

FAANG Source – Final Project Report

ITOM6265 – Database Design Fall Module B

Table of Contents

Introduction to The Organization.....	2
Scope of the Project.....	2-3
Goals of the Project.....	3
Database Design.....	3-4
Shiny User Interface.....	4
Appendix.....	5-20

Introduction to the Organization

Salary transparency is becoming an increasingly relevant topic as the effort to end workplace pay discrimination strengthens. With negotiations being a common component of the job search process, many wonder if they are being grossly underpaid or if other job opportunities could be more lucrative.

FAANG, an acronym for Facebook (Meta), Amazon, Apple, Netflix, and Google (Alphabet), are currently the 5 biggest players in the American technology industry, otherwise known as Big Tech. As positions (specifically, technical roles) within these companies become more sought after, the number of variables potential candidates must consider when beginning their job search, or doing job research grows as well. Our organization and application, FAANG Source, seeks to add a level of transparency to the Big Tech industry.

FAANG Source is an application that invites both potential candidates and current FAANG community members to participate in the building of a platform, separate from potential censorship, that will provide those interested with tools to gain valuable insights into industry compensation standards. Our back-end database contains real-time, user submitted entries which can be utilized by the end-user for research considerations, and in future negotiations. With a particular emphasis on data points such as company, position, years of experience, level of education, and—most importantly—salary, this application will be a trustworthy resource for job seeking candidates and internal employees. FAANG Source will help users put their best foot forward during the recruiting and application process, salary negotiations, and job selections.

Scope of the Project

Our database includes several metrics and categorical attributes that yield valuable information for job seekers in the areas of salaries, positions, and opportunities including:

- FAANG Company
- Position
- Highest level of education achieved
- Years of experience
- Office location
- Salary

Additionally, the application's analytical component drives insight on current trends in Big Tech salary compensation within specific demographics such as:

- FAANG Company
- Position
- Gender

- Age Range
- Education Level
- Years of Experience
- Office Location

The database currently contains a static set of technical and non-technical roles for each of the 5 FAANG companies. To start, our organization selected the 25 roles we felt were the most representative of our ideal end-user: job seekers or currently/previously employed FAANG community members, ages 20-40 years old with no more than 10 years of work experience. In future versions, we hope to massively expand our list of roles and employers by implementing user-submitted, admin-approved data entry capabilities. This will broaden our end-user archetype in the process.

Goals of the Project

The intended audience of users for this platform include those interested in working for a FAANG company, as well as those who are currently or have previously worked in the Big Tech industry. Potential unintended users could be journalists or researchers looking to report or study salary information from the most prominent American technology companies. The primary use of this platform is to gather information and insights regarding salary compensation standards across the FAANG community through the use of user-submitted data. The platform has been developed with both job seekers and current FAANG employees at the center of focus. Namely, a job seeker could log on and get a better understanding of what people with similar characteristics are being paid in Big Tech roles. On the other hand, current FAANG employees may be looking to make a situational change, or ensure that their salary is in line with their archetypal averages. The end goal for FAANG Source is to be the premier destination for honest, up-to-date, and accurate Big Tech compensation information. This platform will play a key role in salary negotiations for both groups of talent.

Database Design

The Entity Relationship Diagram (**Exhibit 1**), shows the 4 unique tables used on the back-end of the FAANG Source application. All 4 tables include information necessary for the goals listed above. The structure and design of our relational database system follows the third normal form in an effort to simplify data management, reduce the duplication of data entries, and avoid any data anomalies.

The final iteration of the ERD came about through conceptualizing the front-end user process our platform would require. At the current moment, all the data within our application has been synthetically aggregated, so it was imperative that each relational table contained data that aligned with a specific user-facing functional component. The first step was to categorize the aggregated data into 4 distinct table entities: *Company*, *Job*, *UserInfo*, and *UserInput*. **Exhibit 2**

explains the relationships between these entities. The *Company* and *Job* tables are both static entities, which exist as base reservoirs for FAANG company and role information. The *UserInfo* table, and the *UserInput* table—an associative entity—are both composed of information submitted by our end-users.

After that, each table was assigned an identity primary key column, which sequentially delineates each row within the table from the rest. All columns in each table contain information exclusive to that specific entity unless it is a foreign key column; these are used to relate a column back to its specific entity. For example, *company_id* is the primary key for the *Company* table, and a foreign key within the *Job* table. Including *company_id* as a foreign key within the *Job* table makes each *job_id* uniquely distinguishable. There may be multiple unique *job_id*'s that have the same *job_title*, but each will have a different *company_id* (**Exhibits 3, 4 A & B**).

Shiny User Interface

FAANG Source's front-end interface was built to be conducive to collecting new data; we wanted to make things for the user as simple and intuitive as we possibly could. With this goal in mind, the application opens to the 'Home' screen (**Exhibit 5 A**) which features a tab glossary. This shows all the platform features in the sequence we designed the user to follow.

The 'Create New User' and 'User Entry' tabs possess the components that collect all the user submitted data for the *UserInfo* and *UserInput* tables. In 'Create New User' (**Exhibit 5 B**), all platform users can create a unique user ID by submitting their first name, last name, email address, gender, and age. In 'User Entry' (**Exhibit 5 C**), users with FAANG work experience can enter their salary information from current or previous roles alongside characteristics such as: highest level of education achieved, years of experience, and office location.

Once finished creating a user ID and submitting a data entry, the user can move forward into the 'Search Database' and 'Salary Analytics' tabs. In 'Search Database' (**Exhibit 5 D**), a job seeking user can thumb through all available user entries to get a stronger grasp of salary levels at different Big Tech companies, positions, experience levels, educational levels, and office locations. The database is searchable with any combination of the given parameters, allowing the user to fully customize the output results. A current FAANG employee can use this tab to compare their salary with other users' entries by searching for a combination of their own characteristics. They could also do market research to see how closely their salary matches up to other's at different companies, or office locations. In 'Salary Analytics' (**Exhibit 5 E**), users can view visualizations displaying average salary ranks across 7 different demographics.

The final tab, 'User Settings' (**Exhibit 5 F**), is a hub where the user can find, view, and manage their entries and user information. Users can produce a table of all their submitted data entries, edit or delete previously submitted entries, and edit any wrongly inputted user information. Users can also retrieve a forgotten user ID by searching for their first and last name.

Appendix:

Exhibit 1: Entity Relationship Diagram (ERD)

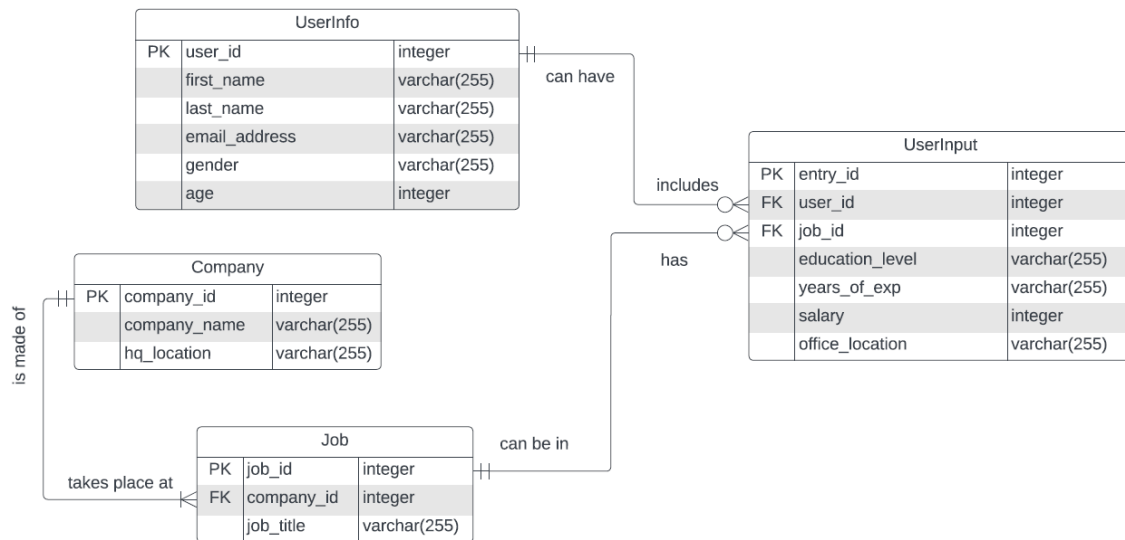


Exhibit 2: Relationships

1. A **user** can have zero to many **user_input**'s.
2. A **user_input** includes only one **user**.
3. A **company** is made of one to many **job**'s.
4. A **job** takes place at only one **company**.
5. A **job** can be in zero to many **user_input**'s.
6. A **user_input** has only one **job**.

Exhibit 3: Data Dictionary

- Company
 - company_id: **(PK)** Unique ID number for each company in the database
 - company: Name of FAANG company
 - hq_location: City and State of company's HQ
- Job
 - job_id: **(PK)** Unique ID number for each job in the database
 - company_id: **(FK)**
 - job_title: Title of position (25 available options)
- UserInfo
 - user_id: **(PK)** Unique ID number for each user in the database
 - first_name: User inputted first name
 - last_name: User inputted last name
 - email_address: User inputted email address
 - gender: User selection: Male, Female, or Unidentified
 - age: User inputted age
- UserInput
 - entry_id: **(PK)** Unique ID number for each entry in the database
 - user_id: **(FK)**
 - job_id: **(FK)**
 - education_level: User selection: High School Degree, Undergraduate Degree, or Postgraduate Degree
 - years_of_exp: User selection: 0-2 Years, 3-5 Years, or 6-10 Years
 - salary: User inputted salary
 - office_location: User selection: Austin, TX, New York, NY, or San Francisco, CA

Exhibit 4: Table List

A. Company Table (company_id values 1-5)

	company_id	company_name	hq_location
1	1	Facebook	Menlo Park, CA
2	2	Amazon	Cupertino, CA
3	3	Apple	Seattle, WA
4	4	Netflix	Los Gatos, CA
5	5	Google	Mountain View, CA

B. Job Table (job_id values 1-30)

	job_id	company_id	job_title
1	1	1	Data Scientist 1
2	2	1	Data Scientist 2
3	3	1	Data Scientist 3
4	4	1	Data Engineer 1
5	5	1	Data Engineer 2
6	6	1	Data Engineer 3
7	7	1	Data Analyst 1
8	8	1	Data Analyst 2
9	9	1	Data Analyst 3
10	10	1	Software Engi...
11	11	1	Software Engi...
12	12	1	Software Engi...
13	13	1	Network Engi...
14	14	1	Network Engi...
15	15	1	Network Engi...
16	16	1	Financial Anal...
17	17	1	Senior Financi...
18	18	1	Finance Mana...
19	19	1	Marketing Ass...
20	20	1	Senior Marketi...
21	21	1	Marketing Ma...
22	22	1	Program Mana...
23	23	1	Project Manager
24	24	1	Technical Bus...
25	25	1	Account Man...
26	26	2	Data Scientist 1
27	27	2	Data Scientist 2
28	28	2	Data Scientist 3
29	29	2	Data Engineer 1
30	30	2	Data Engineer 2

(Note: As described in the text above, job_id 1 & 26 both have a job_title of 'Data Scientist 1' but the foreign key, company_id, makes them distinguishable. With an included foreign key from the Company table, you can tell that job_id 1 is a Facebook position, and job_id 26 is an Amazon position.)

C. UserInfo Table (user_id values 1-30)

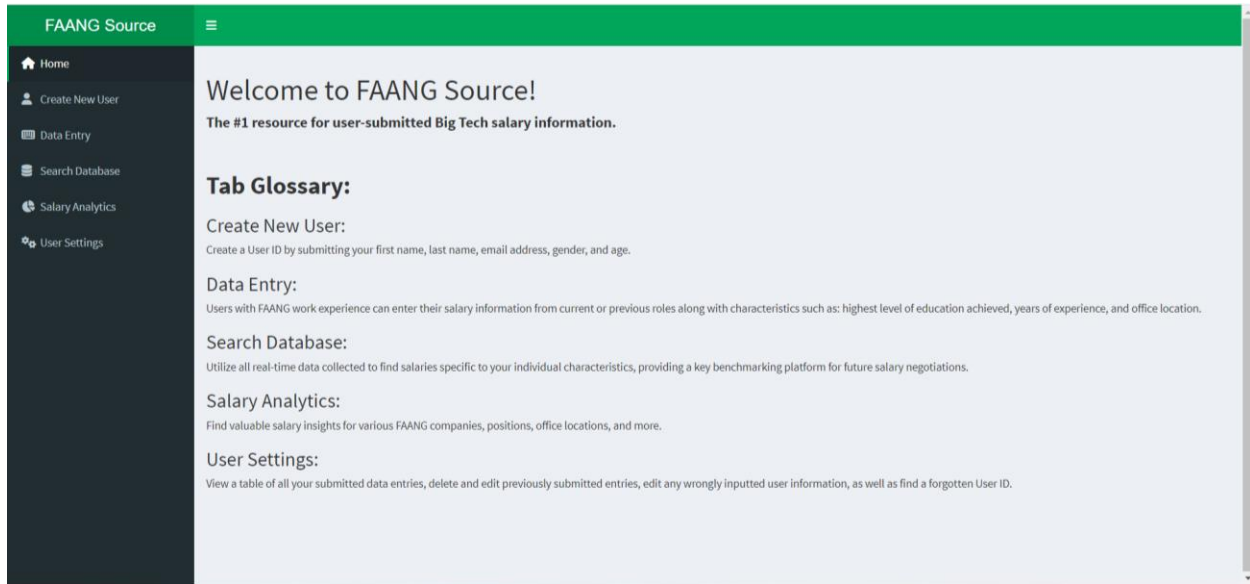
	user_id	first_name	last_name	email_address	gender	age
1	1	Malachi	Clement	malachi.clement85@aol.com	Male	34
2	2	Bernardo	Haskell	bernardo.haskell64@gmail.com	Male	29
3	3	Amari	Culver	amari.culver53@yahoo.com	Male	37
4	4	Lester	Shumaker	lester.shumaker56@yahoo.com	Male	25
5	5	Darin	Valerio	darin.valerio38@yahoo.com	Male	37
6	6	Rohan	Slack	rohan.slack44@gmail.com	Male	27
7	7	Chaim	Desantis	chaim.desantis21@yahoo.com	Male	40
8	8	Kolten	Craft	kolten.craft64@outlook.com	Male	25
9	9	Taylor	Samuel	taylor.samuel63@gmail.com	Male	33
10	10	Santos	Ashford	santos.ashford74@yahoo.com	Male	39
11	11	Harrison	Chamberlin	harrison.chamberlin86@aol.com	Male	22
12	12	Marcello	Vela	marcello.vela68@aol.com	Male	38
13	13	Kylan	Griffiths	kylan.griffiths36@yahoo.com	Male	37
14	14	Myles	Cady	myles.cady60@gmail.com	Male	38
15	15	Josiah	Thrasher	josiah.thrasher17@aol.com	Male	25
16	16	Jade	Guillory	jade.guillory37@outlook.com	Male	37
17	17	Loren	Petty	loren.petty51@outlook.com	Male	39
18	18	Augustine	Chung	augustine.chung12@gmail.com	Male	36
19	19	Bo	Webb	bo.webb58@outlook.com	Male	24
20	20	Leslie	Pace	leslie.pace70@yahoo.com	Male	32
21	21	Don	Gay	don.gay96@yahoo.com	Male	26
22	22	Karson	Fenton	karson.fenton71@outlook.com	Male	40
23	23	Kasey	Grove	kasey.grove37@aol.com	Male	40
24	24	Isaak	Hartley	isaak.hartley97@yahoo.com	Male	39
25	25	Tristian	Bynum	tristian.bynum68@aol.com	Male	23
26	26	Clifford	Strom	clifford.strom87@outlook.com	Male	40
27	27	Kobi	Dickinson	kobi.dickinson84@yahoo.com	Male	29
28	28	Bennett	Busch	bennett.busch15@yahoo.com	Male	30
29	29	Tobias	Jack	tobias.jack76@outlook.com	Male	32
30	30	Kristofer	Salter	kristofer.salter47@gmail.com	Male	21

D. UserInput Table (entry_id values 1-30)

	entry_id	user_id	job_id	education_level	years_of_exp	salary	office_location
1	1	1	65	Postgraduate Degree	6-10 Years	212500	Austin, TX
2	2	2	100	Undergraduate Degree	3-5 Years	197700	New York, NY
3	3	3	109	Postgraduate Degree	6-10 Years	221000	New York, NY
4	4	4	29	Undergraduate Degree	0-2 Years	173500	New York, NY
5	5	5	31	Undergraduate Degree	6-10 Years	206300	San Francis...
6	6	6	27	Undergraduate Degree	3-5 Years	191600	Austin, TX
7	7	7	103	Undergraduate Degree	6-10 Years	202900	San Francis...
8	8	8	29	High School Degree	0-2 Years	162700	New York, NY
9	9	9	55	Undergraduate Degree	3-5 Years	195600	San Francis...
10	10	10	3	Postgraduate Degree	6-10 Years	234400	San Francis...
11	11	11	110	High School Degree	0-2 Years	158800	Austin, TX
12	12	12	9	Undergraduate Degree	6-10 Years	239100	Austin, TX
13	13	13	24	Postgraduate Degree	6-10 Years	245400	New York, NY
14	14	14	9	Postgraduate Degree	6-10 Years	234400	New York, NY
15	15	15	60	Undergraduate Degree	0-2 Years	173600	San Francis...
16	16	16	115	Undergraduate Degree	6-10 Years	207500	New York, NY
17	17	17	87	Postgraduate Degree	6-10 Years	223800	New York, NY
18	18	18	111	High School Degree	3-5 Years	196300	New York, NY
19	19	19	94	Undergraduate Degree	0-2 Years	135500	San Francis...
20	20	20	22	Undergraduate Degree	3-5 Years	244100	New York, NY
21	21	21	61	High School Degree	3-5 Years	196100	New York, NY
22	22	22	53	Undergraduate Degree	6-10 Years	212400	San Francis...
23	23	23	93	Undergraduate Degree	6-10 Years	242300	San Francis...
24	24	24	53	Undergraduate Degree	6-10 Years	209500	San Francis...
25	25	25	60	High School Degree	0-2 Years	168000	Austin, TX
26	26	26	124	Postgraduate Degree	6-10 Years	242400	San Francis...
27	27	27	52	Undergraduate Degree	3-5 Years	187600	San Francis...
28	28	28	50	Undergraduate Degree	3-5 Years	184900	San Francis...
29	29	29	86	Undergraduate Degree	3-5 Years	176400	San Francis...
30	30	30	57	High School Degree	0-2 Years	162700	San Francis...

Exhibit 5: Screenshots of Shiny App

A. Home Screen



B. Create New User Tab

The screenshot shows the "Create New User" tab of the FAANG Source application. The sidebar and header are identical to the home screen. The main content area has a light blue background and displays the "Create New User:" heading. Below the heading, it says: "Please fill out the following information, then click 'Create User' to receive your User ID." A note indicates that fields marked with an asterisk are required. The form includes input fields for "First Name" and "Last Name", an "Email Address" field, a "Gender" dropdown menu (currently set to "Nothing selected"), and an "Age" slider (ranging from 18 to 50, currently set to 20). The "Create User" button is partially visible at the bottom.

C. Data Entry Tab

FAANG Source

Home

Create New User

Data Entry

Search Database

Salary Analytics

User Settings

Enter Your FAANG Job Data:

Please fill out the following information, then click 'Submit' to add an entry into the database.

NOTE: This tab is only for users previously or currently employed by a FAANG company.

* = Required

*User ID:

Type your User ID

*FAANG Company:

Facebook

Apple

Amazon

Netflix

Google

*Position:

Data Analyst 1

*Highest Level of Education Achieved:

High School Degree

*Years of Experience:

0-2 Years

*Office You Worked At:

D. Search Database Tab

FAANG Source

Home

Create New User

Data Entry

Search Database

Salary Analytics

User Settings

Search Database:

Find salary information from employees across Big Tech based on the following parameters.

NOTE: Selecting no parameters will display the entire database.

FAANG Company:

Nothing selected

Position:

Nothing selected

Education Level:

Nothing selected

Years of Experience:

Nothing selected

Office Location:

Nothing selected

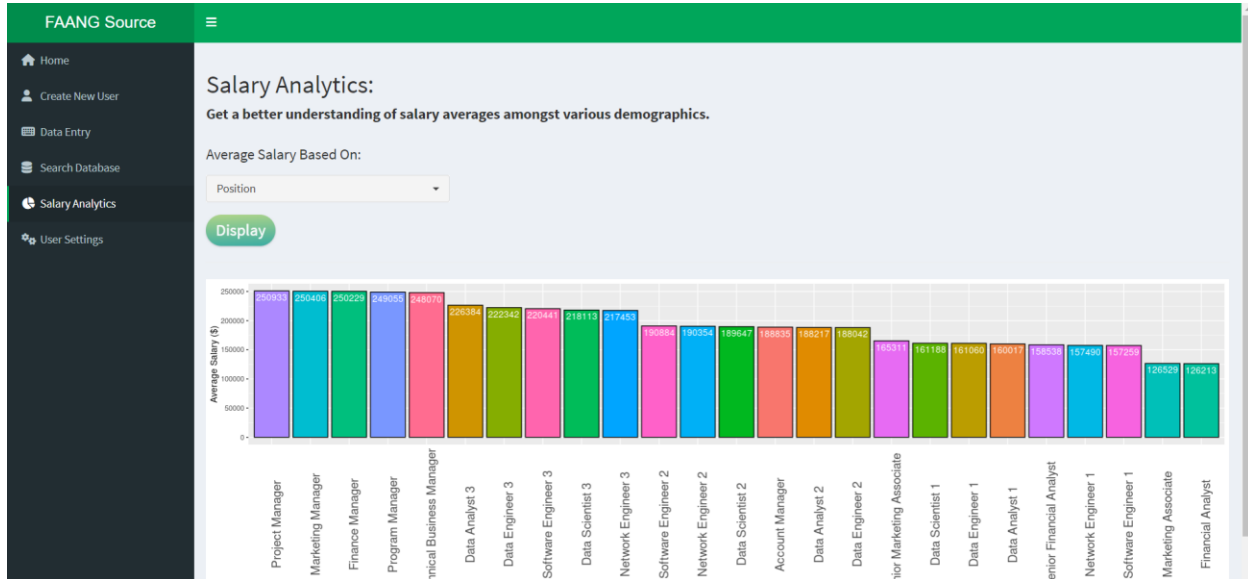
Search

Show 10 entries

Search:

	FAANG Company	Position	Education Level	Years of Experience	Office Location	Salary (\$)
1	Amazon	Program Manager	Undergraduate Degree	6-10 Years	Austin, TX	259900
2	Amazon	Program Manager	High School Degree	3-5 Years	New York, NY	259800
3	Amazon	Marketing Manager	High School Degree	3-5 Years	Austin, TX	259700
4	Apple	Marketing Manager	Postgraduate Degree	6-10 Years	New York, NY	259700

E. Salary Analytics Tab



F. User Settings Tab

FAANG Source

Home
Create New User
Data Entry
Search Database
Salary Analytics
User Settings

User Settings:

My Entries Edit Entry Delete Entry Edit User Info Forgot User ID

Type your User ID, then click 'Search' to view a table of your submitted entries.

User ID:

Type your User ID

Search

Exhibit 6: SQL to Create Blank Tables

```
1  USE ITOM6265_F22_Group25
2
3  --- MAKING BLANK TABLES
4
5  --- Making Company table where company_id is Identity PK
6  CREATE TABLE dbo.Company
7  (company_id int NOT NULL IDENTITY PRIMARY KEY,
8   company_name varchar(255) NOT NULL,
9   hq_location varchar(255) NOT NULL);
10
11 --- Making Job table where company_id is Identity PK and company_id is FK
12 CREATE TABLE dbo.Job
13 (job_id int NOT NULL IDENTITY PRIMARY KEY,
14  company_id int NOT NULL REFERENCES dbo.Company (company_id),
15  job_title varchar(255) NOT NULL);
16
17 --- Making UserInfo table where user_id is Identity PK
18 CREATE TABLE dbo.UserInfo
19 (user_id int NOT NULL IDENTITY PRIMARY KEY,
20  first_name varchar(255) NOT NULL,
21  last_name varchar(255) NOT NULL,
22  email_address varchar(255) NOT NULL,
23  gender varchar(255) NOT NULL,
24  age int NOT NULL);
25
26 --- Making UserInput table where entry_id is Identity PK, AND user_id, job_id are FK's
27 CREATE TABLE dbo.UserInput
28 (entry_id int NOT NULL IDENTITY PRIMARY KEY,
29  user_id int NOT NULL REFERENCES dbo.UserInfo (user_id),
30  job_id int NOT NULL REFERENCES dbo.Job (job_id),
31  education_level varchar(255) NOT NULL,
32  years_of_exp varchar(255) NOT NULL,
33  salary int NOT NULL,
34  office_location varchar(255) NOT NULL);
```

Exhibit 7: SQL Queries Used in Application

A.

```
1  USE ITOM6265_F22_Group25
2
3  --- CREATE NEW USER TAB QUERY
4  INSERT INTO UserInfo (first_name, last_name, email_address, gender, age)
5  VALUES ('Tim', 'Smith', 'tim.smith96@outlook.com', 'Male', '26' );
6
7
8  --- DATA ENTRY TAB QUERIES
9  INSERT INTO UserInput (user_id, education_level, years_of_exp, salary, office_location)
10 VALUES ('501', 'Undergraduate Degree', '2', '200100', 'Austin, TX');
11
12 UPDATE UserInput
13 SET job_id = (SELECT j.job_id
14               FROM Job as j
15               JOIN Company as c
16               ON j.company_id = c.company_id
17               WHERE c.company_name = 'Google' AND j.job_title = 'Data Analyst 2')
18 WHERE user_id = 501 AND salary = 200100;
19
20
21 --- SEARCH DATABASE TAB QUERY
22 SELECT c.company_name AS 'FAANG Company', j.job_title AS Position,
23        uinp.education_level AS 'Education Level',
24        uinp.years_of_exp AS 'Years of Experience',
25        uinp.office_location AS 'Office Location', uinp.salary as 'Salary ($)'
26 FROM UserInput as uinp
27 JOIN Job as j
28 ON uinp.job_id = j.job_id
29 JOIN Company as c
30 ON j.company_id = c.company_id
31 WHERE c.company_name LIKE '%%' AND j.job_title LIKE '%%'
32 AND uinp.education_level LIKE '%%' AND uinp.years_of_exp LIKE '%%'
33 AND uinp.office_location LIKE '%%'
34 ORDER BY uinp.salary DESC;
35
36
37 --- ANALYSIS TAB QUERIES
38 ---- AVG Salary (Company)
39 SELECT c.company_name AS 'FAANG Company', AVG(uinp.salary) AS 'Average Salary'
40 FROM UserInfo as uinf
41 JOIN UserInput as uinp
42 ON uinf.user_id = uinp.user_id
43 JOIN Job as j
44 ON uinp.job_id = j.job_id
45 JOIN Company as c
46 ON j.company_id = c.company_id
47 GROUP BY c.company_name
48 ORDER BY AVG(uinp.salary) DESC;
49
```

B.

```
46 ---- AVG Salary (Job Title)
47 SELECT j.job_title AS 'Position', AVG(uinp.salary) AS 'Average Salary'
48 FROM UserInfo as uinf
49 JOIN UserInput as uinp
50 ON uinf.user_id = uinp.user_id
51 JOIN Job as j
52 ON uinp.job_id = j.job_id
53 JOIN Company as c
54 ON j.company_id = c.company_id
55 GROUP BY j.job_title
56 ORDER BY AVG(uinp.salary) DESC;
57
58 --- AVG Salary (Gender)
59 SELECT uinf.gender AS 'Gender',
60        AVG(uinp.salary) AS 'Average Salary'
61 FROM UserInfo as uinf
62 JOIN UserInput as uinp
63 ON uinf.user_id = uinp.user_id
64 JOIN Job as j
65 ON uinp.job_id = j.job_id
66 JOIN Company as c
67 ON j.company_id = c.company_id
68 GROUP BY uinf.gender
69 ORDER BY AVG(uinp.salary) DESC;
70
71 --- AVG Salary (Age Range)
72 SELECT CASE
73     WHEN uinf.age < 27 THEN '20-26'
74     WHEN uinf.age BETWEEN 27 AND 33 THEN '27-33'
75     WHEN uinf.age BETWEEN 34 AND 40 THEN '34-40'
76     END AS 'Age Range', AVG(uinp.salary) as 'Average Salary'
77 FROM UserInfo as uinf
78 JOIN UserInput as uinp
79 ON uinf.user_id = uinp.user_id
80 JOIN Job as j
81 ON uinp.job_id = j.job_id
82 JOIN Company as c
83 ON j.company_id = c.company_id
84 GROUP BY CASE
85     WHEN uinf.age < 27 THEN '20-26'
86     WHEN uinf.age BETWEEN 27 AND 33 THEN '27-33'
87     WHEN uinf.age BETWEEN 34 AND 40 THEN '34-40'
88     END
89 ORDER BY AVG(uinp.salary) DESC;
90
```

C.

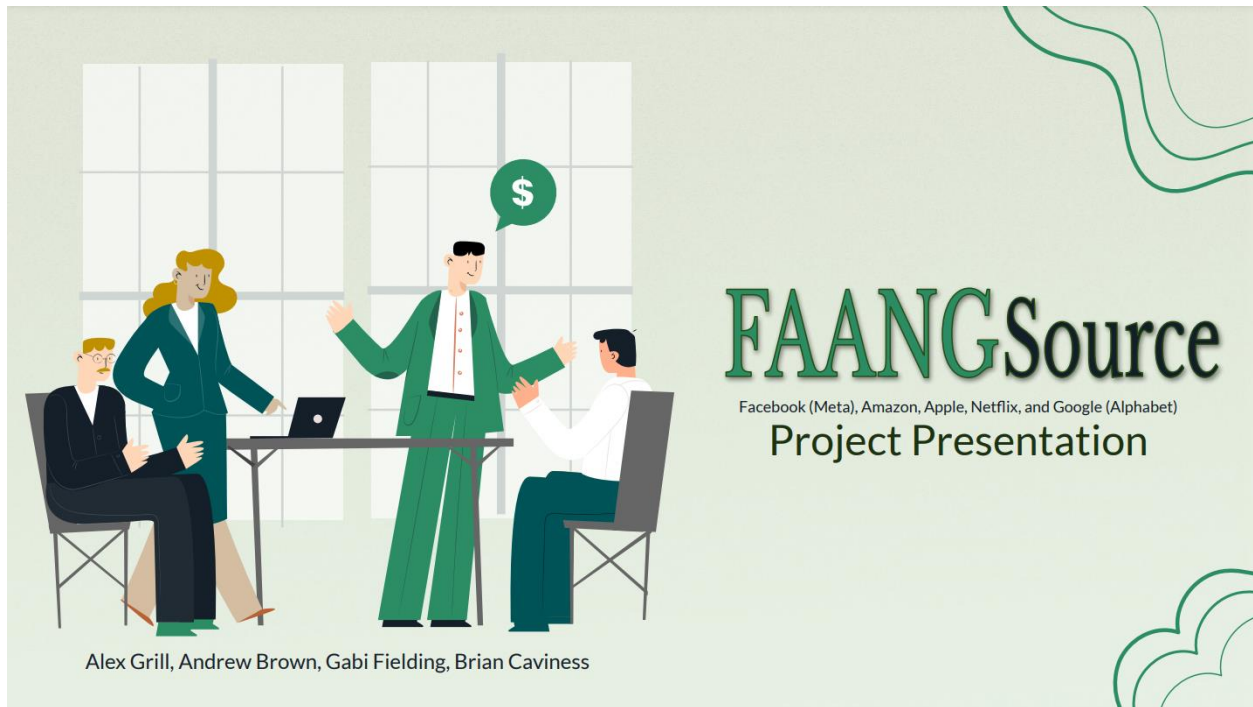
```
91 ---- AVG Salary (Level of Education)
92 SELECT uinp.education_level AS 'Education Level', AVG(uinp.salary) AS 'Average Salary'
93 FROM UserInfo as uinf
94 JOIN UserInput as uinp
95 ON uinf.user_id = uinp.user_id
96 JOIN Job as j
97 ON uinp.job_id = j.job_id
98 JOIN Company as c
99 ON j.company_id = c.company_id
100 GROUP BY uinp.education_level
101 ORDER BY AVG(uinp.salary) DESC;
102
103 ---- AVG Salary (Years of Experience)
104 SELECT uinp.years_of_exp AS 'Years of Experience', AVG(uinp.salary) AS 'Average Salary'
105 FROM UserInfo as uinf
106 JOIN UserInput as uinp
107 ON uinf.user_id = uinp.user_id
108 JOIN Job as j
109 ON uinp.job_id = j.job_id
110 JOIN Company as c
111 ON j.company_id = c.company_id
112 GROUP BY uinp.years_of_exp
113 ORDER BY AVG(uinp.salary) DESC;
114
115 ---- AVG Salary (Office Location)
116 SELECT uinp.office_location AS 'Office Location', AVG(uinp.salary) AS 'Average Salary'
117 FROM UserInfo as uinf
118 JOIN UserInput as uinp
119 ON uinf.user_id = uinp.user_id
120 JOIN Job as j
121 ON uinp.job_id = j.job_id
122 JOIN Company as c
123 ON j.company_id = c.company_id
124 GROUP BY uinp.office_location
125 ORDER BY AVG(uinp.salary) DESC;
126
127
128 --- USER SETTINGS TAB QUERIES
129 --- User Settings Tab: My Entries Query
130 SELECT uinp.entry_id AS 'Entry Number',
131        uinf.first_name + ' ' + uinf.last_name AS 'Name',
132        c.company_name as 'FAANG Company', j.job_title AS Position,
133        uinp.education_level AS 'Education Level',
134        uinp.years_of_exp AS 'Years of Experience',
135        uinp.salary AS 'Salary ($)', uinp.office_location AS 'Office Location'
136 FROM UserInput AS uinp
137 JOIN UserInfo AS uinf ON uinp.user_id = uinf.user_id
138 JOIN Job as j ON uinp.job_id = j.job_id
139 JOIN Company as c ON j.company_id = c.company_id
140 WHERE uinp.user_id = 502;
141
```


D.

```
142 --- User Settings Tab: Edit Entry Queries
143 UPDATE UserInput
144 SET education_level = 'Postgraduate Degree'
145 WHERE user_id = 502 AND entry_id = 505;
146
147 UPDATE UserInput
148 SET salary = 195800
149 WHERE user_id = 502 AND entry_id = 505;
150
151 --- User Settings Tab: Delete Entry Query
152 DELETE
153 FROM UserInput
154 WHERE user_id = 502 AND entry_id = 504;
155
156 --- User Settings Tab: Edit User Info Queries
157 UPDATE UserInfo
158 SET first_name = 'Jeffrey'
159 WHERE user_id = 502;
160
161 UPDATE UserInfo
162 SET gender = 'Male'
163 WHERE user_id = 502;
164
165 UPDATE UserInfo
166 SET age = 26
167 WHERE user_id = 502;
168
169 SELECT first_name + ' ' + last_name AS 'Name',
170        email_address AS 'Email Address',
171        gender AS 'Gender', age AS 'Age'
172 FROM UserInfo
173 WHERE user_id = 502;
174
175 --- User Settings Tab: Forgot User ID Query
176 SELECT uinf.user_id
177 FROM UserInfo as uinf
178 WHERE uinf.first_name = 'Paxton' AND uinf.last_name = 'Greene';
```

Exhibit 8: Presentation Slides

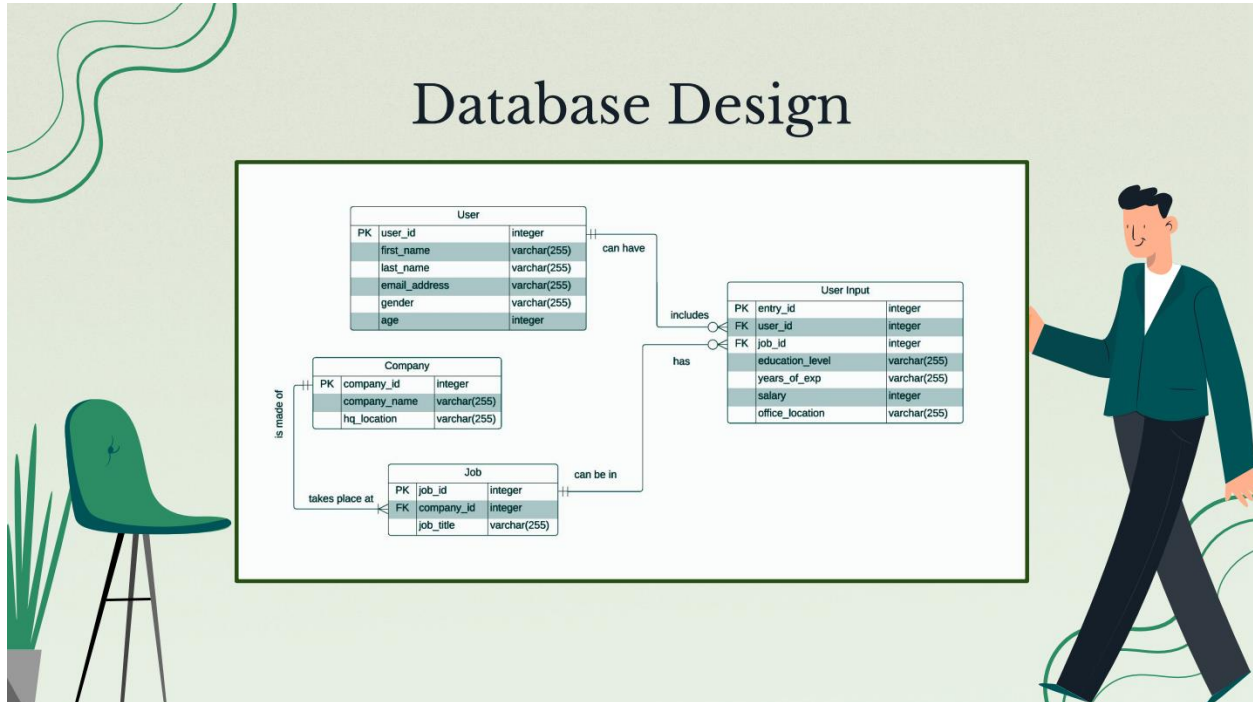
A. Slide 1



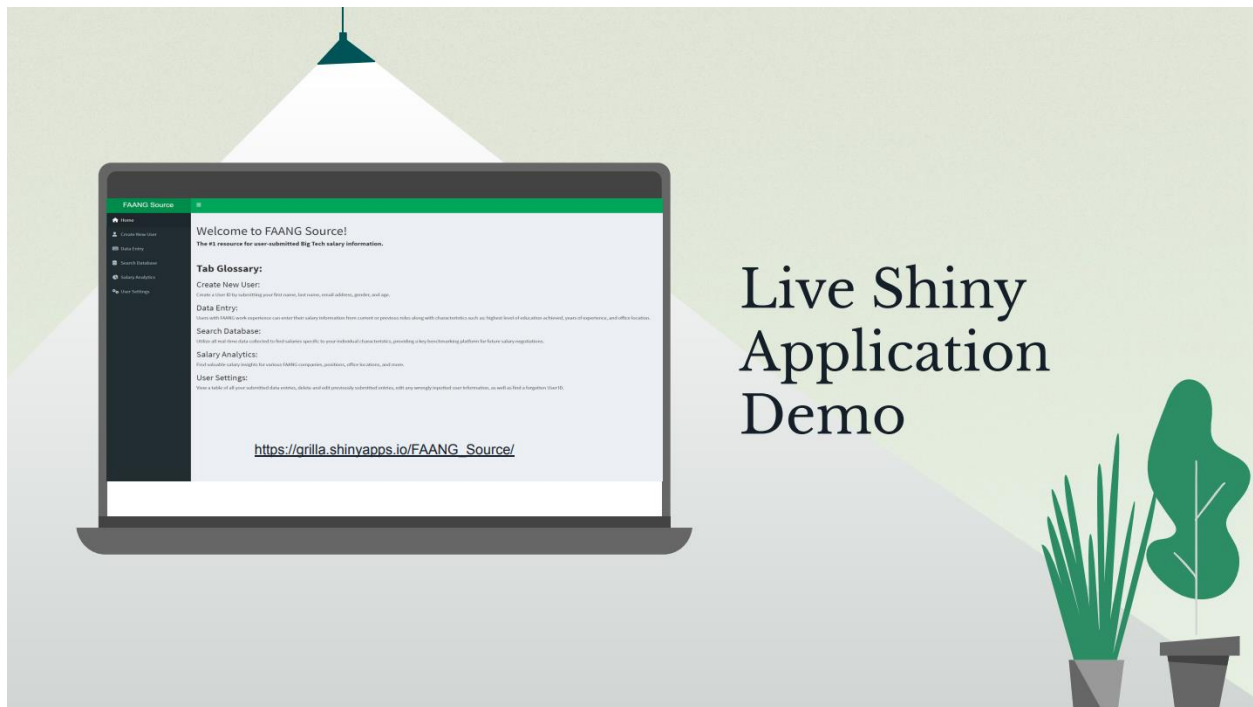
B. Slide 2



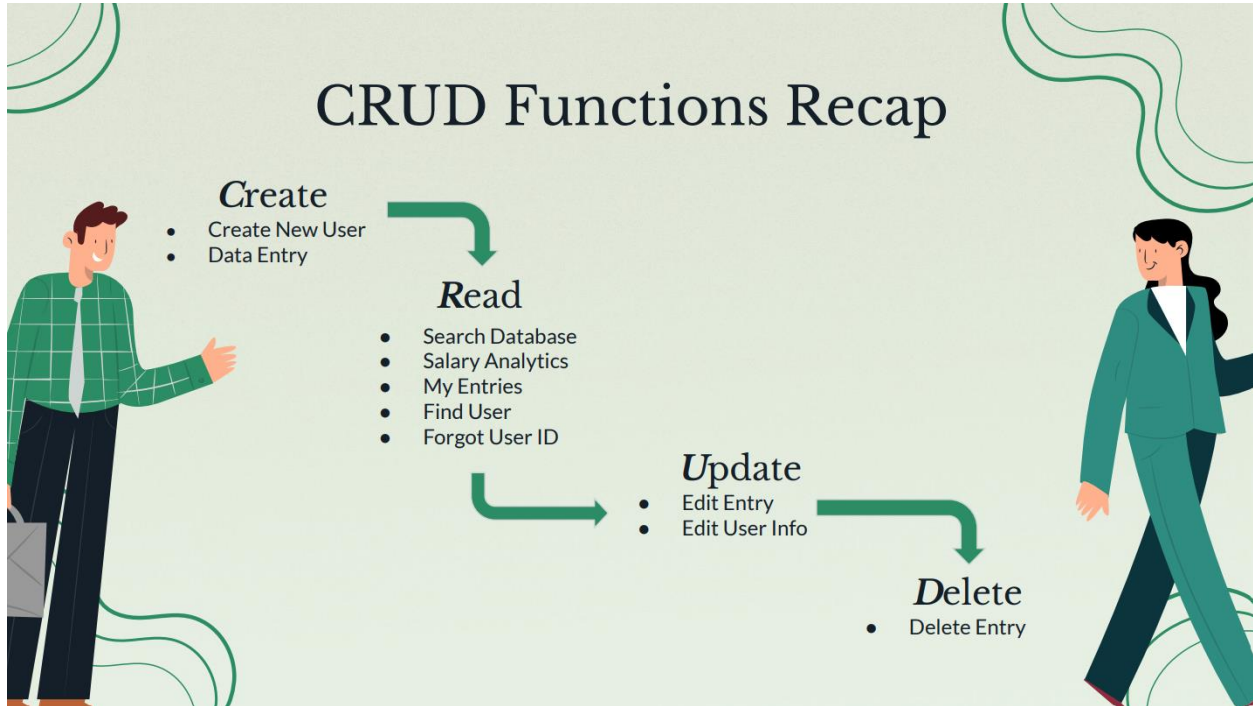
C. Slide 3



D. Slide 4



E. Slide 5



F. Slide 6

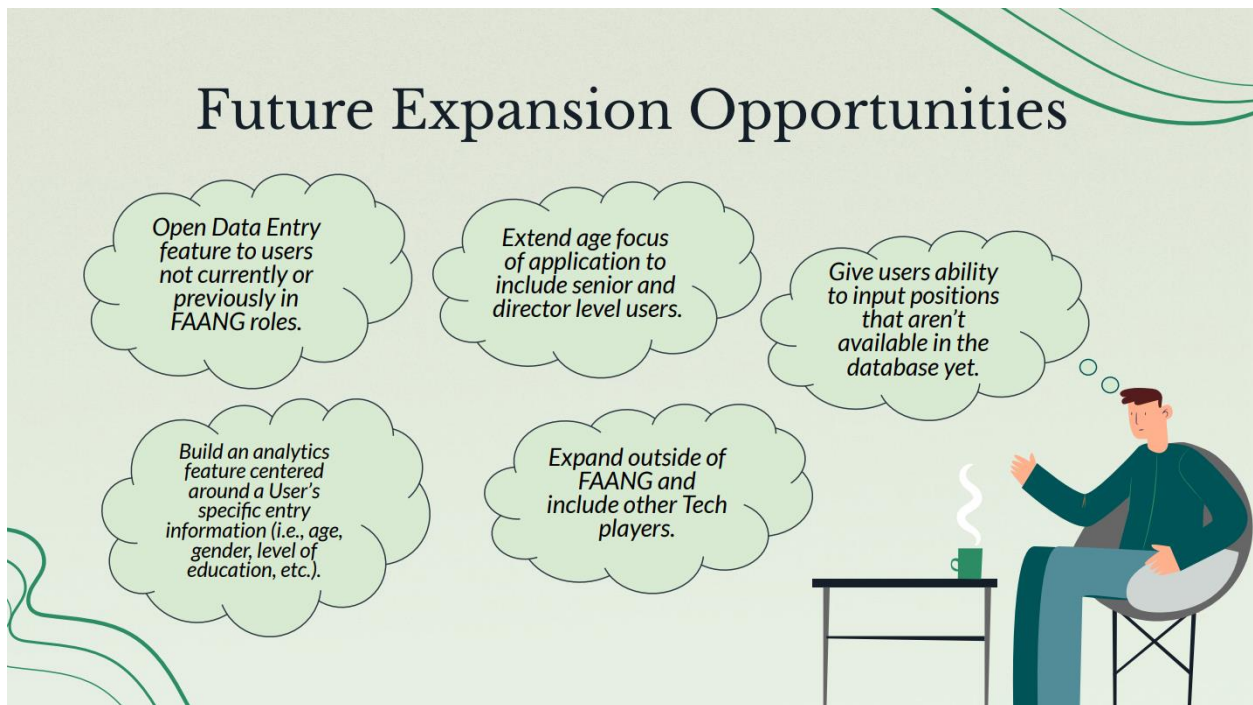




Exhibit 9: FAANG Source, Shiny App Link

https://grilla.shinyapps.io/FAANG_Source/