

KrikeyAI Code Challenge Full Stack Engineer

Completed by Radhika Rajan Joshi

Part 1 SQL Challenge

Solution

1. Who are the first 10 authors ordered by date_of_birth?

Solution:

```
SELECT * FROM authors ORDER BY date_of_birth LIMIT 10;
```

2. What is the sales total for the author named "Lorelai Gilmore"?

Solution:

```
SELECT SUM(si.item_price * si.quantity) AS total_sales_revenue  
FROM sale_items si  
JOIN books b on si.book_id = b.id  
JOIN authors a ON b.author_id = a.id  
WHERE a.name = 'Lorelai Gilmore';
```

3. What are the top 10 performing authors, ranked by sales revenue?

Solution:

```
SELECT a.name, SUM(si.item_price * si.quantity) AS total_revenue  
FROM authors a  
JOIN books b on a.id = b.author_id  
JOIN sale_items si on b.id = si.book_id  
GROUP BY a.name  
ORDER BY total_revenue DESC  
LIMIT 10;
```

Note: Execution screenshots with dummy data for the PostgreSQL queries can be found at the end of this document for reference.

Part 2A and Part 2B

Code for implementation of Part 2A and 2B is present in folder

[KrikeyPart2middlewareAPI](#) with filename *apiServer.js*.

Part 3

The link for deployed React webpage: <https://radhikarj.github.io/krikeycc/>

Please see below for Execution Instructions.

Instructions to run Part 1, 2 and 3 locally:

Goal: React Web Pages uses the API endpoint in part 2 to retrieve author data and display on UI built with provided Figma specifications.

Part 1

For running apiServer.js, first run the PostgreSQL database, create the tables and insert data into the database.

To run apiServer.js, run a terminal at folder location [KrikeyPart2middlewareAPI](#) on your system which contains the API file **apiServer.js**.

Next, run the command:

node apiServer.js

The server will start to run at url and by using the below path, we can view the JSON response of the top 10 performing authors:

<http://localhost:3000/topAuthors>

To find the total revenue of particular author, such as Lorelai Gilmore, provide the author name as shown in the URL below:

http://localhost:3000/topAuthors?author_name=Lorelai Gilmore

Next, considering that NodeJS and ReactJS are installed, we open a terminal at the location where the **krikeycc** folder is cloned/downloaded. To run the react application locally, cd into the folder **krikeycc** folder, run the following command:

npm start

It will request permission to run on another port (since our middleware apiServer.js is running at localhost:3000), enter “y” or type “yes” and the react application will launch on localhost at as:

<http://localhost:3001/>

Here, the ReactJS code is fetching the data from the PostgreSQL database through the apiServer.js API endpoint running at localhost:3000 (server in Part 2) and displaying the list of all team members on the UI.

Part 3

For deploying the web page developed as a React Application, the deployed web page (post deployment on a third party platform such as GitHub Pages) uses static content for the team members from Part 2 for display.

Steps to deploy the React web page on GitHub pages is as follows:

1. Commit and push the react web application at a [GitHub Repository](#).
2. Update the **package.json** file to include the url of the repository and append the **predeploy** and **deploy scripts** as follows:

```
"homepage": "http://RadhikaRJ.github.io/krikeycc",
```

```
"scripts": {  
  "predeploy": "npm run build",  
  "deploy": "gh-pages -d build",  
}
```

3. Next, run the command on terminal of react web application:

```
npm install gh-pages --save-dev
```

4. Commit and push the updated package.json document
5. Create a gh-pages branch on Github repository
6. In the terminal of the web app, run the deploy command:

```
npm run build
```
7. Navigate to github repository, settings tab to get the URL of the deployed react web page.

Screenshots of PostgreSQL Database:

```

7dbs=# SELECT * FROM books;
 id | author_id | isbn
-----+-----+-----
 60 |         30 | JD001
 61 |         31 | JS001
 62 |         32 | AJ001
 63 |         33 | BW001
 64 |         34 | LG001
 65 |         35 | HP001
 66 |         36 | HG001
 67 |         37 | RW001
 68 |         38 | G001
 69 |         39 | FB001
 70 |         40 | BB001
 71 |         41 | A001
 72 |         42 | L001
 73 |         43 | GIM001
 74 |         44 | SG001
 75 |         45 | GOL001
 76 |         46 | SAU001
 77 |         47 | SAR001
 78 |         48 | GAL001
 79 |         49 | E001
(20 rows)

```

```

7dbs=# SELECT * FROM sale_items;
 id | book_id | customer_name | item_price | quantity
-----+-----+-----+-----+-----
  1 |      60 | Customer1     |   $10.00   |        1
  2 |      61 | Customer2     |   $20.00   |        2
  3 |      62 | Customer3     |   $30.00   |        1
  4 |      63 | Customer4     |   $40.00   |        2
  5 |      64 | Customer5     |   $50.00   |        1
  6 |      64 | Customer6     |   $60.00   |        2
  7 |      64 | Customer7     |   $70.00   |        1
  8 |      64 | Customer8     |   $80.00   |        2
  9 |      64 | Customer9     |   $90.00   |       10
 10 |      65 | Customer10    |  $100.00   |        2
 11 |      66 | Customer11    |  $110.00   |        1
 12 |      67 | Customer12    |  $120.00   |        2
 13 |      68 | Customer13    |  $130.00   |        1
 14 |      69 | Customer14    |  $140.00   |        2
 15 |      70 | Customer15    |  $150.00   |        1
 16 |      71 | Customer16    |  $160.00   |        6
 17 |      72 | Customer17    |  $170.00   |        1
 18 |      73 | Customer18    |  $180.00   |        5
 19 |      74 | Customer19    |  $190.00   |        1
 20 |      75 | Customer20    |  $200.00   |        2
 21 |      76 | Customer21    |  $210.00   |        4
 22 |      77 | Customer22    |  $220.00   |        3
 23 |      78 | Customer23    |  $230.00   |        1
 24 |      79 | Customer24    |  $240.00   |        2
(24 rows)

```

```
7dbs=# SELECT * FROM authors ORDER BY date_of_birth LIMIT 10;
```

id	name	email	date_of_birth
47	Saruman	saruman@example.com	1955-06-30 00:00:00
40	Bilbo Baggins	bilbo@example.com	1962-09-22 00:00:00
48	Galadriel	galadriel@example.com	1965-10-24 00:00:00
39	Frodo Baggins	frodo@example.com	1968-09-22 00:00:00
34	Lorelai Gilmore	lorelai@example.com	1968-11-07 00:00:00
43	Gimli	gimli@example.com	1970-12-12 00:00:00
41	Aragorn	aragorn@example.com	1975-03-01 00:00:00
31	Jane Smith	jane@example.com	1975-05-20 00:00:00
44	Samwise Gamgee	samwise@example.com	1978-04-04 00:00:00
37	Ron Weasley	ron@example.com	1979-03-01 00:00:00

(10 rows)

```
7dbs=# SELECT SUM(si.item_price * si.quantity) AS total_sales_revenue
FROM sale_items si
JOIN books b on si.book_id = b.id
JOIN authors a ON b.author_id = a.id
WHERE a.name = 'Lorelai Gilmore';
```

total_sales_revenue
\$1,300.00

(1 row)

```

7dbs=# SELECT a.name, SUM(si.item_price * si.quantity) AS total_revenue
FROM authors a
JOIN books b on a.id = b.author_id
JOIN sale_items si on b.id = si.book_id
GROUP BY a.name
ORDER BY total_revenue DESC
LIMIT 10;

```

name	total_revenue
Lorelai Gilmore	\$1,300.00
Aragorn	\$960.00
Gimli	\$900.00
Sauron	\$840.00
Saruman	\$660.00
Elrond	\$480.00
Gollum	\$400.00
Frodo Baggins	\$280.00
Ron Weasley	\$240.00
Galadriel	\$230.00

(10 rows)

```

7dbs=# SELECT * FROM authors;

```

id	name	email	date_of_birth
30	John Doe	john@example.com	1980-01-15 00:00:00
31	Jane Smith	jane@example.com	1975-05-20 00:00:00
32	Alice Johnson	alice@example.com	1990-08-10 00:00:00
33	Bob Williams	bob@example.com	1988-03-25 00:00:00
34	Lorelai Gilmore	lorelai@example.com	1968-11-07 00:00:00
35	Harry Potter	harry@example.com	1979-07-31 00:00:00
36	Hermione Granger	hermione@example.com	1979-09-19 00:00:00
37	Ron Weasley	ron@example.com	1979-03-01 00:00:00
38	Gandalf	gandalf@example.com	
39	Frodo Baggins	frodo@example.com	1968-09-22 00:00:00
40	Bilbo Baggins	bilbo@example.com	1962-09-22 00:00:00
41	Aragorn	aragorn@example.com	1975-03-01 00:00:00
42	Legolas	legolas@example.com	1982-08-17 00:00:00
43	Gimli	gimli@example.com	1970-12-12 00:00:00
44	Samwise Gamgee	samwise@example.com	1978-04-04 00:00:00
45	Gollum	gollum@example.com	
46	Sauron	sauron@example.com	
47	Saruman	saruman@example.com	1955-06-30 00:00:00
48	Galadriel	galadriel@example.com	1965-10-24 00:00:00
49	Elrond	elrond@example.com	3000-01-01 00:00:00

(20 rows)