# Final Report

## Abstract

This project addresses the challenge of accessing specific Major League Baseball (MLB) statistical information from a vast dataset, particularly for users with little to no programming experience. By leveraging a simple web interface hosted on AWS, users can obtain aggregated and analyzed statistics on-demand, along with customized visualizations.

## Problem Statement

With a 150+ year history, Major League Baseball (MLB) boasts an extensive array of statistical information. However, finding specific data or answering specific questions can be daunting, especially for those without programming skills. The Lahman datasets, a gold standard for MLB statistics, contain thousands of entries, making manual data retrieval complex. This project aims to solve this problem by providing a user-friendly interface for data access and analysis.

## Related Work

There are various approaches that have been explored to handle large datasets and provide meaningful insights through user interfaces. Existing solutions typically cater to users with some technical expertise. This project differentiates itself by focusing on ease of use, enabling broader accessibility to MLB statistics through an intuitive web interface.

## Your Solution and Its Significance

This project’s solution involves:

* **Backend**: Python methods to read and analyze Lahman datasets (CSV format) for various user queries. The use of Flask to handle HTTP API requests.
* **Frontend**: A web interface in React will allow users to make data requests without technical knowledge.
* **AWS Infrastructure**: Utilizing serverless architecture for efficient, scalable, and cost-effective performance. Services such as AWS Lambda, API Gateway, and S3.

## Novelties

* **Integration of extensive datasets**: Seamlessly integrating complex datasets into a user-friendly interface.
* **Automated visualizations**: Providing dynamic visual representations of data.
* **Serverless architecture**: Ensuring robustness and scalability.

Evaluation Results

The evaluation will focus on:

* **Performance**: Response time and scalability of the system under various loads.
* **Usability**: User feedback on the ease of using the interface.
* **Accuracy**: The precision of data retrieval and analysis.

## Contributions

***Cole Bromfield***

* Assembled complex MLB datasets.
* Developed backend functions for user queries (e.g., stat leaders, team info).
* Implemented import statements and data cleaning for backend accessibility.
* Updated README for Windows instructions.
* Added visualization functions.
* Began work on the presentation.

***Mamadou Ndiong***

* Worked on the frontend web page.
* Created wireframe for user-friendly data analysis requests.
* Added visualization features.
* Built frontend (connectivity to backend pending).

***Dennis Reyes***

* Created IAM Roles.
* Set up API Gateway for REST API requests.
* Developed Lambda functions for backend handling.
* Configured S3 for static website hosting.
* Built CI/CD pipeline for deployments.
* Integrated React with Serverless Framework.
* Created a sample implementation for frontend-backend communication.

## Implementation Tools

* **Python**: Pandas, NumPy, Matplotlib, possibly Scikit-learn.
* **Backend:** Flask, CSV, REST API, Python
* **Frontend:** JavaScript, React, Vite
* **AWS**: Serverless Framework, CloudFormation, Lambda, S3, API Gateway, IAM.

Softcopy Artifacts

Please include the following softcopy artifacts with your submission:

* **Source Code Repository** [**URL**](https://github.com/ColeBromfield01/DATA650_project)
* **Backend** [**URL**](https://osir3dme2h.execute-api.us-east-1.amazonaws.com/dev/)
  + **Backend GET method routes**:
    - /read\_csv
    - /team\_lookup
    - /batting\_leader
    - /pitching\_leader
    - /period\_batting\_leader
    - /period\_pitching\_leader
    - /visualize\_batting\_leaders
    - /visualize\_pitching\_leaders
* **Frontend Website** [**URL**](http://msml-data-650-front-end-static-cb-dr-mn.s3-website-us-east-1.amazonaws.com/)