Team: **DINGO**
Members: *Jason Neal*
Hanna Loboda
Nachal Perichiappan

Project Name: **Career Explorer**

MediaSpace: https://mediaspace.illinois.edu/media/t/1_fk0poz0x
Server Link: http://sp18-cs411-dso-009.cs.illinois.edu:8000
GitHub Link: https://github.com/mahajanashu/CS411_Team_DINGO

Project Description:

Career Explorer is an interactive web application that utilizes datasets from Bureau of Labor Statistics for a period of 3 years (2014-2016). The dataset is ported into a database which is the main source behind the queries displayed on the website. The user is able to sign up, create/update a profile, see ranking statistics with respect to his/her occupation, run various interactive queries and save favorite queries on the profile. The data is dynamically updated and populated following a change, and is accompanied by summary tables, visual graphs, and charts.

Stage 5:

ADVANCED FUNCTIONALITY:

- **Enhanced the Profile page** where User's current profile input is evaluated against the following metrics: salary, residing state, education, and the employment popularity. The data from the user's profile is compared against the dataset averages for that occupation and allows user to see their overall standing on the metrics. The percentage is calculated based on user's current salary and against the overall national and state. In addition, a degree level breakdown is provided to the user to how his/her occupation is compared in terms of degree requirements and count holders. Whenever user is updating the profile, the metrics in the tables are re-calculated and updated. Furthermore, a visual chart was added that displays the most employed states for user's occupation. From here user can also check the average salaries in the interested states.
- **Implement Favorites:** Allow user to save favorite queries on the profile page so that there is never a need to rerun the same queries. Favorite queries can be saved, deleted, or updated. Favorite queries are stored in the table "Favorites" and there is a foreign key reference to the user's table. No favorite saved query should "linger" without a user.
- **Explorer Page** allows user to query various information from the database. Explorer queries are run using Stored Procedure technique that lives on the database itself.

- **Stored Procedures:** The following procedure is getting top ten jobs on the decline:

```
CREATE DEFINER=`remote`@`%` PROCEDURE `GET_DECLINE_JOB`(yearid int(12))
BEGIN
SET @ROWN=0, @OCC1=' ',@STATE1 = ' ';
SELECT OCC_TITLE from (
SELECT DISTINCT CASE WHEN NUM1 =1 AND YEAR_ID = yearid THEN 'DECLINE'
WHEN NUM1 = 1 AND YEAR_ID<> yearid THEN 'PROGRESS'  END AS
decline1,state_id,occ_code1,OCC_TITLE from (
SELECT @ROWN :=CASE WHEN @OCC1 = OCC_CODE AND @STATE1 = STATE_ID THEN  @ROWN +
1
ELSE 1 END AS NUM1,
            @OCC1 := OCC_CODE as occ_code1, @STATE1 := STATE_ID as
state_id,TOT_EMP,YEAR_ID,OCC_TITLE
FROM OCC_STATS WHERE OCC_CODE <> '00-0000' ORDER BY STATE_ID,OCC_CODE,TOT_EMP)
as t1) as d1 WHERE decline1 = 'DECLINE' limit 10;
END
```

Another procedure is getting educational levels data by the occupation:

```
CREATE DEFINER=`remote`@`%` PROCEDURE `GET_EDUL_DATA`(LEVELD VARCHAR(100))
BEGIN
Select OCC_TITLE, STATE_ID,MAX(H_MEDIAN) FROM OCC_STATS WHERE OCC_CODE IN
(SELECT OCC_CODE FROM OCCUPATION WHERE EDUCATIONLEVELID IN
(SELECT EDUCATIONLEVELID FROM EDUCATIONALLEVEL WHERE LEVELDESCRIPTION like
LEVELD)) GROUP BY OCC_TITLE;
END
```

We've implemented 8 such stored procedures in our database for this project.

- **Searchable, Sortable, Filterable Data:** We added tables that allow sorting and filtering for specific results, which are also accompanied by a chart that responds to filtering of the returned data.