



**NYU**

**TANDON SCHOOL  
OF ENGINEERING**



**Foot365**

**A football manager for you**

## **Project Report**

**Aditya Jain, aj2529**

**Amartya Singh, as11867**

**Nikhila Dwarakanath, nd1556**

**Pratik Pravin Rathi, ppr239**

## Contents

<b>Problem Statement.....</b>	<b>3</b>
<b>1. Overview.....</b>	<b>4</b>
<b>2. Assumptions .....</b>	<b>4</b>
<b>3. Architecture Diagram .....</b>	<b>4</b>
<b>4. Technology Stack.....</b>	<b>5</b>
<b>5. External API integrations .....</b>	<b>5</b>
<b>6. Implementation.....</b>	<b>5</b>
<b>Frontend.....</b>	<b>5</b>
<b>Integration .....</b>	<b>6</b>
<b>APIs: .....</b>	<b>6</b>
<b>AWS Managed Kafka.....</b>	<b>7</b>
<b>Lex .....</b>	<b>8</b>
<b>SNS .....</b>	<b>9</b>
<b>AWS SageMaker .....</b>	<b>9</b>
<b>7. Future Work.....</b>	<b>11</b>

## **Problem Statement**

We aim to serve the football community by providing round-the-clock services regarding match fixtures, standings, schedules and screenings available to them.

The main idea behind this project is to unite passionate football enthusiasts who wish to be up-to-date with the Football seasons. We manage their teams' schedule, provide them reminders about upcoming games, suggest sports screening recommendations near them, set up email and SMS reminders on a request basis, speech-to-text interaction with our application with whatever queries the user has.

Users can signup/register into our app to avail these features. After signing up, the user can select favorite teams and he will get round-the-clock information regarding schedules, fixtures, lineups, standings, predictions, live scores and other related information. Once a user gets this information, they can plan accordingly.

Furthermore, the user can converse with the application to know the nearest restaurant/bar screening for the upcoming match of interest. For instance, if a user is in Brooklyn and requests the app to show the nearest sports bars that provide screening for that match, the application can cater to this request and make a reservation at the given restaurant/bar of choice.

Integrating this with Alexa, it becomes a state-of-the-art speech-to-text engine with a highly scalable application for football enthusiasts.

## 1. Overview

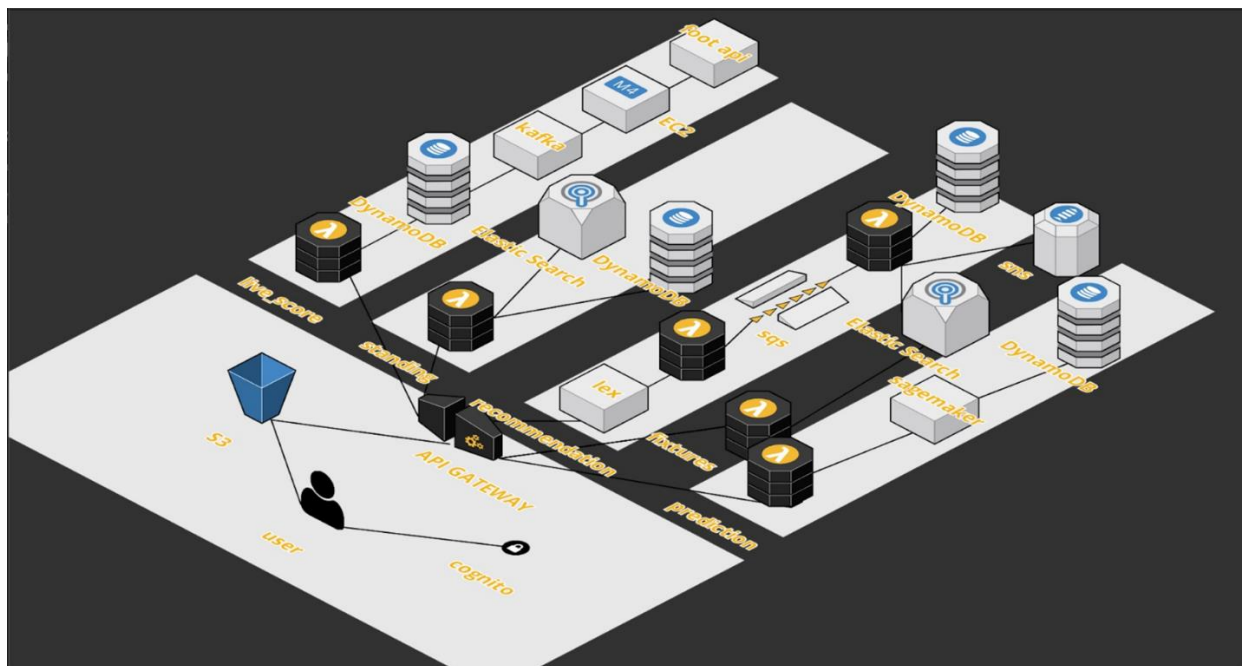
Our objective with this project was to integrate cloud-based services for the football community. We manage the teams' schedule, provide them reminders about upcoming games, suggest sports screening recommendations near them, set up SMS reminders on a request basis.

We have also incorporated features such as predictions for the final week of the season as well as live scores for the ongoing matches.

## 2. Assumptions

- We have implemented our application with the league in consideration to be “English Premier League”
- We started building the app with a given set of ideas but now the league has ended. And hence fixtures are not available for the next season, so we are currently assuming the league for 2018/2019 is in play.
- During the demonstration, we have simulated already ended games and given data such that live scores can be displayed using this data.
- Our predictions for the final week of season assumes that all the matches played in the current season barring the last week are considered.

## 3. Architecture Diagram



## 4. Technology Stack

### Front end:

- HTML
- CSS
- Bootstrap
- JQuery

### Back end:

- Python
- Node

### AWS Services used:

- Cognito & IAM for Authentication, Security Management,
- S3 for Static Web Hosting, Data Storage,
- API Gateway for routing and configuring business logic
- Lambda for incorporating modular logic functionality
- SQS and SNS for queuing and notification services
- Sagemaker for match predictions using ML
- EC2 for Kafka (with Apache Avro) for live score update
- DynamoDB and Elasticsearch for database & indexing.
- Lex for Chatbot application to provide recommendations
- Cloudwatch for monitoring, trigger and analytics

## 5. External API integrations

- FOOTBALL API
- Google Places API

## 6. Implementation

Here, we discuss how each module in the architecture of FitBook works, and their integration with each other.

### Frontend

The frontend was implemented using JavaScript, jQuery and Adobe PhoneGap, an app development platform for web developers to build amazing cross platform mobile, web, and desktop apps.

This allows your frontend to be highly responsive and act as a mobile web application, and can even be built into a mobile app for IOS, Android and Windows.

The frontend is hosted on Amazon S3 and integrated with Cognito for authentication services. Users signed up are added to the Cognito user pool.



### Integration

Our application follows a PULL, PUSH and GET model with regards to various modules. Below are details about the APIs that were hit from the frontend.

#### APIs:

Get Fixtures

Get Live Scores

Get Score Predictions

Get recommendations of bars on where a match is being screened

Gets results of previous matches.

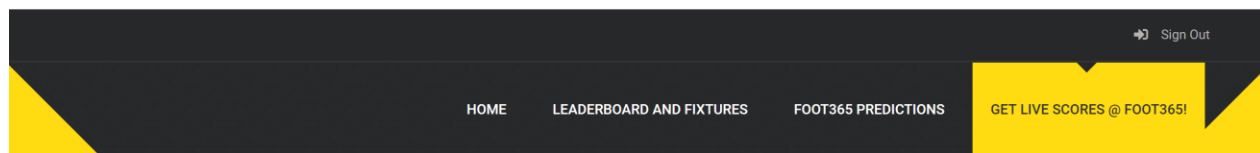
Get standings of Teams in EPL

Resources	Actions ▾
▼ /	
▼ /fixtures	GET OPTIONS
▼ /livescores	GET OPTIONS
▼ /predictions	GET OPTIONS
▼ /recommend	OPTIONS POST
▼ /results	GET OPTIONS
▼ /standings	GET OPTIONS
/userprofile	

The application will call these APIs only when required to.

### AWS Managed Kafka

Amazon Managed Streaming for Kafka (Amazon MSK), a fully managed service that made it easy for build and run Foot365 app that used Apache Kafka to process streaming data. Amazon MSK provided us the control-plane operations and lets you use Apache Kafka data-plane operations, such as those for producing and consuming live score data.

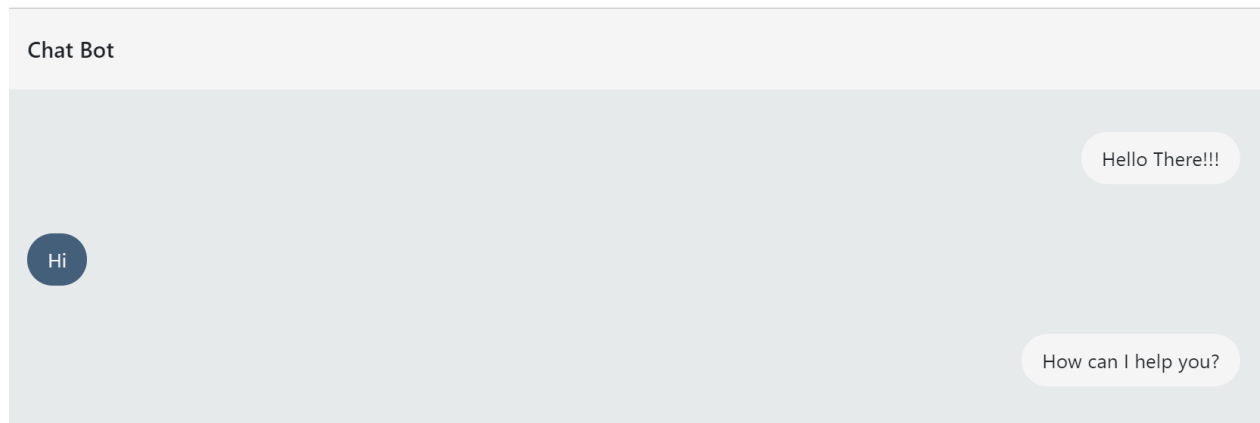


LIVE SCORE				
H1	Everton	398668	Arsenal	H0
Last Updated 1558122173972				
H2	Manchester United	400925	West Ham United	H1
Last Updated 1558122173976				
H1	Manchester City	403899	Tottenham Hotspur	H0
Last Updated 1558122173979				
H0	Watford	405092	Arsenal	H1
Last Updated 1558122173977				

## Lex

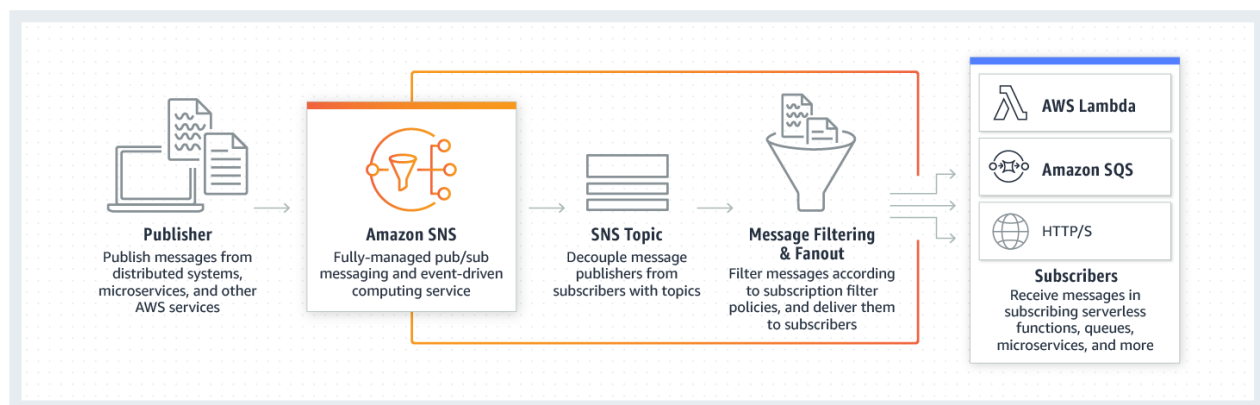
We used Amazon Lex for building conversational interfaces into Foot365 app using voice and text. Amazon Lex provided the advanced deep learning functionalities of automatic speech recognition (ASR) for converting speech to text, and natural language understanding (NLU) to recognize the intent of the text, to enabled us to build applications with highly engaging user experiences and lifelike conversational interactions further providing the fixtures, match results and recommendations of places, where the match is being screened to the user.





## SNS

We used Amazon Simple Notification Service (SNS) for a highly available, durable, secure, fully managed pub/sub messaging service that enabled us to decouple microservices, distributed systems, and serverless applications. Amazon SNS provided topics for high-throughput, push-based, many-to-many messaging. Using Amazon SNS topics, we could fan out messages to a large number of subscriber endpoints for parallel processing, including Amazon SQS queues, AWS Lambda functions, and HTTP/S webhooks and sending the match results, fixtures and recommendations to the user by integrating it with Lex.



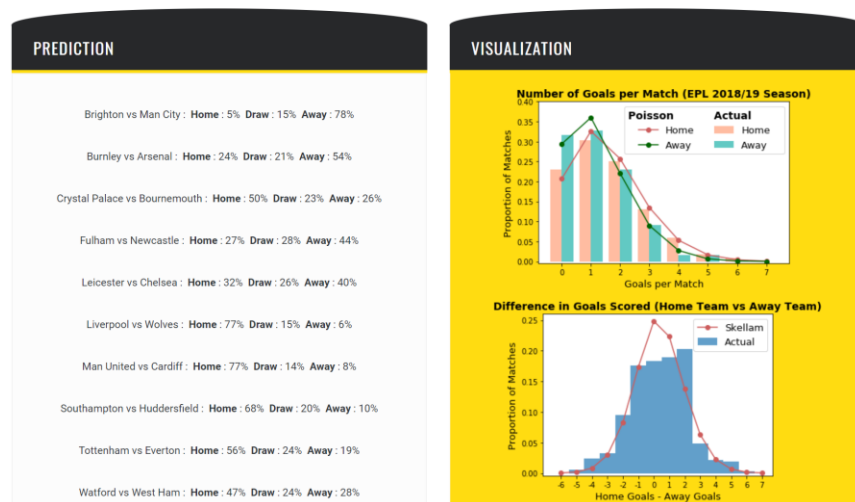
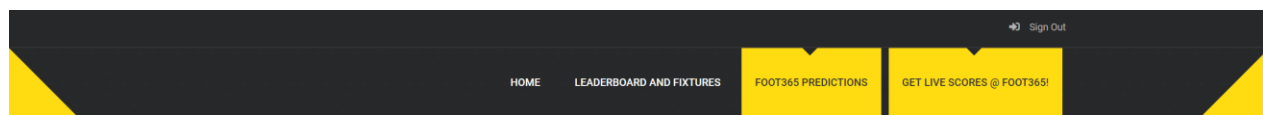
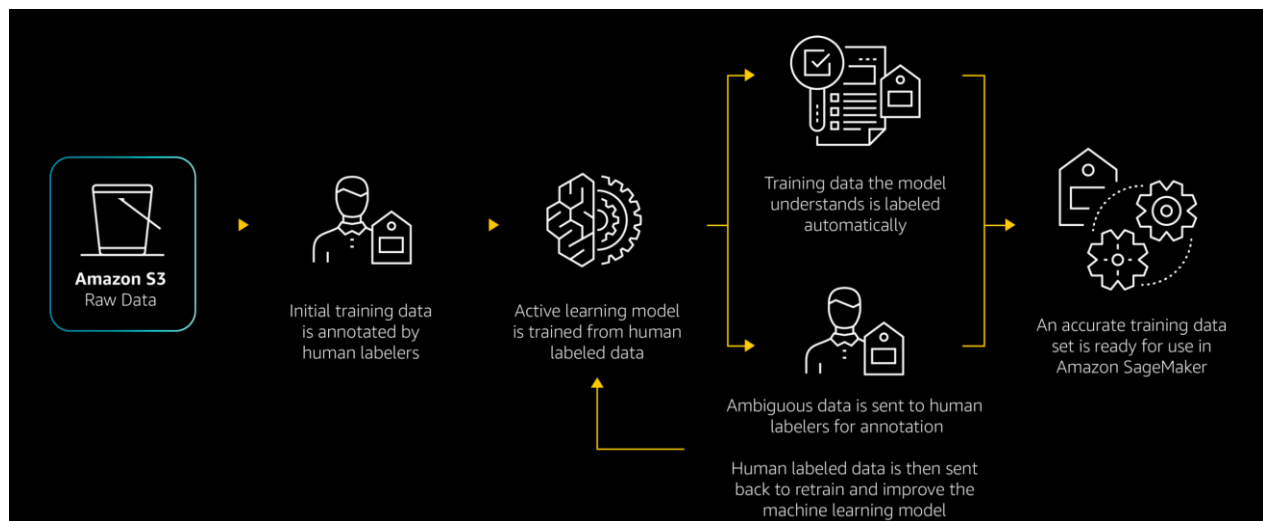
## DynamoDB

We used Amazon DynamoDB for a NoSQL database as it supports key-value and document data models, and enables developers to build modern, serverless applications that can start small and scale globally to support petabytes of data and tens of millions of read and write requests per second. We store the match results, stats and fixtures in the dynamo DB further connecting it with Elastic Search.

## AWS SageMaker

We used Amazon SageMaker to build, train, and deploy machine learning models quickly. Amazon SageMaker provided us a fully-managed service that covers the entire machine learning

workflow to label and prepare your data, choose an algorithm, train the model, tune and optimize it for deployment, make predictions of probability of match draws and win, and take action.



## Challenges Faced

- Security and Session Handling for multiple pages
- Simulating live stream logic for Kafka
- Machine Learning based complexities

- Integration with cloud and merging back-end and front-end logic.
- Data Scraping to usable formats within AWS
- Adhering to AWS dependencies
- Sticking to 100\$ limit with Student Plan.

## **7. Future Work**

- Expanding the domain to other Football leagues across the world.
- Incorporating fantasy football games for the userbase with Machine Learning techniques.
- Including Live Commentary Feature for the ongoing games.
- Incorporating FIFA World Cup matches
- Exporting our services to games such as EA and PES.