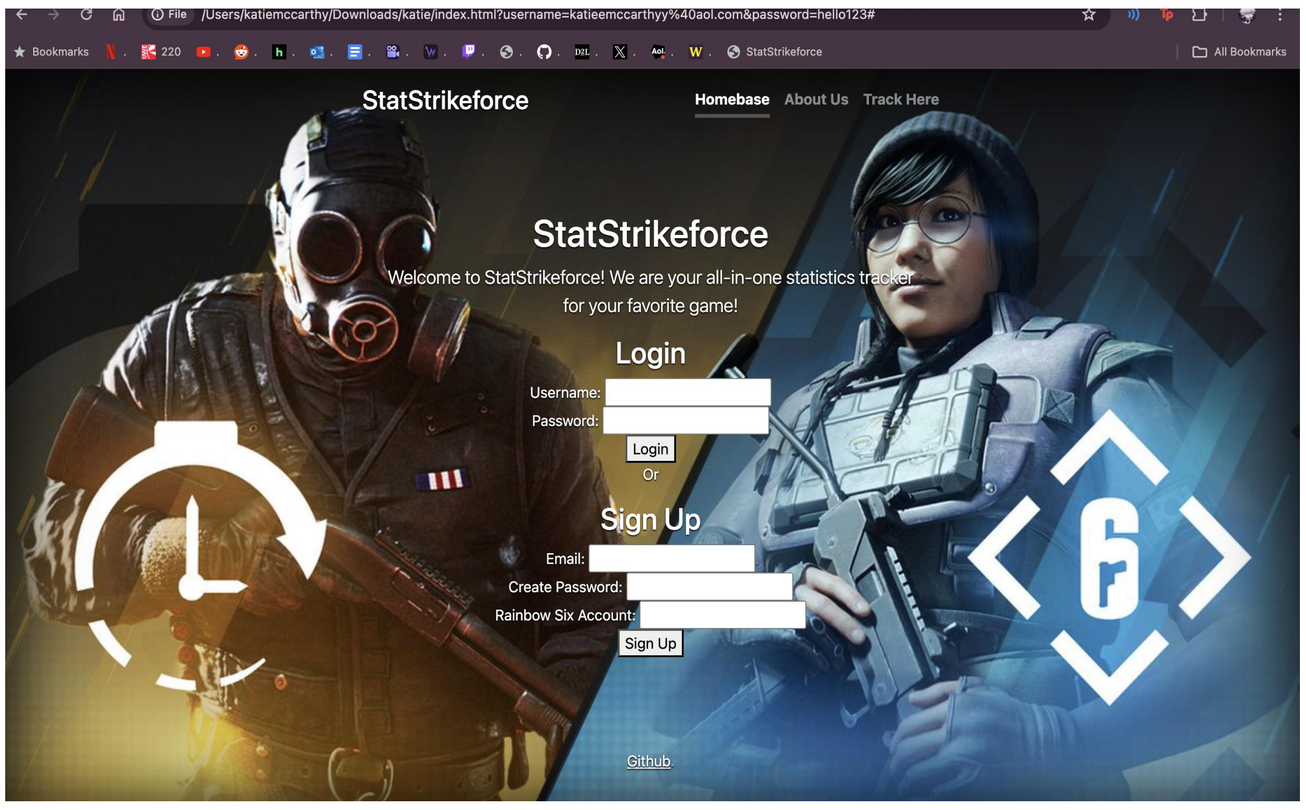
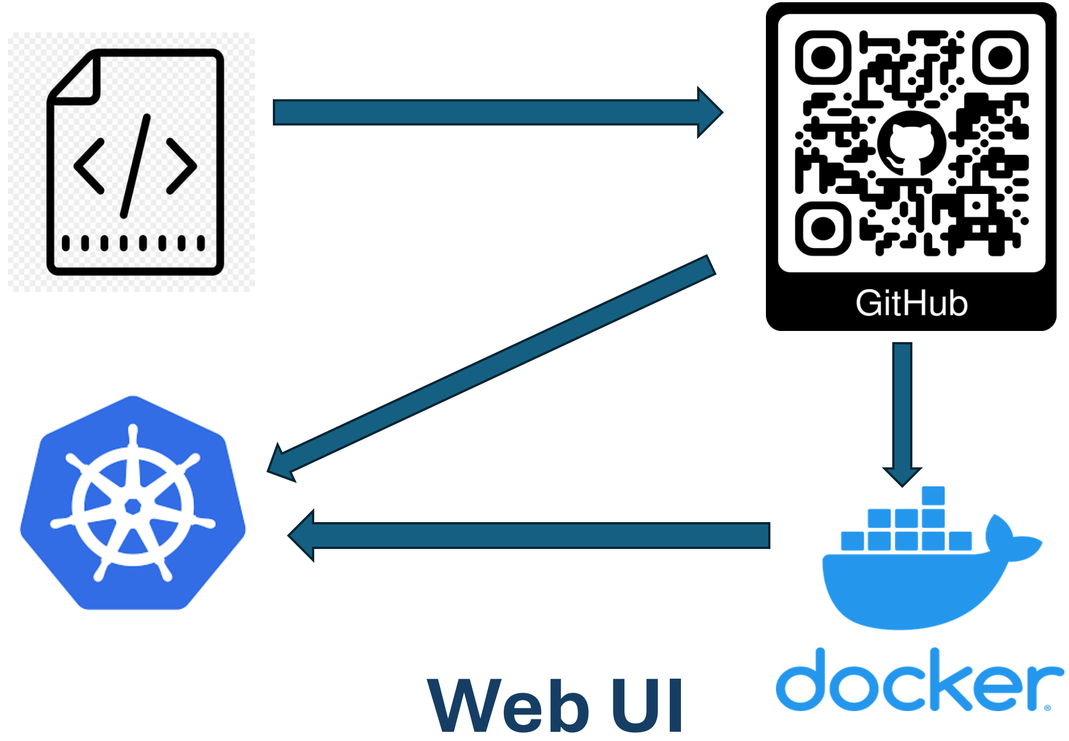
STATSTRIKEFORCE

Katherine McCarthy, Maxwell Mendenhall Ryan Sayre, Tobyn Sitar

PROJECT OVERVIEW

StatStrikeforce is a cloud-base web application, designed to reinvent the way you track your favorite video game statistics. The application allows for users to track their headshot percentage, Kill/Death ratio, win rate, and many other useful statistics with ease. StatStrikeforce gives players a competitive edge through its prediction feature, which uses machine learning to predict a player's performance and probability of a win in future matches based on recent match performance.





Web UI

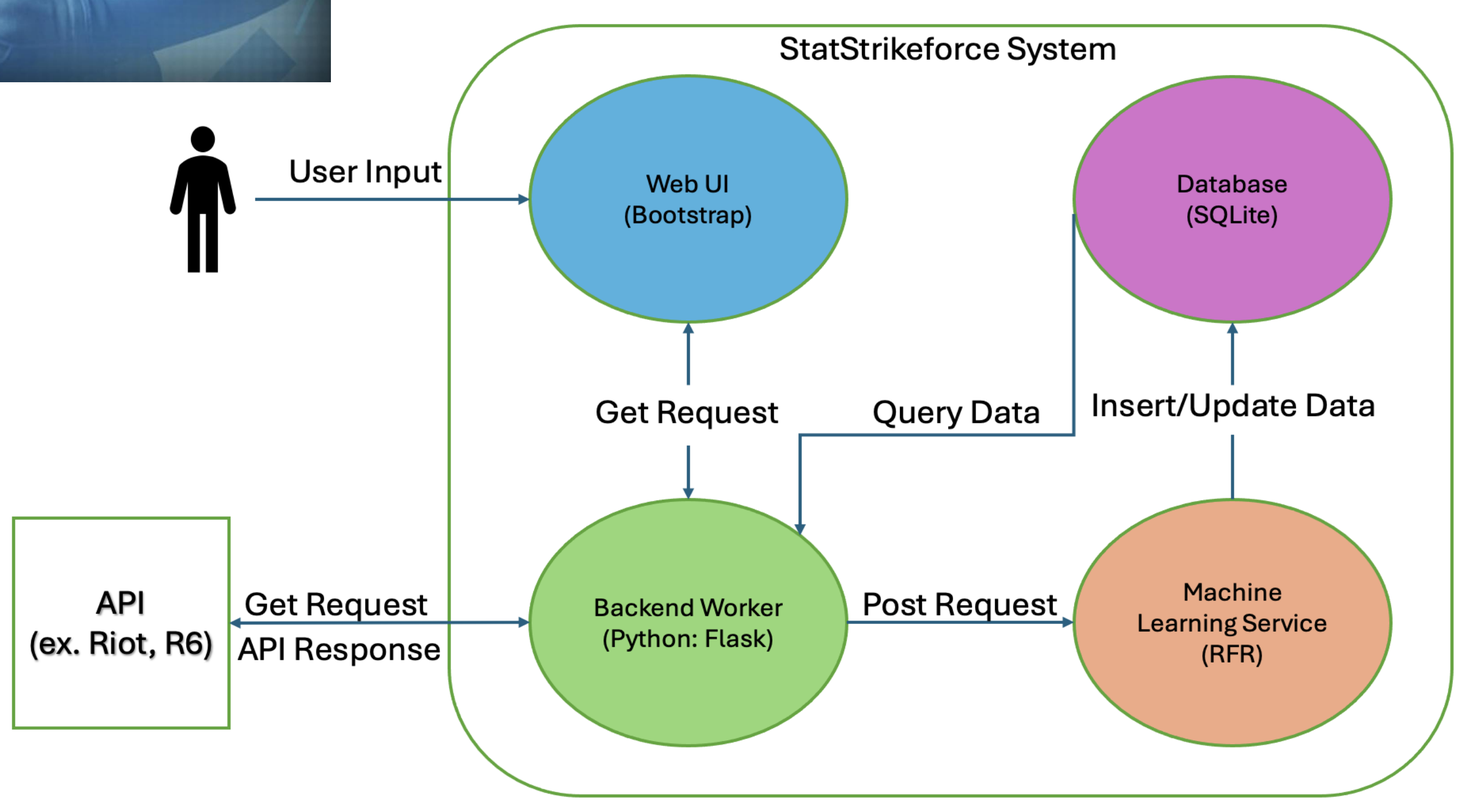
The website features three sections:

- Homepage: Prompt user to create an account with their R6 username, then login. Will be redirected to Track Here page afterwards.
- About Us: General information about team members, what each of us contributed on and goals for the project in the future.
- Track Here: Prompts user to generate new stats, then displays relevant statistics from Siege API along with our machine learning prediction.

We used Bootstrap and its libraries along with JavaScript to form our vision for the user interface.

Concept Architecture

1. The backend worker will communicate with the front end via GET requests.
2. Then backend will communicate with the Rainbow Six API to get user info.
3. POST requests will be sent to the Machine Learning service.
4. ML service will then query the data into the database, with the primary key being UserID.
5. Each UserID will have corresponding prediction values made with the user data from the API.



schema.sql

```sql
DROP TABLE IF EXISTS user_stats;
DROP TABLE IF EXISTS user;

CREATE TABLE user_stats
(
    id          INTEGER PRIMARY KEY AUTOINCREMENT,
    user_id     TEXT UNIQUE NOT NULL,
    mse_attack REAL,
    mse_defend REAL,
    FOREIGN KEY (user_id) REFERENCES user (id)
);

CREATE TABLE user
(
    id            INTEGER PRIMARY KEY AUTOINCREMENT,
    username      TEXT UNIQUE NOT NULL,
    password_hash TEXT        NOT NULL,
    r6_user_id    TEXT,
    created_at    DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

Database Design

- For our database design, we chose SQLite due to its space-saving capabilities and its compatibility with our backend through Python libraries, enabling efficient data storage and retrieval.
  - User Stats Table: Linked to the user via '**user_id**', this table tracks game statistics. It uses the '**user_id**' to connect user profiles performance from Rainbow Six Siege to their metrics in '**mse_attack**' and '**mse_defend**'
  - User Table: This table is for our login system. It stores the '**id**' as the primary key along with '**username**', '**password_hash**' for secure authentication, and the '**r6_user_id**'. The '**id**' is unique to every user and auto-increments with new entries.

Cloud Deployment

- Source code written and tested locally, before pushing to our project's GitHub repository.
- GitHub will then trigger the build process to have Docker push updated images.
- Kubernetes will pull images from DockerHub and run them in pods.
- GitHub will also push a deployment update to Kubernetes directly.
- Once the application is deployed, we can test functionality and plan for future builds.