Tech Career Analytics Dashboard

By Ismail Omer, Yargi Kilinc, Archit Hallan, Tania Barrera

PROJECT OVERVIEW

Objective: Empower job seekers, employers, and HR firms with salary insights.

Target Audience: Data professionals, employers, bootcamp candidates.

Benefits: Informed job decisions, global HR strategies.

The main purpose of our dashboard is to offer actionable insights to a variety of users:

- Employers can make informed hiring decisions.
- Job seekers can identify high-paying opportunities.
- Bootcamp candidates can assess their career prospects.
- HR companies can refine their global strategies

Our dashboard addresses several key questions, including:

- Which countries offer the highest salaries?
- What are the top-paying job titles by country?
- How do salaries vary with experience levels?"

ETL PROCESS

Extraction:

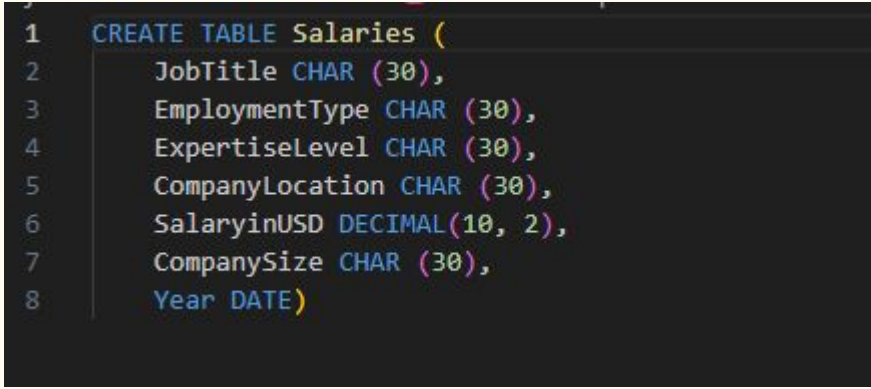
- Source data from a CSV file on Kaggle.
- Table schema is provided for understanding the data structure.

Transformation:

- Conducted data cleaning, focusing on: Dropped duplicate columns related to experience/expertise level, location, and salaries.
- Ensured data consistency and quality.

Loading:

- Created an engine to establish a connection to a PostgreSQL database.
- Checked if the database exists; if not, created it.
- Exported the transformed data into the PostgreSQL database.



```sql
CREATE TABLE Salaries (
    JobTitle CHAR (30),
    EmploymentType CHAR (30),
    ExpertiseLevel CHAR (30),
    CompanyLocation CHAR (30),
    SalaryinUSD DECIMAL(10, 2),
    CompanySize CHAR (30),
    Year DATE)
```

FLASK API ROUTING

In our project, we've integrated Flask to create a robust system of API routes. To ensure a user-friendly experience, we've designed our API routes following a clear and intuitive structure: /api/v1.0/...

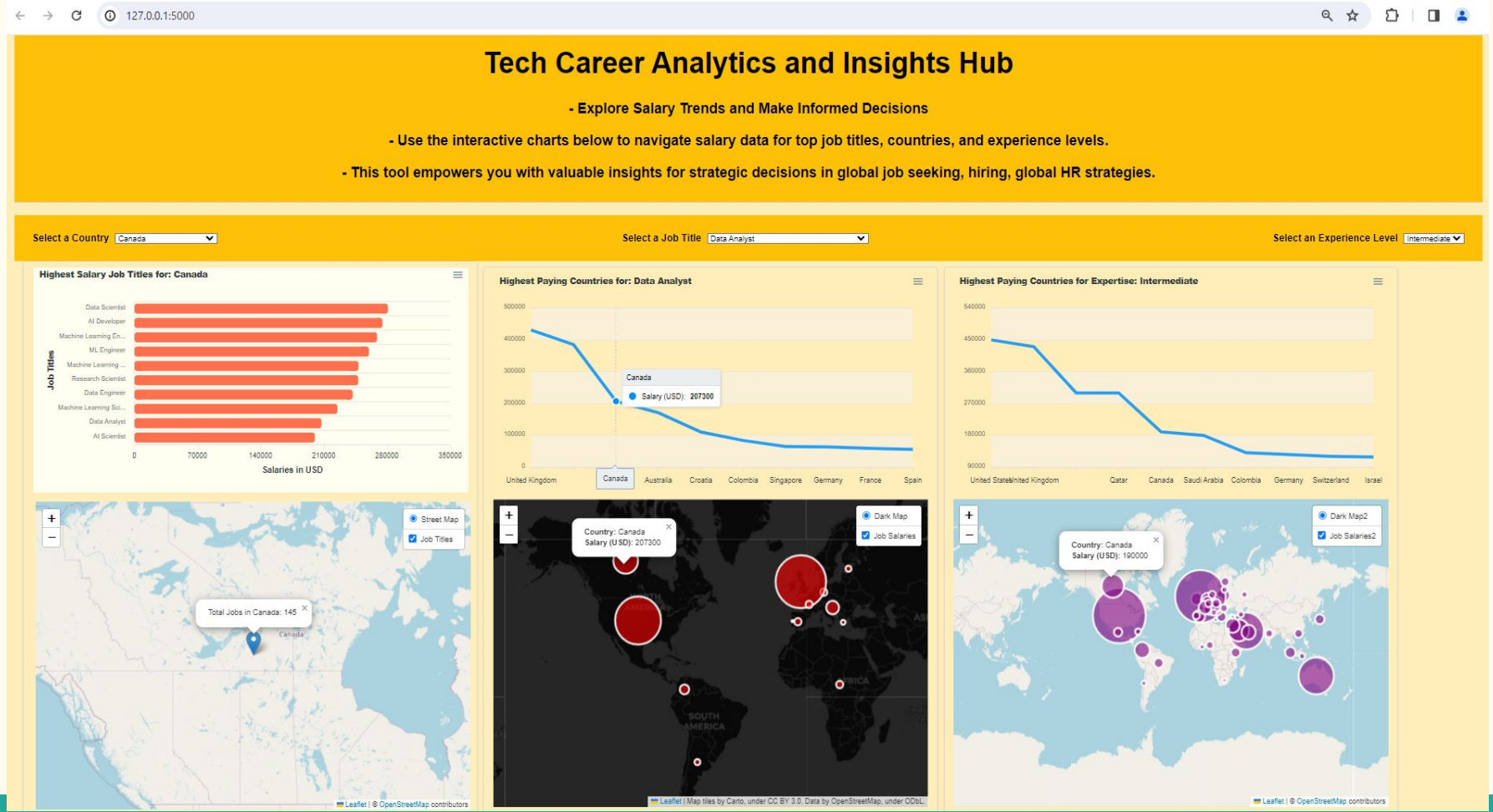
Example API Routes:

- /api/v1.0/salaries: Endpoint for retrieving salary data.
- /api/v1.0/country/{country}/top10_job_titles: Endpoint to get the top 10 job titles in a specific country by salary.
- /api/v1.0/job_title/{job_title}/top10_countries: Endpoint to get the top 10 countries for a specific job title by salary.
- /api/v1.0/experience_level/{experience_level}/top10_countries: Endpoint to get the top 10 countries for a specific experience level by salary.
- /api/v1.0/experience_level/{experience_level}/all_countries: Endpoint to get salary data for all countries for a specific experience level.

Enhanced Data Exploration:

These API routes are the backbone of our dashboard's interactivity.
They empower users to tailor their data queries, enhancing the overall user experience.
In summary, our Flask API routing system simplifies data access, making our dashboard highly customizable and user-centric.

DASHBOARD OVERVIEW



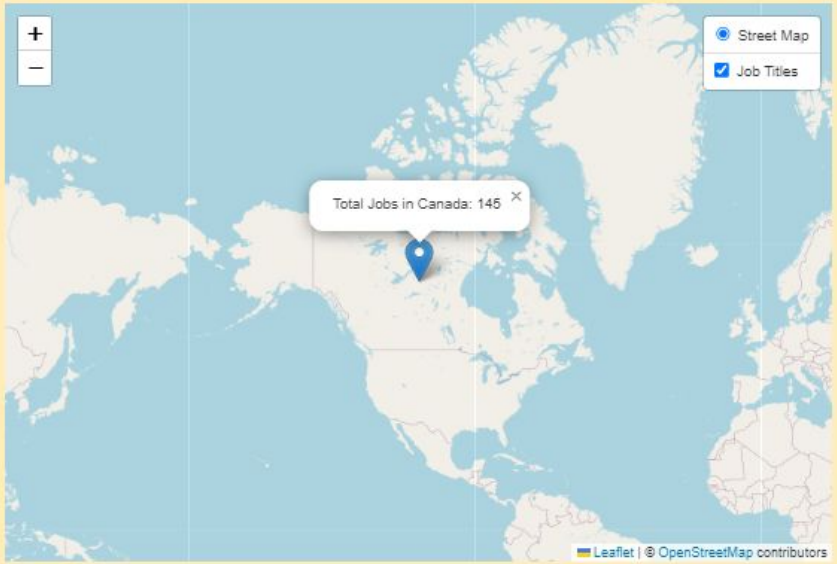
FRONT END

How to Access the Dashboard:

- To access the dashboard, follow these steps:
- Clone our project repository to your local machine.
- Navigate to the project directory using your terminal.
- Run the Flask app using the command 'python app.py'.
- Open a web browser and go to 'http://localhost:5000/' to access the dashboard.

Key features of the dashboard include:

- Interactive charts for exploring the dataset and salary trends.
- Dropdown menus to select countries, job titles, and experience levels for customized insights.
- Real-time updates and visualizations that respond to user selections.



CHALLENGES FACED

During our project journey, we encountered several key challenges, each driving us to think creatively and find solutions. These challenges included:

Introducing New Libraries: We explored new possibilities by using the jsdelivr.com library. This helped us enhance our project with interactive charts using ApexCharts, making data visualization more engaging.

Extending API Routes: As our project evolved, we needed to offer users more data options. To meet this need, we expanded our Flask application with additional API routes, like /api/v1.0/job_title/<job_title_name>/top10_countries, giving users more insights and flexibility.

Handling Missing Geocodes: We faced difficulties when plotting data on maps due to missing latitude and longitude information. To address this, we relied on the Nominatim service from openstreetmap.org, ensuring accurate geographic representation.

Code Organization: Initially, our code was in a single file (logic.js). But as the project grew in complexity, we split it into three parts: country.js, jobs.js, and experience.js, making it easier to manage and understand.

Navigating Naming Challenges: Naming functions, variables, URLs, and map creation methods was sometimes tricky. These names are crucial for user understanding and system coherence.

HTML Refinements: Our HTML file underwent multiple revisions, mainly concerning the structure and div class IDs. These changes were vital to seamlessly integrate our dashboard components.

Addressing these challenges sparked creativity and strategic thinking, resulting in a more robust, user-friendly, and feature-rich project.

CONCLUSION

Critical Questions Answered: Our project successfully addresses a series of critical questions:

Salary Disparities: We've uncovered which countries offer the highest salaries, giving users the ability to explore lucrative markets.
Top-Paying Job Titles: Users can easily identify the top-paying job titles by country, a valuable asset for job seekers and employers alike.
Experience-Level Analysis: Our dashboard provides insights into how salaries vary based on experience levels, aiding candidates in their career planning.

In conclusion, our project has achieved its objectives by delivering a comprehensive and user-friendly analytics tool. We're confident that this tool will empower our audience, fostering better decision-making and strategic thinking in the data job market. Thank you for your time and attention, and we welcome any questions or feedback you may have.