SQL Query Generator Using ReactJS

MID SEMESTER REPORT

SEWP ZG628T DISSERTATION

By

Harinath G

2018HW70092

Dissertation Work carried out at

Wipro Technologies, Chennai CDC-2

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
PILANI (Rajasthan) India

November 2022

SEWP ZG628T DISSERTATION

SQL Query Generator Using ReactJS

Submitted in partial fulfillment of the requirements of

M.S. Software Engineering Degree Program

By
Harinath G
2018HW70092

Under the supervision of

Rudrabhatla Vakula
Senior project Engineer

Dissertation Work carried out at

Wipro Technologies, Chennai CDC-2

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
PILANI (Rajasthan) India

(November 2022)

BIRLA INSTITUTE OF TECHNOLOGY, PILANI

CERTIFICATE

This is to certify that the Dissertation entitled **SQL Query Generator Using ReactJS** and submitted by **Harinath G** ID No: **2018HW70092** in partial fulfillment of the requirement of **SEPWZG628T** Dissertation embodies the work done by him/her under my supervision.

R. Vakula

Date: 18.11.2022

Signature of the Supervisor

Name: Rudrabhatla Vakula

Designation: Senior project Engineer

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

First Semester 2022-23

SEWP ZG628T DISSERTATION

MID SEMESTER EVALUATION FORM

Section I

ID No	: **2018HW70092**
Name of Student	: **Harinath G**
Name of Supervisor	: **Rudrabhatla Vakula**
Name of the Examiner(s)	: **Divya Lakshmi S** **Asha Goduguchinta**
Dissertation Title	: **SQL Query Generator Using ReactJS**

Section II

Comments on the dissertation from Examiner and Supervisor (Select Y or N)

1. **Quantum of work**
    a. **Justifiable as efforts for 8 weeks duration** — [✓ **Y**] / **N**
    b. **Work is in line with the commitments made in outline** — [✓ **Y**] / **N**
2. **Type of work**
    a. **Client assignment** — [✓ **Y**] / **N**
    b. **Organization specific task** — **Y** / [✓ **N**]
    c. **General study project such as white paper** — **Y** / [✓ **N**]
    d. **Any other (kindly elaborate below in a line or two if Y** — **Y** / [✓ **N**]

3. **Nature of work**
   a. **Routine in nature** **Y / [✓ N]**
   b. **Involved creativity and rational thinking** **[✓ Y] / N**
   Kindly elaborate below if answer for above is "Y"

Editing and Executing "N" number of queries is very tedious job and time taking job. With the help if this tool, we can automate the activity which will reduce the efforts and time taken. This can be incorporated across other projects which involve editing multiple queries by copying from excel and executing them.

4. **Evaluation methodology**
   a. **Evaluation done based on presentation to supervisor and examiner** **[✓ Y] / N**
   b. **Evaluation done through Viva conducted by supervisor and examiner** **[✓ Y] / N**
   c. **Student regularly interacted with supervisor and incorporated the suggestions made** **[✓ Y] / N**
   d. **Brief description of the report submitted, quality of presentation and suggestions given for improvement**

The progress of the project is in line with the timelines mentioned at the beginning. This tool is at a very good stage now where most of the coding related activities have been completed by Harinath. I don't see any scope of improvement as he has delivered the project very meticulously as of today.

5. **Mid semester evaluation matrix:**
Tick the appropriate box (1 is lowest and 5 is the highest):

Dimension **Rank ➔**	**1**	**2**	**3**	**4**	**5**
Student abilities in general					
Understanding of the subject of dissertation					✓
Creative thinking ability to come up with new ideas					✓
Viva / Seminar presentation					
Communication ability					✓
Organization of material					✓
Response to review questions					✓
Cohesive thinking ability					✓
Report submitted					
Report structure and format					✓
Technical content of the report					✓
Explanation on the significance of the assignment					✓
Analysis of alternative approaches					✓

[illegible]

G.H. Asha

R. Vakula

Date: **18.11.2022**

Signature of examiner(s)

Name 1: Divya Lakshmi S

Name 2: Asha Goduguchinta

Signature of Supervisor

Name: Rudrabhatla Vakula

SQL Query Generator Using ReactJS

ACKNOWLEDGEMENTS

A good supervisor is required to advise a pupil on the right route based on his or her expertise. **Rudrabhatla Vakula**, my supervisor, carefully examined each step and patiently instructed me wherever necessary. Thank you so much for your assistance.

I'd like to express my heartfelt gratitude to my project manager, project lead, and project colleagues for their assistance and cooperation in providing feedback on the project at each stage and sharing their thoughts on how to improve the quality of the application and add more features to make it even smarter.

My examiners, **Divya Lakshmi** S and **Asha Goduguchinta**, are indeed assisting me with this project. My heartfelt gratitude to them.

I am also grateful to **WIPRO** Technologies for providing me with this chance and all of the materials needed for this dissertation.

Thank you to **BITS, PILANI**, and my instructors for constantly encouraging me to learn more.

[illegible]

Harinath G
2018HW70092

ABSTRACT

Writing the same SQL queries with different data too many times is a time-consuming and tedious chore for our team. It is also more prone to make mistakes, which demands more work from our crew.

So here is my solution that solves most of the above-mentioned concerns by creating the SQL queries that we need by accepting minimal inputs and generating queries in minutes rather than hours/days doing it manually, and it greatly reduces/eliminates the risks of errors.

Because this tool comprises only a webpage, it can be used on any platform with any web browser (recommended Chrome). This tool is incredibly simple to use and requires no training. I am utilizing web technologies such as (HTML, CSS, ReactJS, and so on) for front-end development and (JavaScript) for back-end processing, as well as tools such as (VSCode and GitHub) for coding and repository administration.

Table of contents

Contents

List of Symbols & Abbreviations used

The following are the acronyms and symbols that I used in this document:

➢ **HTML**	: Hyper Text Markup language
➢ **CSS**	: Cascading Style Sheets
➢ **JS**	: JavaScript
➢ **JSX**	: JavaScript XML
➢ **DB**	: Data Base
➢ **SQL**	: Structured Query Language

List of Tables / Figures

CHAPTER – 1

Introduction

1.1.Title of the project:

SQL Query Generator Using ReactJS.

1.2.Overview of project:

a. About

My team's typical task is to insert/update data into the database that developers provide in Excel format. So, when we have a large quantity of data to update, we copy the same insert query the appropriate number of times and then manually copy each field data from excel and paste it in the correct position in SQL query, which takes a long time and has a high possibility of error. So, my project will assist my team in producing SQL queries in a timely and effective manner.

b. Problem Statement

For example, if we get 100 items to be inserted in DB then we must paste the insert query 100 times and then manually copy the data from excel and paste it into the queries which takes hours/ days of time and even has many chances to make the mistakes while doing it manually.

c. Goal

My major aim for this project is to minimise the amount of time we spend writing queries from hours/days to minutes and to reduce or eliminate the risks of errors while transferring data from excel, which resulted in another update to correct that mistake.

d. Objective

This application will assist my team in automatically generating/replicating SQL queries in an uncomplicated manner rather than just copying/pasting the data for hours/days and would also lessen their possibilities of making mistakes when copying/pasting the data (For example: having unwanted spaces in a field which leads to the errors for developers and takes another update to rectify that).

e. Scope

- Reduce the amount of time you spend on tedious copy/paste tasks.
- Allows the team to concentrate on more exciting tasks.
- Reduces the likelihood of making mistakes, which takes more time to correct.
- Keeps track of who makes the queries and for whom this update is being performed, which is difficult to accomplish manually.

1.3.Background of the process:

- ➢ This tool will take the SQL query and excel data as input and replicate the same query required number of times based on number of rows that excel sheet has.
- ➢ Only the data which is provided in excel will be changed for every query and the remaining data will be same as in the query that we provided as input.
- ➢ It will remove the unwanted spaces in the data of excel before generating the queries.
- ➢ Finally, after generating the queries, it will be available to copy and run it in SQL query runner.

1.4.Benefits:

Below are the benefits that my team will get if I develop this project.

- ★ It makes life easier.
- ★ This encourages us to concentrate on more exciting tasks.
- ★ This eradicates the probability that you'll make errors, which is highly probable when performing it manually.
- ★ Keeps track of who and for whom queries are produced, which will come in handy if there are any issues about those changes in the future.

CHAPTER – 2

Existing work vs My work

Below is the comparison between already existing work and work done with my project.

Topic	Existing Work(Manual Query writing)	My Work(Generating Queries automatically)
Time	Preparing Queries manually takes hours / days of time for huge data.	This project helps my team to generate Queries in minutes with just few clicks.
Errors	While preparing queries manually we did mistake many times while copy pasting the data.	Here the chances of errors are less even I can assure it is almost no error solution.
Validations	In manual work we need to do the validations like checking for extra spaces or unwanted characters by our self.	Here I did write validations in background which eliminates unwanted spaces and characters.
Required Knowledge	To write queries manually basic level of SQL knowledge is required.	To use this application and generate queries no SQL knowledge is required.
Efficiency	Efficiency of result is less here due to it is error prone.	Efficiency of results is high here since chances of errors are less.
Tracking	Tracking the manual changes are not done due to it requires more queries to be created.	Here tracking would be easier since those queries will also be generated.

CHAPTER – 3

Software Development Life Cycle I Used

For this project planning I am using Agile Software Development Life Cycle.

2.1. What is Agile Software Development?

The Agile methodology is a technique of project management that divides a project into stages. It entails ongoing engagement with stakeholders as well as continual development at each level. When the job begins, teams go through a cycle of planning, execution, and evaluation.

Following figure shows the continuous flow of different phases of Agile Method.

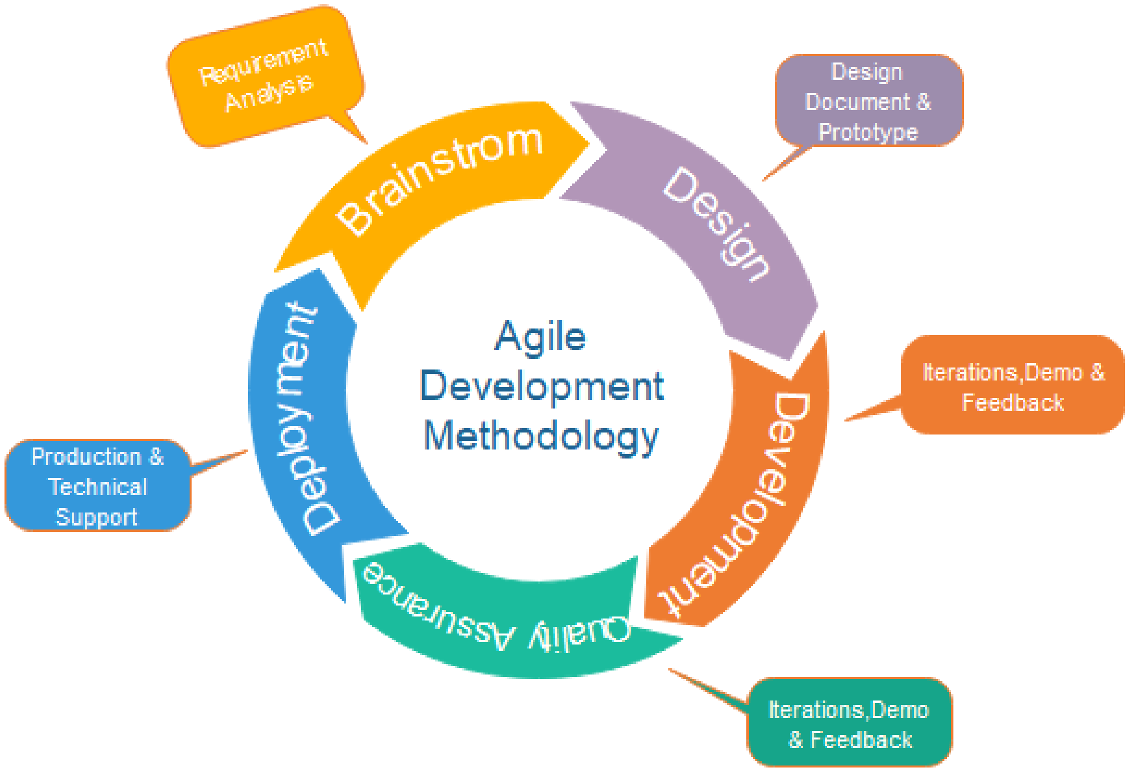

Requirement Analysis
Brainstrom
Design Document & Prototype
Design
Iterations,Demo & Feedback
Development
Iterations,Demo & Feedback
Quality Assurance
Production & Technical Support
Deployment
Agile Development Methodology

Figure 1.Agile Method

2.2. Why is Agile Software Development Method used in this Project?

Here the required features of the application are not well defined, and it can be changed time to time.

When the product vision or features are not well defined, agile works well. Agile enables product owners to change requirements and objectives as they go along to capitalise on opportunities and eventually provide a superior product to all project stakeholders.

I'm adopting Agile SDLC for this project because I don't want to construct a project once and then hand it over to the customer even if it doesn't fulfil their requirements. I want to keep upgrading the tool till the clients are perfectly satisfied.

My customer wants this application to be smarter, thus I am in constant communication with them to gather requirements not only at the beginning of the project, but also at any phase of project, which makes the product smarter and more efficient.

In this first iteration, I am designing and building with the basic needs to perform the desired work, and in subsequent iterations, I am gathering feedback from my clients to add features to make this tool even better.

CHAPTER – 3

Requirement Analysis

3.1. Functional Requirements:

Functional requirements for this application are listed below.

- When the URL is clicked/visited it should be landed to the home page where it has a form to take multiple inputs.
- The form should not be submitted until all needed inputs are provided, and these should be checked and displayed as errors if invalid or submitted if valid.
- After submitting the input form, it should generate complete required data in background.
- The home page should then be forwarded to the preview data page, where the generated data should be shown in the form of a table so that the data may be viewed and checked before producing the queries.
- To eliminate duplication, the preview data page should have a Check for Duplicates button that produces a select query to check whether any of the provided data already exists in the database.
- After pressing the Check for Duplicates button, the needed select query should be produced in the background and displayed on a new webpage called ViewSelectQuery.
- Resubmit the form and Generate Final Queries should be available on the ViewSelectQuery page.
- If there is any duplication in the database, the button will offer the option to resubmit the form, which will redirect to the homepage and give the inputs with a fresh data sheet with the duplicate items eliminated.
- If there is no duplication in the database, the button will provide the option to produce the final queries in the background using the previously generated complete data.
- Finally, the website is navigated to a new web page named ViewFInalQueries, which displays the created final queries.
- Instead of choosing all the queries and copying them, ViewFInalQueries and ViewSelectQuery should include a function that allows them to copy the whole query with a single click.

3.2. Non - Functional Requirements:

Non-Functional requirements for this application are listed below.

3.2.1. Usability:

- The application should be useful and user-friendly, allowing users to traverse its interface with ease.
- The system should be straightforward, and users should be able to figure out what it is capable of.
- The application should function properly, which means that the application's functionality should perform as expected by the user.
- The application's user interface (UI) should be neat and pleasant, with few colours and font styles.
- The created data should be shown in a tabular manner on the PreviewData page, with no missing data, so that it can be readily examined and checked that all the data generated is correct.
- To make the queries more understandable, while displaying queries, each query should be shown as a new line.
- Each query should be terminated with a semicolon (;) to distinguish it from other queries when copying and executing in the database.

3.2.2. Performance:

- The response and processing time for each page should be less than a minute while generating in the background and showing the result.
- The processing pace of the system should be high.
- The results should be efficient and require little manual work.

3.2.3. Portability:

- This web application should work with any operating system and web browser.
- This application should be accessible from several devices at the same time.

3.2.4. Extensibility:

- The application should be able to add new features at any stage.
- A system's ability to adapt to future changes through flexible architecture, design, or execution.

3.3. System Requirements:

The system requirements required to run/use this application without interruption are listed below.

- This tool may be run on any device, such as a mobile phone, tablet, or computer, however it is best used on a laptop or computer.
- This can be used on any operating system, including Windows, Linux, and Android.
- It is preferable to use a 4GB or greater RAM compatible machine to avoid response latency.
- Because this is a web-based application, a web browser such as Chrome, Firefox, or Edge is necessary to use it, however Chrome is preferred for uninterrupted operation.
- MS-Excel is required to modify and produce an excel sheet with data that must be updated in the database, and this excel sheet is used to submit in the application for query creation.

CHAPTER – 4

Design Documentation

4.1. Component Diagram:

A component diagram, often known as a UML component diagram, illustrates how physical components in a system are structured and wired together. Component diagrams are frequently used to assist model implementation details and double-check that planned development covers all part of the system's essential functions.

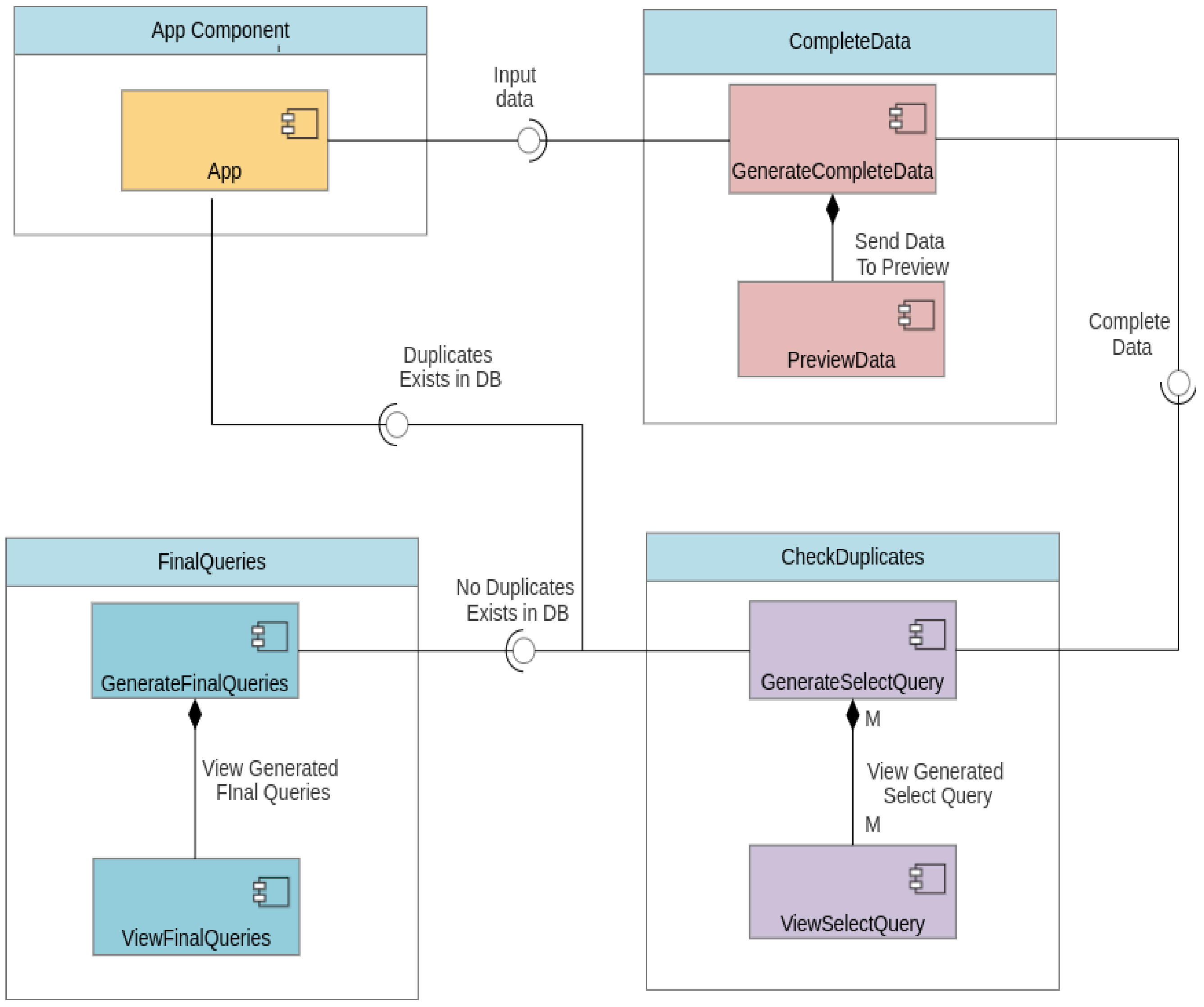


Figure 2.Component Diagram

Figure 2 depicts all of the major components of this project as well as the communication between them. The same is explained in more detail below.

App Component:

This is the primary component that serves as the project's home page/landing page, and the project execution will begin with this component, which is connected to index.html.

This component mostly features an html input for the user to give data for query creation, as well as a few functions that check or handle the data based on the inputs supplied, before sending data to the following component, CompleteData Component.

CompleteData Component:

This component is divided into two parts. One is GenerateCompleteData, which is the component that gets data from the App component and generates the entire data needed to make queries and stores it in local storage of web browser.

The next component is the PreviewData component, which has html code for displaying the created data to the user to determine whether the data appears okay.

CheckDuplicates Component:

This component is used to verify the database for duplication. This has two sub-components as well.

The first is GenerateSelectQuery, which accesses the locally stored complete data and creates a select query with the names of items that can be performed in SQL to see whether there are any items in the database with the same name to avoid duplication.

The second one is ViewSelectQuery, which has html code for displaying the created select query to the user, which can be duplicated and executed in SQL. And contains a button that may be pressed to proceed with final query creation if no duplicates exist, or to send the user to the App component again to resubmit the data if duplicates exist.

FinalQueries Component:

This component will prepare the final queries that will be used to update the needed data in the database. This also contains two sub-components.

The first method is GenerateFinalQueries, which reads the locally stored full data and generates final insert/update queries that can be performed in SQL to update the data in the database.

The second is ViewFinalQueries, which has html code for showing to the user the created final queries, which may be replicated and performed in SQL.

4.2. Activity Diagram:

An activity diagram is essentially a flowchart that depicts the flow from one activity to another. The action can be defined as a system operation. The control flow is directed from one operation to the next. This flow might be sequential, branching, or concurrent in nature.

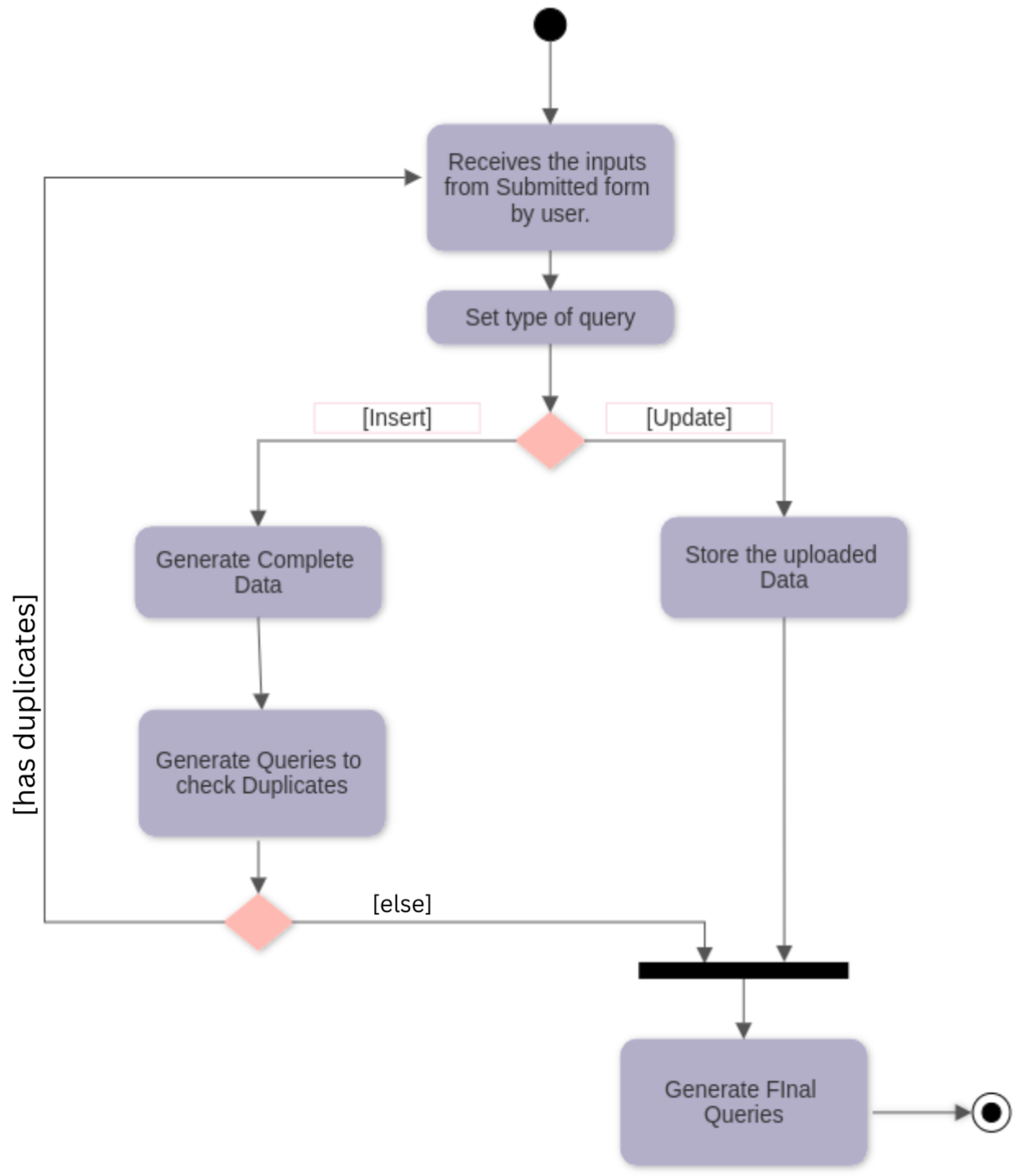


Figure 3.Activity Diagram

Figure 3 is an activity diagram that depicts the flow of each action in this application as well as the activities listed below.

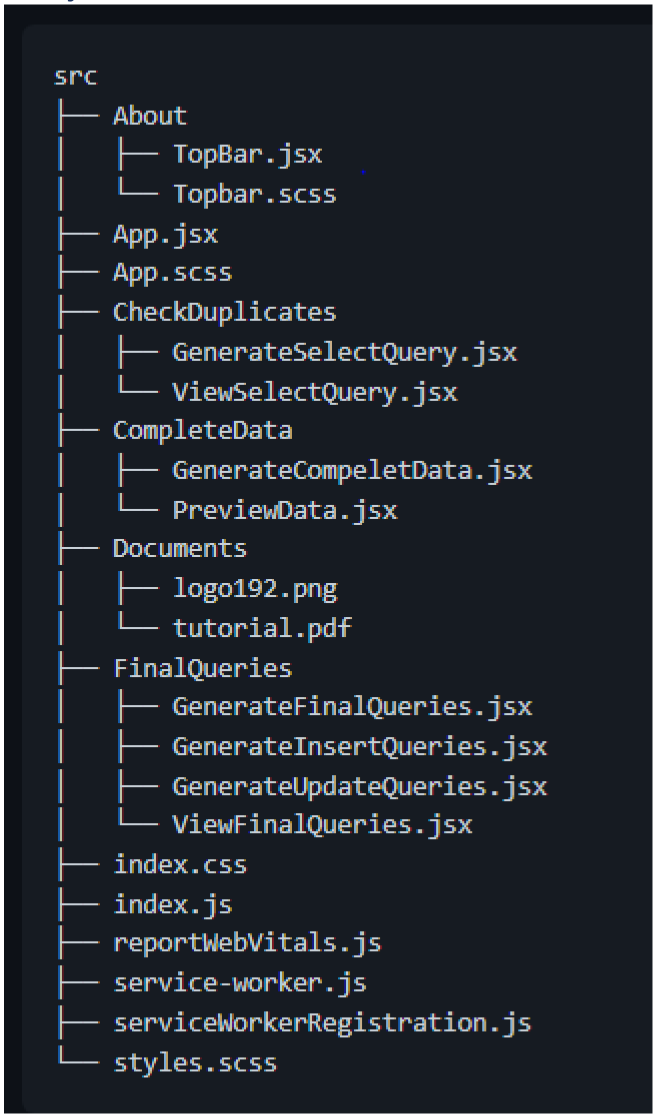
- First, the application receives user input data via the input form on the homepage.
- The data is then validated and assigned to the appropriate variables.
- If the query type is "insert," the application will generate complete data based on the excel data that is uploaded in the input form, and empty cells will be saved with the default values that are established in the background.
- Now it will build a select query that can be used to discover duplicates in the database with the same name and avoid duplication.
- If any duplicates are discovered, the user can return to the home page and resubmit the data after eliminating the duplicate item from the excel sheet.
- If there are no duplicates in the database, the user may click the "Generate Final Queries" button, which will generate final insert queries that can be copied and executed in SQL to add the entries to the database.
- If the query type is "update" the application will produce the necessary data depending on the excel data submitted.
- Because the user is altering already existing data in the database, there is no need to check for duplication.
- It will directly produce the final update queries that can be executed in SQL to update the entries in the database.

CHAPTER – 5

Development

In this chapter, I'll go through the specifics of what I've created so far. And provide some screenshots of the code and the result.

5.1. Structure of Repository:



```
src
├── About
│   ├── TopBar.jsx
│   └── Topbar.scss
├── App.jsx
├── App.scss
├── CheckDuplicates
│   ├── GenerateSelectQuery.jsx
│   └── ViewSelectQuery.jsx
├── CompleteData
│   ├── GenerateCompeletData.jsx
│   └── PreviewData.jsx
├── Documents
│   ├── logo192.png
│   └── tutorial.pdf
├── FinalQueries
│   ├── GenerateFinalQueries.jsx
│   ├── GenerateInsertQueries.jsx
│   ├── GenerateUpdateQueries.jsx
│   └── ViewFinalQueries.jsx
├── index.css
├── index.js
├── reportWebVitals.js
├── service-worker.js
├── serviceWorkerRegistration.js
└── styles.scss
```

The above image shows the structure of the application code repository.

src is the main folder where we maintain out coding files. All details about each file in this folder is provided below.

5.2. About each file in the repository:

5.2.1. index.jsx:

This is the index file, which is the first file in this repository that will be run. If we don't have this file, the execution will fail. Code is specified below.

This is the file used to interact across components, and we will import third-party libraries such as bootstrap and others here so that we may use them in any other component file without importing the library.

This file also specifies the routing between the components so that we can quickly route (go to) any of the pages after each step.

Source Code URL:

https://github.com/Harinathlee/upoint-query-builder/blob/master/src/index.js

5.2.2. App.jsx:

This file will have code for home page/landing page of this application where we will have an input for to collect the input data from user to manage the data and generate the queries.

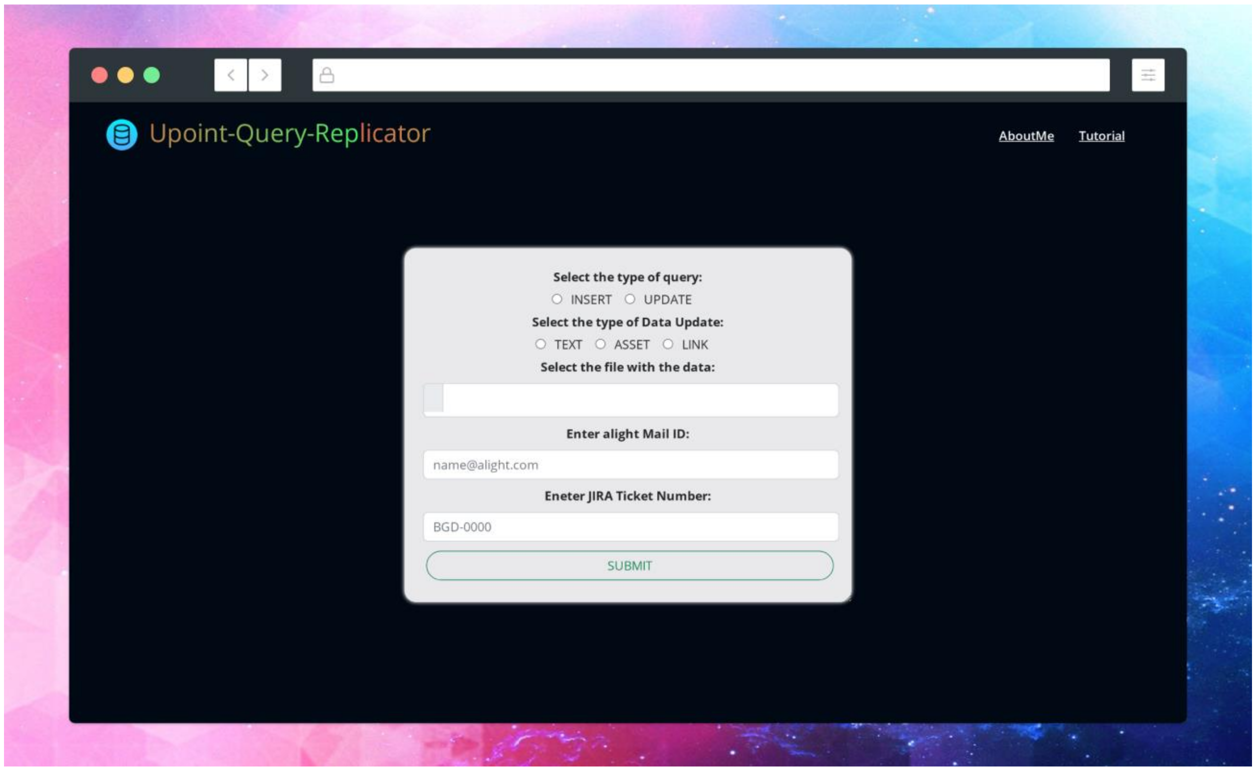
This file will have both HTML code of the page and JS code for background to do the validation of input data and assign the data to the variables in local storage of the web browser. So that we can access the same data where ever we want in this application code.

Few lines of code snippet is share in below images and also the screen shot of the output of this page in web browser.

Source code of App.jsx:

https://github.com/Harinathlee/upoint-query-builder/blob/master/src/App.jsx

Output Page of App.jsx Browser:



Select the type of query:
INSERT UPDATE
Select the type of Data Update:
TEXT ASSET LINK
Select the file with the data:
Enter alight Mail ID:
name@alight.com
Eneter JIRA Ticket Number:
BGD-0000
SUBMIT

5.2.3. App.scss and styles.scss:

These are the 2 files where all the CSS styles for the application exists. Like background, Font styles, positioning etc.

App.scss Code:

https://github.com/Harinathlee/upoint-query-builder/blob/master/src/App.scss

styles.scss Code:

https://github.com/Harinathlee/upoint-query-builder/blob/master/src/styles.scss

5.2.4. CompleteData/GenerateCompleteData.jsx:

This is the component that creates the whole data set depending on the user inputs and stores it in the Browser's local storage to make it accessible in any file at any time while the application is running. This file don't have any HTML code this is just JS code which runs in Background. And few lines of code is presented below.

<u>GenerateCompleteData.jsx Code:</u>

https://github.com/Harinathlee/upoint-query-builder/blob/master/src/CompleteData/GenerateCompeletData.jsx

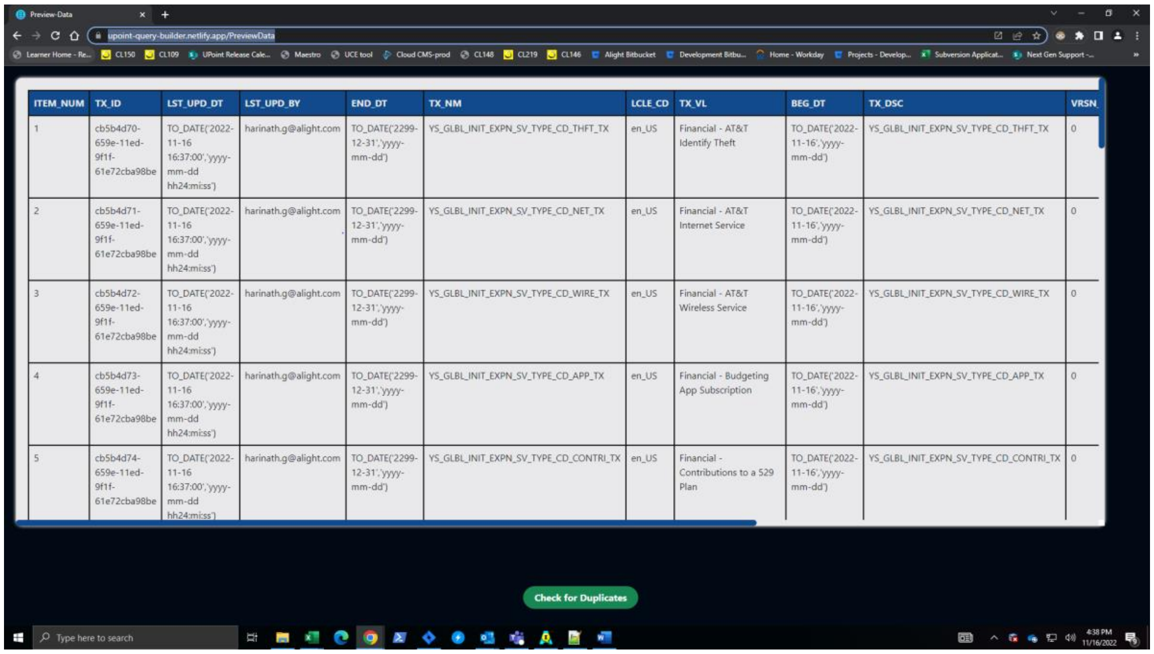
5.2.5. CompleteData/PreviewData.jsx:

This component file mostly contains HTML code for presenting the generated data in table format in a browser for the user to inspect and review. And includes some JS code to access the produced data from the browser's local storage. And it will have a button called CheckDuplicates which will route CheckDuplicates component.

<u>PreviewData.jsx Code:</u>

https://github.com/Harinathlee/upoint-query-builder/blob/master/src/CompleteData/PreviewData.jsx

<u>PreviewData.jsx Output:</u>



ITEM_NUM	TX_ID	LST_UPD_DT	LST_UPD_BY	END_DT	TX_NM	LCLE_CD	TX_VL	BEG_DT	TX_DSC	VRSN_
1	cb5b4d70-659e-11ed-9f1f-61e72cba98be	TO_DATE('2022-11-16 16:37:00','yyyy-mm-dd hh24:mi:ss')	harinath.g@alight.com	TO_DATE('2299-12-31','yyyy-mm-dd')	YS_GLBL_INIT_EXPN_SV_TYPE_CD_THFT_TX	en_US	Financial - AT&T Identify Theft	TO_DATE('2022-11-16','yyyy-mm-dd')	YS_GLBL_INIT_EXPN_SV_TYPE_CD_THFT_TX	0
2	cb5b4d71-659e-11ed-9f1f-61e72cba98be	TO_DATE('2022-11-16 16:37:00','yyyy-mm-dd hh24:mi:ss')	harinath.g@alight.com	TO_DATE('2299-12-31','yyyy-mm-dd')	YS_GLBL_INIT_EXPN_SV_TYPE_CD_NET_TX	en_US	Financial - AT&T Internet Service	TO_DATE('2022-11-16','yyyy-mm-dd')	YS_GLBL_INIT_EXPN_SV_TYPE_CD_NET_TX	0
3	cb5b4d72-659e-11ed-9f1f-61e72cba98be	TO_DATE('2022-11-16 16:37:00','yyyy-mm-dd hh24:mi:ss')	harinath.g@alight.com	TO_DATE('2299-12-31','yyyy-mm-dd')	YS_GLBL_INIT_EXPN_SV_TYPE_CD_WIRE_TX	en_US	Financial - AT&T Wireless Service	TO_DATE('2022-11-16','yyyy-mm-dd')	YS_GLBL_INIT_EXPN_SV_TYPE_CD_WIRE_TX	0
4	cb5b4d73-659e-11ed-9f1f-61e72cba98be	TO_DATE('2022-11-16 16:37:00','yyyy-mm-dd hh24:mi:ss')	harinath.g@alight.com	TO_DATE('2299-12-31','yyyy-mm-dd')	YS_GLBL_INIT_EXPN_SV_TYPE_CD_APP_TX	en_US	Financial - Budgeting App Subscription	TO_DATE('2022-11-16','yyyy-mm-dd')	YS_GLBL_INIT_EXPN_SV_TYPE_CD_APP_TX	0
5	cb5b4d74-659e-11ed-9f1f-61e72cba98be	TO_DATE('2022-11-16 16:37:00','yyyy-mm-dd hh24:mi:ss')	harinath.g@alight.com	TO_DATE('2299-12-31','yyyy-mm-dd')	YS_GLBL_INIT_EXPN_SV_TYPE_CD_CONTRI_TX	en_US	Financial - Contributions to a 529 Plan	TO_DATE('2022-11-16','yyyy-mm-dd')	YS_GLBL_INIT_EXPN_SV_TYPE_CD_CONTRI_TX	0

5.2.6. CheckDuplicates/GenerateSelectQuery.jsx:

This component file creates a SQL select query to see whether there are any other objects in the database with the same names. And stores the select query in local storage of browser to access it in any other component.

GenerateSelectQuery.jsx Code:

https://github.com/Harinathlee/upoint-query-builder/blob/master/src/CheckDuplicates/GenerateSelectQuery.jsx

5.2.7. CheckDuplicates/ViewSelectQuery.jsx:

This component file contains HTML code that displays the produced select query on the browser for the user to examine and allows the user to copy the query to run in SQL to check for duplication.

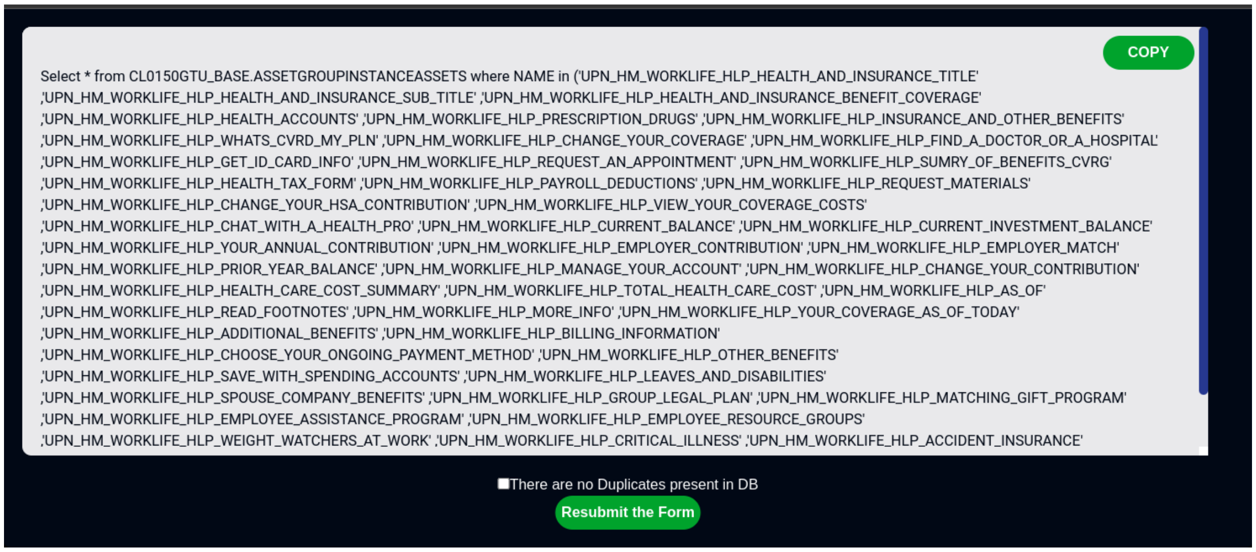
There will be a single button at the bottom of the page, and if we find any duplication, we will be sent to the home page to resubmit fresh data. Otherwise, we may use the same button to navigate to the FianlQueries component and construct final queries.

ViewSelectQuery.jsx Code:

https://github.com/Harinathlee/upoint-query-builder/blob/master/src/CheckDuplicates/ViewSelectQuery.jsx

ViewSelectQuery.jsx Output:

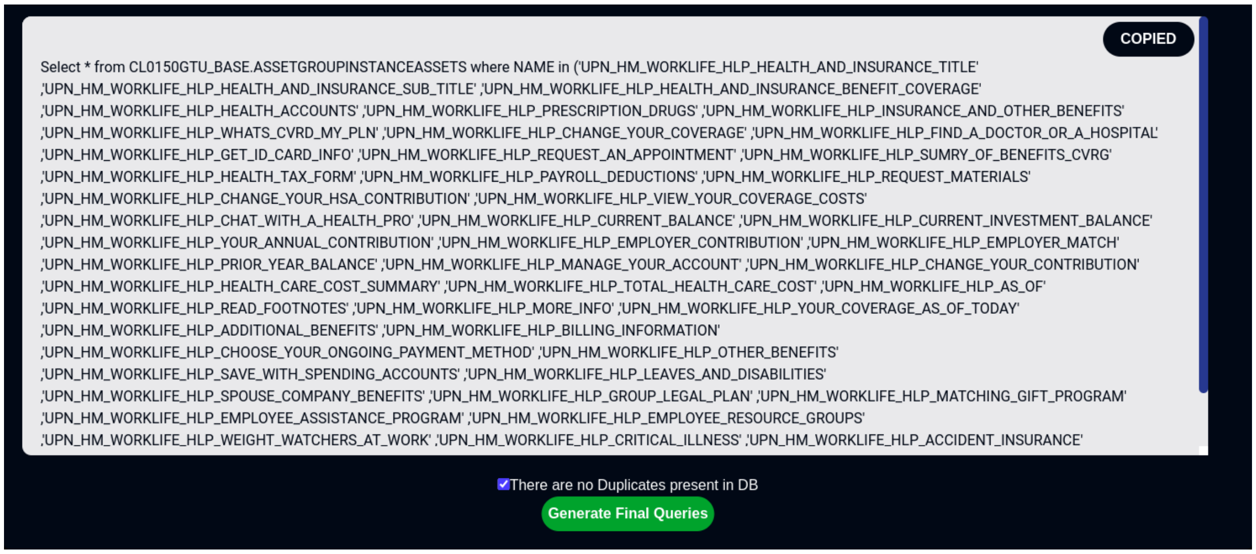
If there are duplicates:



```
Select * from CL0150GTU_BASE.ASSETGROUPINSTANCEASSETS where NAME in ('UPN_HM_WORKLIFE_HLP_HEALTH_AND_INSURANCE_TITLE'
,'UPN_HM_WORKLIFE_HLP_HEALTH_AND_INSURANCE_SUB_TITLE' ,'UPN_HM_WORKLIFE_HLP_HEALTH_AND_INSURANCE_BENEFIT_COVERAGE'
,'UPN_HM_WORKLIFE_HLP_HEALTH_ACCOUNTS' ,'UPN_HM_WORKLIFE_HLP_PRESCRIPTION_DRUGS' ,'UPN_HM_WORKLIFE_HLP_INSURANCE_AND_OTHER_BENEFITS'
,'UPN_HM_WORKLIFE_HLP_WHATS_CVRD_MY_PLN' ,'UPN_HM_WORKLIFE_HLP_CHANGE_YOUR_COVERAGE' ,'UPN_HM_WORKLIFE_HLP_FIND_A_DOCTOR_OR_A_HOSPITAL'
,'UPN_HM_WORKLIFE_HLP_GET_ID_CARD_INFO' ,'UPN_HM_WORKLIFE_HLP_REQUEST_AN_APPOINTMENT' ,'UPN_HM_WORKLIFE_HLP_SUMRY_OF_BENEFITS_CVRG'
,'UPN_HM_WORKLIFE_HLP_HEALTH_TAX_FORM' ,'UPN_HM_WORKLIFE_HLP_PAYROLL_DEDUCTIONS' ,'UPN_HM_WORKLIFE_HLP_REQUEST_MATERIALS'
,'UPN_HM_WORKLIFE_HLP_CHANGE_YOUR_HSA_CONTRIBUTION' ,'UPN_HM_WORKLIFE_HLP_VIEW_YOUR_COVERAGE_COSTS'
,'UPN_HM_WORKLIFE_HLP_CHAT_WITH_A_HEALTH_PRO' ,'UPN_HM_WORKLIFE_HLP_CURRENT_BALANCE' ,'UPN_HM_WORKLIFE_HLP_CURRENT_INVESTMENT_BALANCE'
,'UPN_HM_WORKLIFE_HLP_YOUR_ANNUAL_CONTRIBUTION' ,'UPN_HM_WORKLIFE_HLP_EMPLOYER_CONTRIBUTION' ,'UPN_HM_WORKLIFE_HLP_EMPLOYER_MATCH'
,'UPN_HM_WORKLIFE_HLP_PRIOR_YEAR_BALANCE' ,'UPN_HM_WORKLIFE_HLP_MANAGE_YOUR_ACCOUNT' ,'UPN_HM_WORKLIFE_HLP_CHANGE_YOUR_CONTRIBUTION'
,'UPN_HM_WORKLIFE_HLP_HEALTH_CARE_COST_SUMMARY' ,'UPN_HM_WORKLIFE_HLP_TOTAL_HEALTH_CARE_COST' ,'UPN_HM_WORKLIFE_HLP_AS_OF'
,'UPN_HM_WORKLIFE_HLP_READ_FOOTNOTES' ,'UPN_HM_WORKLIFE_HLP_MORE_INFO' ,'UPN_HM_WORKLIFE_HLP_YOUR_COVERAGE_AS_OF_TODAY'
,'UPN_HM_WORKLIFE_HLP_ADDITIONAL_BENEFITS' ,'UPN_HM_WORKLIFE_HLP_BILLING_INFORMATION'
,'UPN_HM_WORKLIFE_HLP_CHOOSE_YOUR_ONGOING_PAYMENT_METHOD' ,'UPN_HM_WORKLIFE_HLP_OTHER_BENEFITS'
,'UPN_HM_WORKLIFE_HLP_SAVE_WITH_SPENDING_ACCOUNTS' ,'UPN_HM_WORKLIFE_HLP_LEAVES_AND_DISABILITIES'
,'UPN_HM_WORKLIFE_HLP_SPOUSE_COMPANY_BENEFITS' ,'UPN_HM_WORKLIFE_HLP_GROUP_LEGAL_PLAN' ,'UPN_HM_WORKLIFE_HLP_MATCHING_GIFT_PROGRAM'
,'UPN_HM_WORKLIFE_HLP_EMPLOYEE_ASSISTANCE_PROGRAM' ,'UPN_HM_WORKLIFE_HLP_EMPLOYEE_RESOURCE_GROUPS'
,'UPN_HM_WORKLIFE_HLP_WEIGHT_WATCHERS_AT_WORK' ,'UPN_HM_WORKLIFE_HLP_CRITICAL_ILLNESS' ,'UPN_HM_WORKLIFE_HLP_ACCIDENT_INSURANCE'
```

☐ There are no Duplicates present in DB

If there are no duplicates:



```
Select * from CL0150GTU_BASE.ASSETGROUPINSTANCEASSETS where NAME in ('UPN_HM_WORKLIFE_HLP_HEALTH_AND_INSURANCE_TITLE'
,'UPN_HM_WORKLIFE_HLP_HEALTH_AND_INSURANCE_SUB_TITLE' ,'UPN_HM_WORKLIFE_HLP_HEALTH_AND_INSURANCE_BENEFIT_COVERAGE'
,'UPN_HM_WORKLIFE_HLP_HEALTH_ACCOUNTS' ,'UPN_HM_WORKLIFE_HLP_PRESCRIPTION_DRUGS' ,'UPN_HM_WORKLIFE_HLP_INSURANCE_AND_OTHER_BENEFITS'
,'UPN_HM_WORKLIFE_HLP_WHATS_CVRD_MY_PLN' ,'UPN_HM_WORKLIFE_HLP_CHANGE_YOUR_COVERAGE' ,'UPN_HM_WORKLIFE_HLP_FIND_A_DOCTOR_OR_A_HOSPITAL'
,'UPN_HM_WORKLIFE_HLP_GET_ID_CARD_INFO' ,'UPN_HM_WORKLIFE_HLP_REQUEST_AN_APPOINTMENT' ,'UPN_HM_WORKLIFE_HLP_SUMRY_OF_BENEFITS_CVRG'
,'UPN_HM_WORKLIFE_HLP_HEALTH_TAX_FORM' ,'UPN_HM_WORKLIFE_HLP_PAYROLL_DEDUCTIONS' ,'UPN_HM_WORKLIFE_HLP_REQUEST_MATERIALS'
,'UPN_HM_WORKLIFE_HLP_CHANGE_YOUR_HSA_CONTRIBUTION' ,'UPN_HM_WORKLIFE_HLP_VIEW_YOUR_COVERAGE_COSTS'
,'UPN_HM_WORKLIFE_HLP_CHAT_WITH_A_HEALTH_PRO' ,'UPN_HM_WORKLIFE_HLP_CURRENT_BALANCE' ,'UPN_HM_WORKLIFE_HLP_CURRENT_INVESTMENT_BALANCE'
,'UPN_HM_WORKLIFE_HLP_YOUR_ANNUAL_CONTRIBUTION' ,'UPN_HM_WORKLIFE_HLP_EMPLOYER_CONTRIBUTION' ,'UPN_HM_WORKLIFE_HLP_EMPLOYER_MATCH'
,'UPN_HM_WORKLIFE_HLP_PRIOR_YEAR_BALANCE' ,'UPN_HM_WORKLIFE_HLP_MANAGE_YOUR_ACCOUNT' ,'UPN_HM_WORKLIFE_HLP_CHANGE_YOUR_CONTRIBUTION'
,'UPN_HM_WORKLIFE_HLP_HEALTH_CARE_COST_SUMMARY' ,'UPN_HM_WORKLIFE_HLP_TOTAL_HEALTH_CARE_COST' ,'UPN_HM_WORKLIFE_HLP_AS_OF'
,'UPN_HM_WORKLIFE_HLP_READ_FOOTNOTES' ,'UPN_HM_WORKLIFE_HLP_MORE_INFO' ,'UPN_HM_WORKLIFE_HLP_YOUR_COVERAGE_AS_OF_TODAY'
,'UPN_HM_WORKLIFE_HLP_ADDITIONAL_BENEFITS' ,'UPN_HM_WORKLIFE_HLP_BILLING_INFORMATION'
,'UPN_HM_WORKLIFE_HLP_CHOOSE_YOUR_ONGOING_PAYMENT_METHOD' ,'UPN_HM_WORKLIFE_HLP_OTHER_BENEFITS'
,'UPN_HM_WORKLIFE_HLP_SAVE_WITH_SPENDING_ACCOUNTS' ,'UPN_HM_WORKLIFE_HLP_LEAVES_AND_DISABILITIES'
,'UPN_HM_WORKLIFE_HLP_SPOUSE_COMPANY_BENEFITS' ,'UPN_HM_WORKLIFE_HLP_GROUP_LEGAL_PLAN' ,'UPN_HM_WORKLIFE_HLP_MATCHING_GIFT_PROGRAM'
,'UPN_HM_WORKLIFE_HLP_EMPLOYEE_ASSISTANCE_PROGRAM' ,'UPN_HM_WORKLIFE_HLP_EMPLOYEE_RESOURCE_GROUPS'
,'UPN_HM_WORKLIFE_HLP_WEIGHT_WATCHERS_AT_WORK' ,'UPN_HM_WORKLIFE_HLP_CRITICAL_ILLNESS' ,'UPN_HM_WORKLIFE_HLP_ACCIDENT_INSURANCE'
```

✓ There are no Duplicates present in DB

Generate Final Queries

5.2.8. FinalQueries/GenerateFinalQueries.jsx:

This component file contains the JS code to generate the final insert/update queries by using the complete generated data an it calls the insert or update functions based on the type we selected.

GenerateFinalQueries.jsx Code:

https://github.com/Harinathlee/upoint-query-builder/blob/master/src/FinalQueries/GenerateFinalQueries.jsx

5.2.9. FinalQueries/ GenerateInsertQueries.jsx:

This component file contains the JS code to generate the final insert queries by using the complete generated data.

GenerateInsertQueries.jsx Code:

https://github.com/Harinathlee/upoint-query-builder/blob/master/src/FinalQueries/GenerateInsertQueries.jsx

5.2.10. FinalQueries/ GenerateUpdateQueries.jsx:

This component file contains the JS code to generate the final update queries by using the complete generated data.

GenerateFinalQueries.jsx Code:

https://github.com/Harinathlee/upoint-query-builder/blob/master/src/FinalQueries/GenerateUpdateQueries.jsx

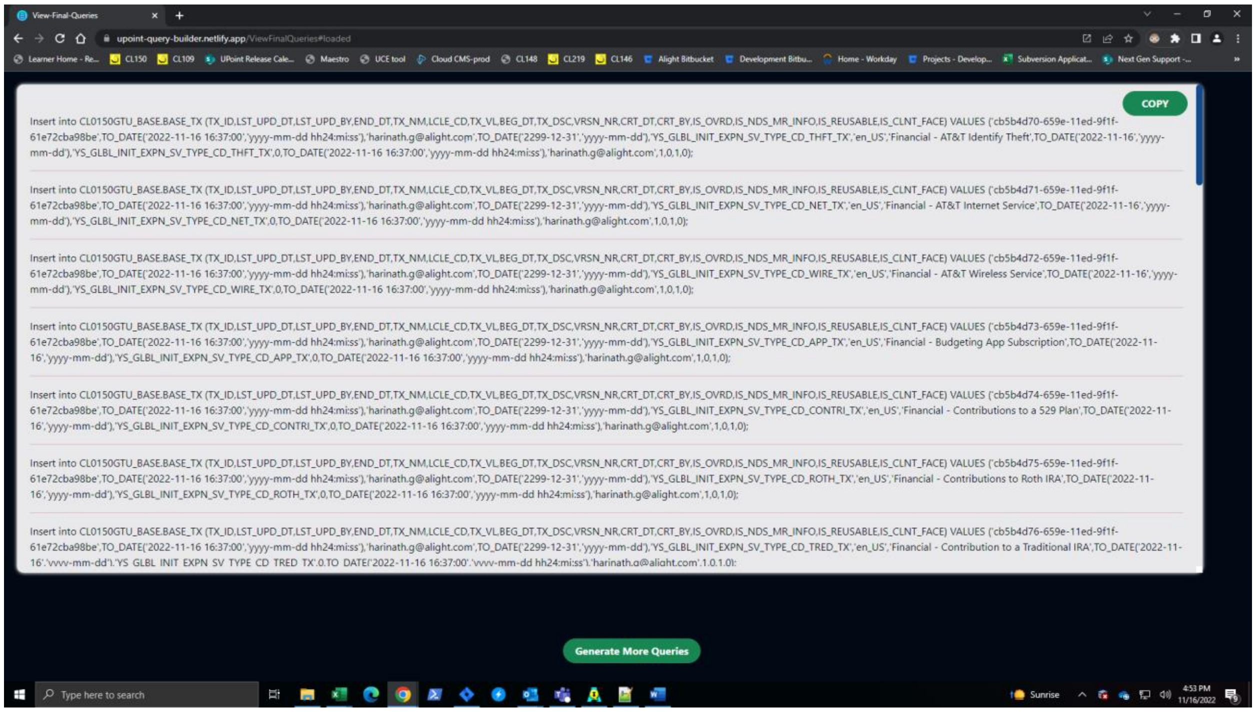
5.2.11. FinalQueries/ViewFinalQueries.jsx:

This component file contains HTML code that displays the produced final insert/update queries on the browser for the user to examine and allows the user to copy the query to run in SQL to update the data in DB. There will be a button to copy the queries with single click.

ViewFinalQueries.jsx Code:

https://github.com/Harinathlee/upoint-query-builder/blob/master/src/FinalQueries/ViewFinalQueries.jsx

ViewFinalQueries.jsx Output:



```
Insert into CL0150GTU_BASE.BASE_TX (TX_ID,LST_UPD_DT,LST_UPD_BY,END_DT,TX_NM,LCLE_CD,TX_VL,BEG_DT,TX_DSC,VRSN_NR,CRT_DT,CRT_BY,IS_OVRD,IS_NDS_MR_INFO,IS_REUSABLE,IS_CLNT_FACE) VALUES ('cb5b4d70-659e-11ed-9f1f-61e72cba98be',TO_DATE('2022-11-16 16:37:00','yyyy-mm-dd hh24:mi:ss'),'harinath.g@alight.com',TO_DATE('2299-12-31','yyyy-mm-dd'),'YS_GLBL_INIT_EXPN_SV_TYPE_CD_THFT_TX','en_US','Financial - AT&T Identify Theft',TO_DATE('2022-11-16','yyyy-mm-dd'),'YS_GLBL_INIT_EXPN_SV_TYPE_CD_THFT_TX',0,TO_DATE('2022-11-16 16:37:00','yyyy-mm-dd hh24:mi:ss'),'harinath.g@alight.com',1,0,1,0);

Insert into CL0150GTU_BASE.BASE_TX (TX_ID,LST_UPD_DT,LST_UPD_BY,END_DT,TX_NM,LCLE_CD,TX_VL,BEG_DT,TX_DSC,VRSN_NR,CRT_DT,CRT_BY,IS_OVRD,IS_NDS_MR_INFO,IS_REUSABLE,IS_CLNT_FACE) VALUES ('cb5b4d71-659e-11ed-9f1f-61e72cba98be',TO_DATE('2022-11-16 16:37:00','yyyy-mm-dd hh24:mi:ss'),'harinath.g@alight.com',TO_DATE('2299-12-31','yyyy-mm-dd'),'YS_GLBL_INIT_EXPN_SV_TYPE_CD_NET_TX','en_US','Financial - AT&T Internet Service',TO_DATE('2022-11-16','yyyy-mm-dd'),'YS_GLBL_INIT_EXPN_SV_TYPE_CD_NET_TX',0,TO_DATE('2022-11-16 16:37:00','yyyy-mm-dd hh24:mi:ss'),'harinath.g@alight.com',1,0,1,0);

Insert into CL0150GTU_BASE.BASE_TX (TX_ID,LST_UPD_DT,LST_UPD_BY,END_DT,TX_NM,LCLE_CD,TX_VL,BEG_DT,TX_DSC,VRSN_NR,CRT_DT,CRT_BY,IS_OVRD,IS_NDS_MR_INFO,IS_REUSABLE,IS_CLNT_FACE) VALUES ('cb5b4d72-659e-11ed-9f1f-61e72cba98be',TO_DATE('2022-11-16 16:37:00','yyyy-mm-dd hh24:mi:ss'),'harinath.g@alight.com',TO_DATE('2299-12-31','yyyy-mm-dd'),'YS_GLBL_INIT_EXPN_SV_TYPE_CD_WIRE_TX','en_US','Financial - AT&T Wireless Service',TO_DATE('2022-11-16','yyyy-mm-dd'),'YS_GLBL_INIT_EXPN_SV_TYPE_CD_WIRE_TX',0,TO_DATE('2022-11-16 16:37:00','yyyy-mm-dd hh24:mi:ss'),'harinath.g@alight.com',1,0,1,0);

Insert into CL0150GTU_BASE.BASE_TX (TX_ID,LST_UPD_DT,LST_UPD_BY,END_DT,TX_NM,LCLE_CD,TX_VL,BEG_DT,TX_DSC,VRSN_NR,CRT_DT,CRT_BY,IS_OVRD,IS_NDS_MR_INFO,IS_REUSABLE,IS_CLNT_FACE) VALUES ('cb5b4d73-659e-11ed-9f1f-61e72cba98be',TO_DATE('2022-11-16 16:37:00','yyyy-mm-dd hh24:mi:ss'),'harinath.g@alight.com',TO_DATE('2299-12-31','yyyy-mm-dd'),'YS_GLBL_INIT_EXPN_SV_TYPE_CD_APP_TX','en_US','Financial - Budgeting App Subscription',TO_DATE('2022-11-16','yyyy-mm-dd'),'YS_GLBL_INIT_EXPN_SV_TYPE_CD_APP_TX',0,TO_DATE('2022-11-16 16:37:00','yyyy-mm-dd hh24:mi:ss'),'harinath.g@alight.com',1,0,1,0);

Insert into CL0150GTU_BASE.BASE_TX (TX_ID,LST_UPD_DT,LST_UPD_BY,END_DT,TX_NM,LCLE_CD,TX_VL,BEG_DT,TX_DSC,VRSN_NR,CRT_DT,CRT_BY,IS_OVRD,IS_NDS_MR_INFO,IS_REUSABLE,IS_CLNT_FACE) VALUES ('cb5b4d74-659e-11ed-9f1f-61e72cba98be',TO_DATE('2022-11-16 16:37:00','yyyy-mm-dd hh24:mi:ss'),'harinath.g@alight.com',TO_DATE('2299-12-31','yyyy-mm-dd'),'YS_GLBL_INIT_EXPN_SV_TYPE_CD_CONTRI_TX','en_US','Financial - Contributions to a 529 Plan',TO_DATE('2022-11-16','yyyy-mm-dd'),'YS_GLBL_INIT_EXPN_SV_TYPE_CD_CONTRI_TX',0,TO_DATE('2022-11-16 16:37:00','yyyy-mm-dd hh24:mi:ss'),'harinath.g@alight.com',1,0,1,0);

Insert into CL0150GTU_BASE.BASE_TX (TX_ID,LST_UPD_DT,LST_UPD_BY,END_DT,TX_NM,LCLE_CD,TX_VL,BEG_DT,TX_DSC,VRSN_NR,CRT_DT,CRT_BY,IS_OVRD,IS_NDS_MR_INFO,IS_REUSABLE,IS_CLNT_FACE) VALUES ('cb5b4d75-659e-11ed-9f1f-61e72cba98be',TO_DATE('2022-11-16 16:37:00','yyyy-mm-dd hh24:mi:ss'),'harinath.g@alight.com',TO_DATE('2299-12-31','yyyy-mm-dd'),'YS_GLBL_INIT_EXPN_SV_TYPE_CD_ROTH_TX','en_US','Financial - Contributions to Roth IRA',TO_DATE('2022-11-16','yyyy-mm-dd'),'YS_GLBL_INIT_EXPN_SV_TYPE_CD_ROTH_TX',0,TO_DATE('2022-11-16 16:37:00','yyyy-mm-dd hh24:mi:ss'),'harinath.g@alight.com',1,0,1,0);

Insert into CL0150GTU_BASE.BASE_TX (TX_ID,LST_UPD_DT,LST_UPD_BY,END_DT,TX_NM,LCLE_CD,TX_VL,BEG_DT,TX_DSC,VRSN_NR,CRT_DT,CRT_BY,IS_OVRD,IS_NDS_MR_INFO,IS_REUSABLE,IS_CLNT_FACE) VALUES ('cb5b4d76-659e-11ed-9f1f-61e72cba98be',TO_DATE('2022-11-16 16:37:00','yyyy-mm-dd hh24:mi:ss'),'harinath.g@alight.com',TO_DATE('2299-12-31','yyyy-mm-dd'),'YS_GLBL_INIT_EXPN_SV_TYPE_CD_TRED_TX','en_US','Financial - Contribution to a Traditional IRA',TO_DATE('2022-11-16','yyyy-mm-dd'),'YS_GLBL_INIT_EXPN_SV_TYPE_CD_TRED_TX',0,TO_DATE('2022-11-16 16:37:00','yyyy-mm-dd hh24:mi:ss'),'harinath.g@alight.com',1,0,1,0);
```

CHAPTER – 6

Working procedure of the application

To launch the tool/application click on this link https://upoint-query-builder.netlify.app/ and proceed with steps listed below to work with the application.

- On launching the application, it will show the homepage with a top bar which has name of the application and links to about and tutorials pages and the input form to receive the inputs from user which is used to generate queries.
- User need to select the type of update we need to perform(insert/update) and then need to select what is the update(text/links/ assets).
- If ASSET is selected extra inputs are needed which will appear on form and need to provide the client email ID and the ticket number which is used to track down who and when the update is done.
- And mainly an excel sheet with the data needs to be submitted which is required to generate the SQL queries required.
- Every input value is required so the application won't allow us to submit until all the inputs are provide.
- After submitting the form data then it leads us to PreviewData page where it displays the whole data which is generated in the background JavaScript functions. So, after checking the data looks fine, then the user can click on the button below which will be "Check For Duplicates" if the type of update 'insert' or "Generate Final Queries" if type of update is 'update'.
- If it in insert update, then the button leads to ViewSelectQuery page where a select query is displayed to copy and run it in SQL to check whether the names of the items in data is already in use in Database so that we can rename/remove that item to prevent the duplication.
- This page will have a check box as "there are no duplicates in Database" makes button as "Generate Final Queries" if selected or makes it as "Resubmit the form" if not selected.
- Resubmit the form button redirect us to home page to resubmit the form with updated excel data.
- Generate Final Queries with direct us to the final page ViewFinalQueries where the final queries which is generated in background are displayed to check and copy the queries with just one click on copy button and then we can run the copied queries in SQL to insert/update the data in Data Base.
- This page also has a button as "Generate More Queries" which will redirect us to home page to submit more data to submit generate more queries.

Conclusion

All of the features that are currently implemented in this application are listed here, along with future implementation plans. The implementations shown above were completed in the first iteration of my project.

To conclude, I have finished the first half of my dissertation project work, which totaled 8 weeks of effort. And future work plans are outlined below.

Recommendations/ Directions for future work

In the future, I want to integrate further functionality as described below, and if necessary, additional functions that are not listed below may be implemented. It is entirely dependent on how the application functions and whether further features are required to make it even smarter.

- Need to implement the about and tutorial pages which have details about the project and the steps to work with the application.

Detailed Plan of Work:

Serial number of Task	Task or Subtasks to be done	Planned duration of weeks	Specific Deliverable in terms of the project
Mid Semester Scope: Completed			
1	Gathering Requirements	2	Gathered the requirements from my team and analyse those requirements.
2	Design	2	• High Level Architecture • Technical Design Document
3	Development	4	Developed few features like checking for duplicates and generate final queries in first iteration.
Final Semester Scope: Completed			
4	Development	4	Have plans to implement more features in future.
5	Impact & risk assessment-Testing	2	Test the code and evaluating the test results.
6	Deploying/hosting the Code	1	Hosting the developed website to make it accessible by anyone anywhere with the site address.
7	Completion of Dissertation Report	1	Final Report

Table 1. Detailed Plan of Work

Technologies Used:

Category	Names	Description
Languages	HTML	Markup language to develop Web pages.
	CSS	To style the web pages to look good.
	JavaScript	Run/handle the background operations.
	SQL	Language used to write queries to perform operations on Database.
Frame Works/ Libraries	Bootstrap	It's open-source and free to use yet features numerous HTML and CSS templates for UI interface elements such as buttons and forms.
	React JS	ReactJS is an open-source JavaScript library that is used for building user interfaces specifically for single-page applications.
	Sheet JS	It is an open-source solution for extracting useful data from almost any complex spreadsheet and generating new spreadsheets that will work with legacy and modern software alike.
	Google Fonts	To get the different font styles to make website looks prettier.
	UUID	Library to generate UUID for id field using react.
Tools	GitHub	GitHub is a code hosting platform for version control and collaboration.
	Vs Code	To write and run the code of project.
	Netlify	Tool used to host the tool on web.

Table 2. Technologies Used

Why React JS is used in this project?

- React JS code is simple to learn and implement; only a basic understanding of JavaScript is required.
- Implementing in react js has lesser code than using the regular JavaScript.
- This is the best JS library available to implement the best web applications.
- Reusing components is one of the key advantages of using React JS.
- Developers save time since they don't have to build several programs for the same functionalities. Additionally, if modifications are made to one area of the application, they won't have an impact elsewhere.
- Its goal is to make it simple for programmers to design quick user interfaces for both websites and applications. Virtual DOM is React.js's central idea.
- It helps to build rich user interfaces.
- It makes developers work more productive.
- It comes with a developer tool where we can use to debug the React components.
- It is easy to re render the pages other than entire application when particular values are changed.
- It has better code stability.
- And we have a wide range of community where people can help with our queries regarding React Js , which helps us to learn more and improve our skills.
- The React basically allows developers to utilize individual parts of their application on both client-side and the server-side, which ultimately boosts the speed of the development process.

Lighthouse Score for this application

Each website that a search engine crawls is given a score based on one of five criteria: performance, accessibility, best practices, SEO, and progressive web applications (PWA). A score between 0 and 100 is assigned to this. The higher your website appears on a search engine, the better your lighthouse score is.

Each of these categories are judged on the following:

Performance is judged on how quick it takes your webpage to load.

Accessibility is judged by how accessible your website is. Especially for users who might require technology such as a screen reader or have difficulty with colors.

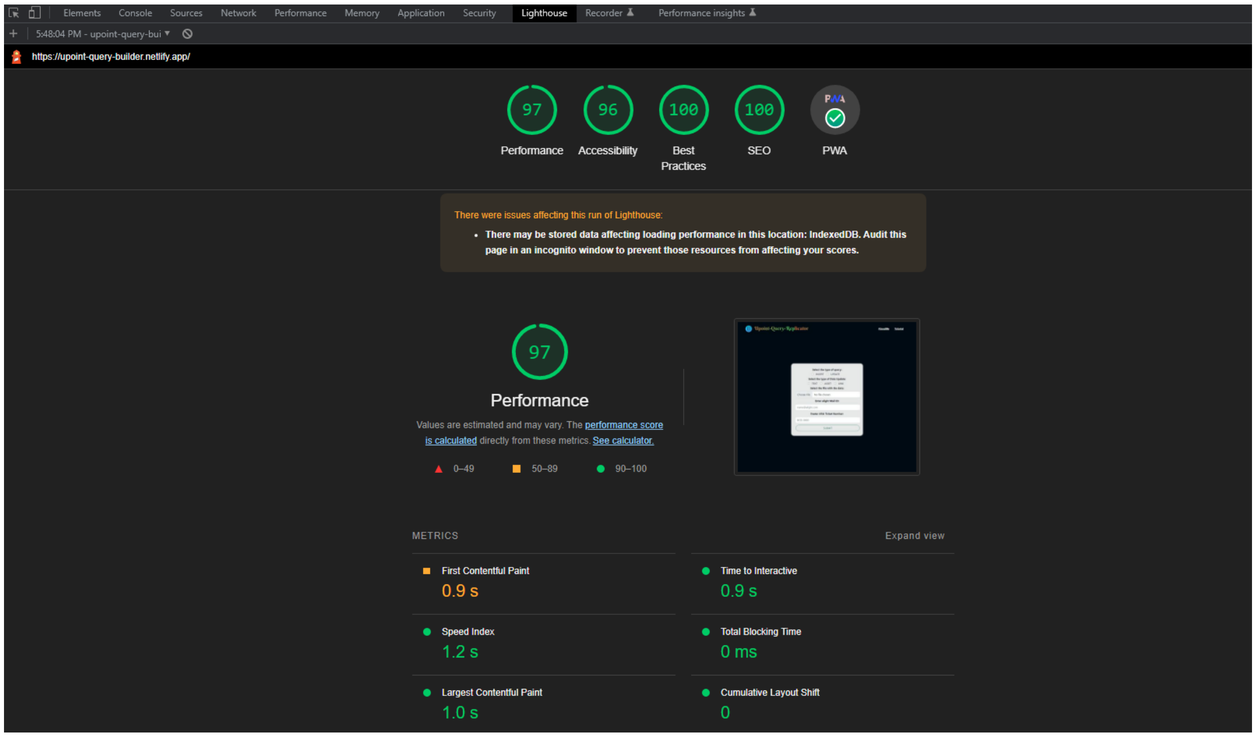
Best Practices are judged by factors which will usually only be apparent to developers. This will be on code health, for example, using deprecated Libraries/APIs, asking for permission if you want the users locations and making sure that it is a secure connection of HTTPS.

SEO (Search Engine Optimization) is judged by making sure the page is optimized for search engine results. This is a broad area of website design, but some straightforward examples include heading names, using keywords, and giving images names that are descriptive so search engines can recognize them.

PWA (Progressive Web Apps) does not receive a score, it is either there or not. This is still a rather early technology but makes websites run faster on repeated views.

With a good score, your website satisfies Google's SEO best practices and guidelines for performance and accessibility. Lighthouse is a crucial tool since it can spot typical issues that degrade the caliber of your websites and suggest fixes.

Below is the light house score for this tool:



https://upoint-query-builder.netlify.app/

97	96	100	100	✓
Performance	Accessibility	Best Practices	SEO	PWA

There were issues affecting this run of Lighthouse:

- There may be stored data affecting loading performance in this location: IndexedDB. Audit this page in an incognito window to prevent those resources from affecting your scores.

97

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

METRICS

■ First Contentful Paint 0.9 s	● Time to Interactive 0.9 s
● Speed Index 1.2 s	● Total Blocking Time 0 ms
● Largest Contentful Paint 1.0 s	● Cumulative Layout Shift 0

Reference and Bibliography

Below are the list of reference that I used for coding or preparing the document.

For Coding:

- https://stackoverflow.com/
- https://www.w3schools.com/
- https://www.geeksforgeeks.org/

For Document Preparation:

- https://www.tutorialspoint.com/
- https://www.geeksforgeeks.org/
- Software Engineering: A Practitioner's Approach (Roger S. Pressman) McGraw-Hill Science/Engineering/Math; 7th edition (November 1, 2010).

For Diagrams preparation:

- https://www.smartdraw.com/