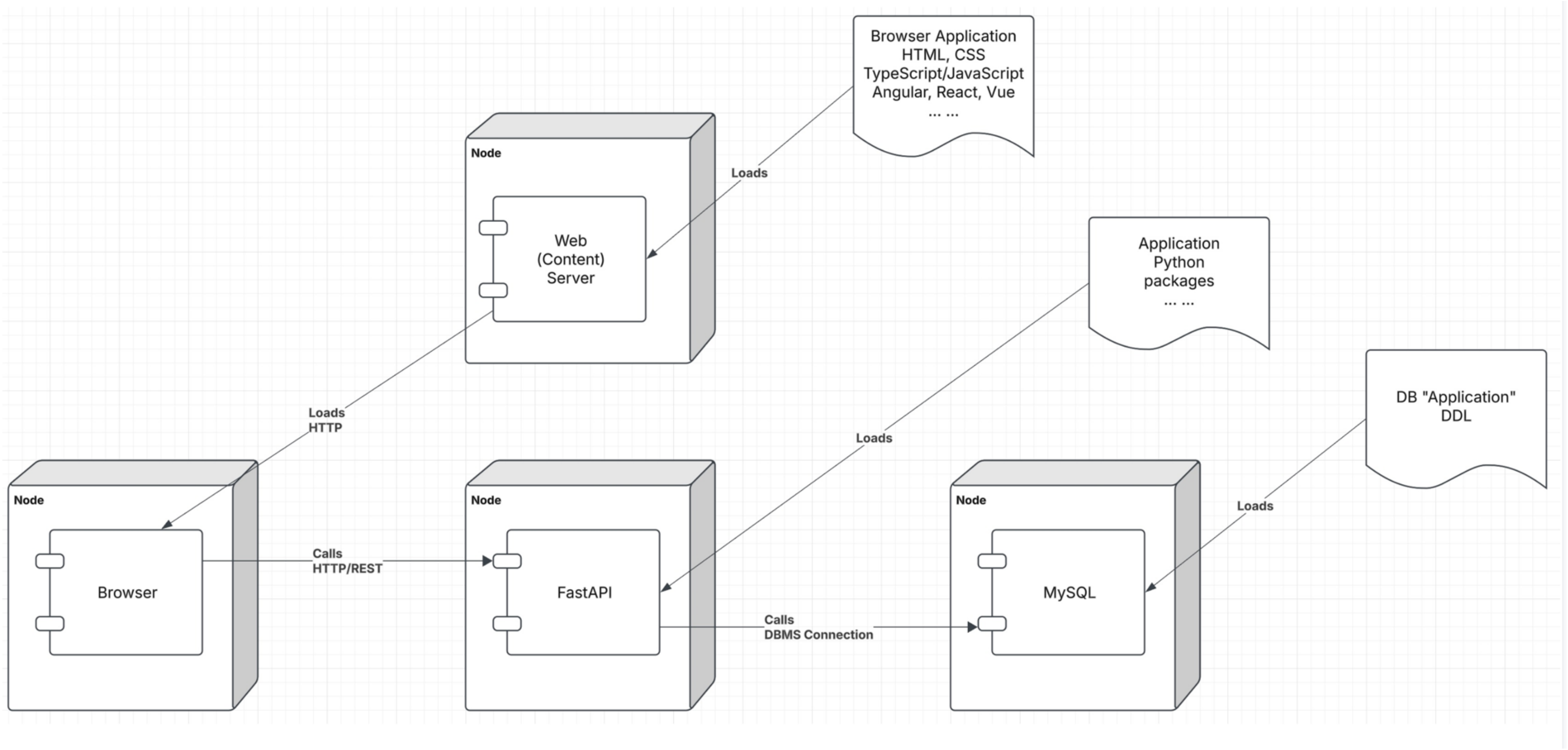
- Data engineering is an extremely complex and challenging topics and environment. There are also complex, powerful specialized tools.
- We cannot cover the complexities and tools in sufficient detail in this class.
- So, data engineering will be confined to
  - Complex queries to extract data from Neo4j and MongoDB and load into MySQL.
  - Defining a data model that better supports complex query.
  - Transforming the loaded files, MongoDB and Neo4j data into the to be schema using relatively complex SQL
  - Creating views using complex SQL.

Example

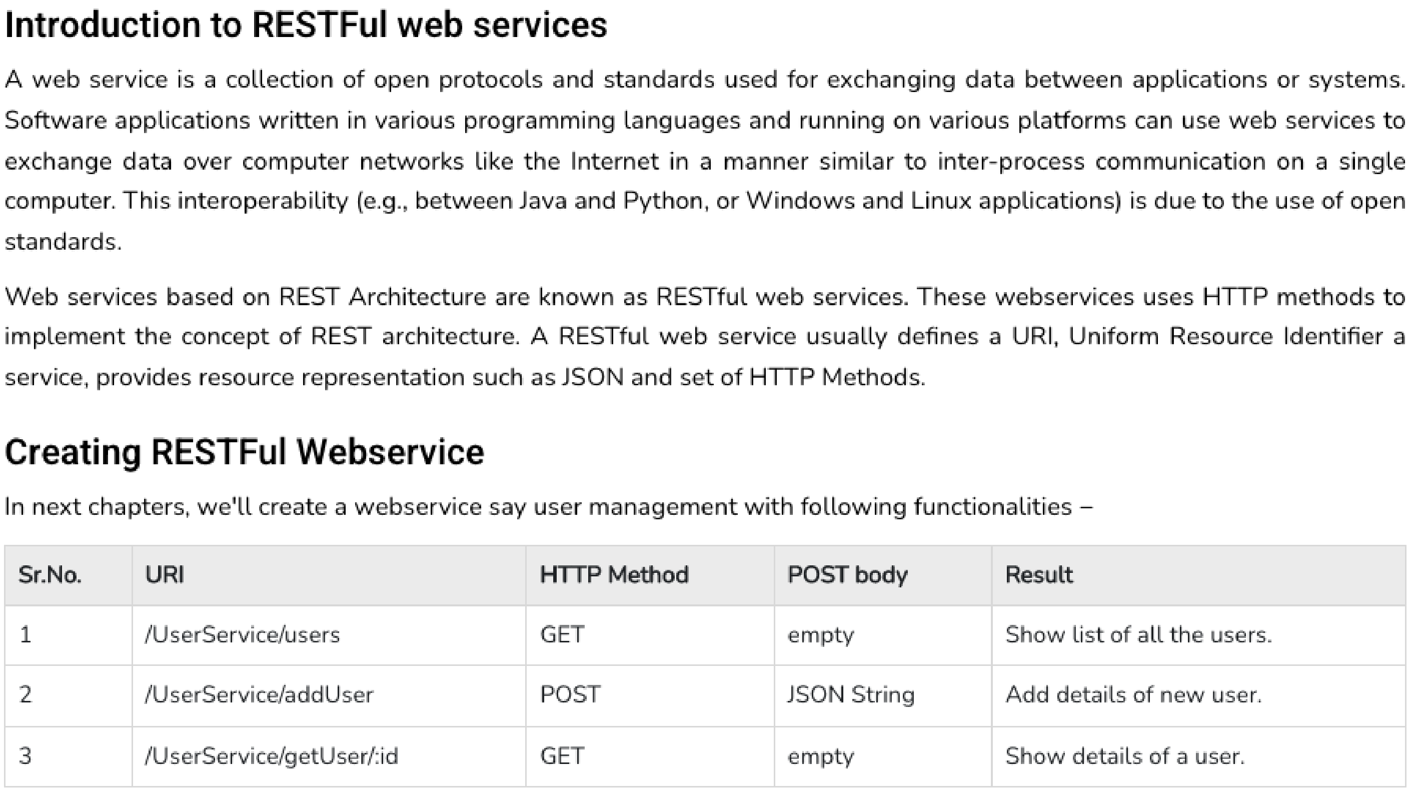
```
with one as (select, nconst, name_basics.primaryProfession,
        substr(name_basics.primaryProfession, 1, locate(',', name_basics.primaryProfession)-1) as p1,
        substr(name_basics.primaryProfession, locate(',', name_basics.primaryProfession)+1)  as remainder
    from name_basics),
    two as ( select nconst, one.primaryProfession,
            p1, substr(remainder, 1, locate(',', remainder) -1) as p2,
          substr(remainder, locate(',', remainder)+1)  as p3 from one),
    three as ( select nconst, primaryProfession,
            if(p1='', NULL, p1) as p1, if(p2='', NULL, p2) as p2, if(p3='', NULL, p3) as p3 from two),
    four as (select nconst, p1 as profession
             from three union select nconst, p2 as profession from three
             union select nconst, p3 as profession from three)
select
    *
from four;
```

Full-Stack Web Application

Let’s Look at the Programming Project



REST (https://www.tutorialspoint.com/restful/restful_introduction.htm)



Introduction to RESTFul web services

A web service is a collection of open protocols and standards used for exchanging data between applications or systems. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer. This interoperability (e.g., between Java and Python, or Windows and Linux applications) is due to the use of open standards.

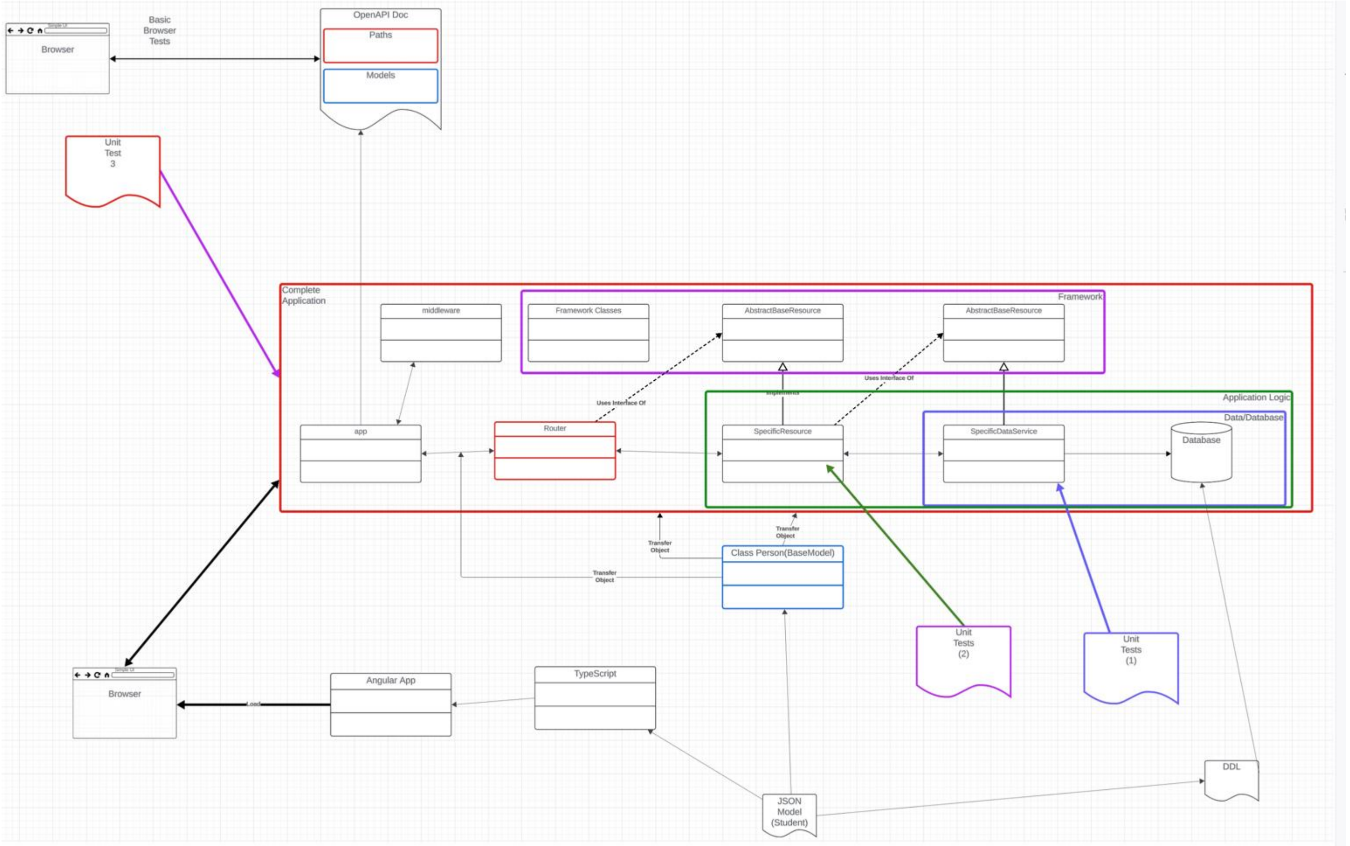
Web services based on REST Architecture are known as RESTful web services. These webservices uses HTTP methods to implement the concept of REST architecture. A RESTful web service usually defines a URI, Uniform Resource Identifier a service, provides resource representation such as JSON and set of HTTP Methods.

Creating RESTFul Webservice

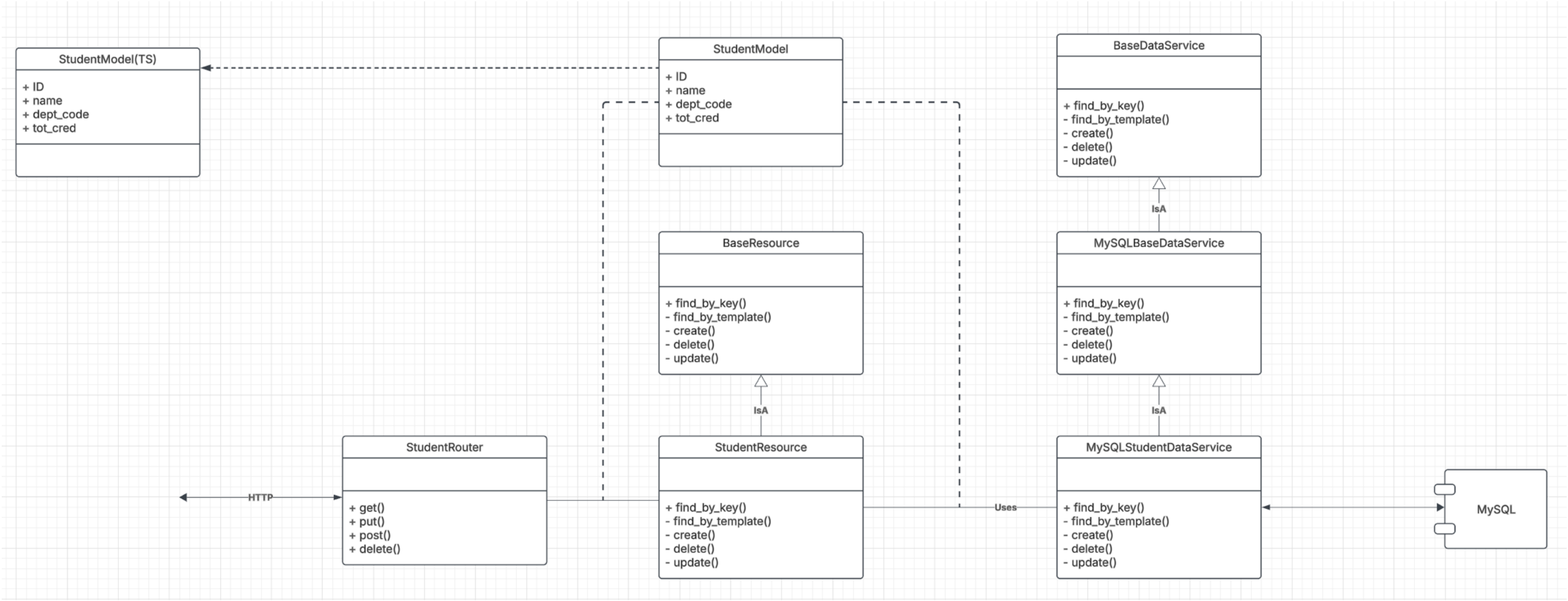
In next chapters, we'll create a webservice say user management with following functionalities –

Sr.No.	URI	HTTP Method	POST body	Result
1	/UserService/users	GET	empty	Show list of all the users.
2	/UserService/addUser	POST	JSON String	Add details of new user.
3	/UserService/getUser/:id	GET	empty	Show details of a user.

Let’s Look at the Programming Project



Let’s Look at the Programming Project



Sprint 0

Sprint 1 – Homework 3B

Very simple

1. Clone the project.
2. Install the required packages. (pip and requirements.txt)
3. Demonstrate that you can execute all of the cells in the Sprint-1 notebook.
4. Provide a short description or link to the dataset you want to use, or indicate that you plan to use the IMDB, GoT sample data.
5. You can try to execute interactive_app/application/main.py
   and go to localhost:8001/docs
6. You can try to install Angular (https://v17.angular.io/guide/setup-local) and execute interactive_app/web_ui, but we will do another tutorial.