

METAZEE

A wide-angle, low-perspective shot of a soccer stadium at night. The pitch is a vibrant green, with white lines marking the field. In the center foreground, a black and white soccer ball sits on the grass. In the background, a goal is visible. The stadium stands are filled with spectators, and the sky is dark with some clouds. The overall atmosphere is dramatic and high-energy.

IS 635 - TECHNOLOGY AND START-UPS

TABLE OF CONTENTS

1. Business/Company Overview
2. Business Model Canvas
3. Value Proposition Canvas
4. Hypotheses
5. Methods
6. Monetization
7. Product Prototype



METAZEE

B E T T H E G L O B E

MetaZee provides a digital collectibles platform based on the Zee blockchain

MetaZee Founders



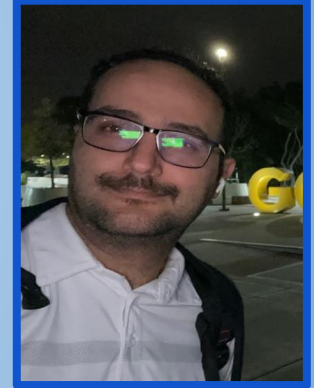
Alaba Adewunmi



Aigiun Guseinova



Chakrapani Suresh



Alireza Eftekhari

MetaZee HQ is located in London, England, United Kingdom

Funding Type: Venture



Joining Meta experience with Zee (our own currency)



- integrating Metaverse technology to create a more fascinating experience for our fans to experience the game;
- aims to revolutionize the gaming and financial services industry;
- quick transactions without using costly intermediaries

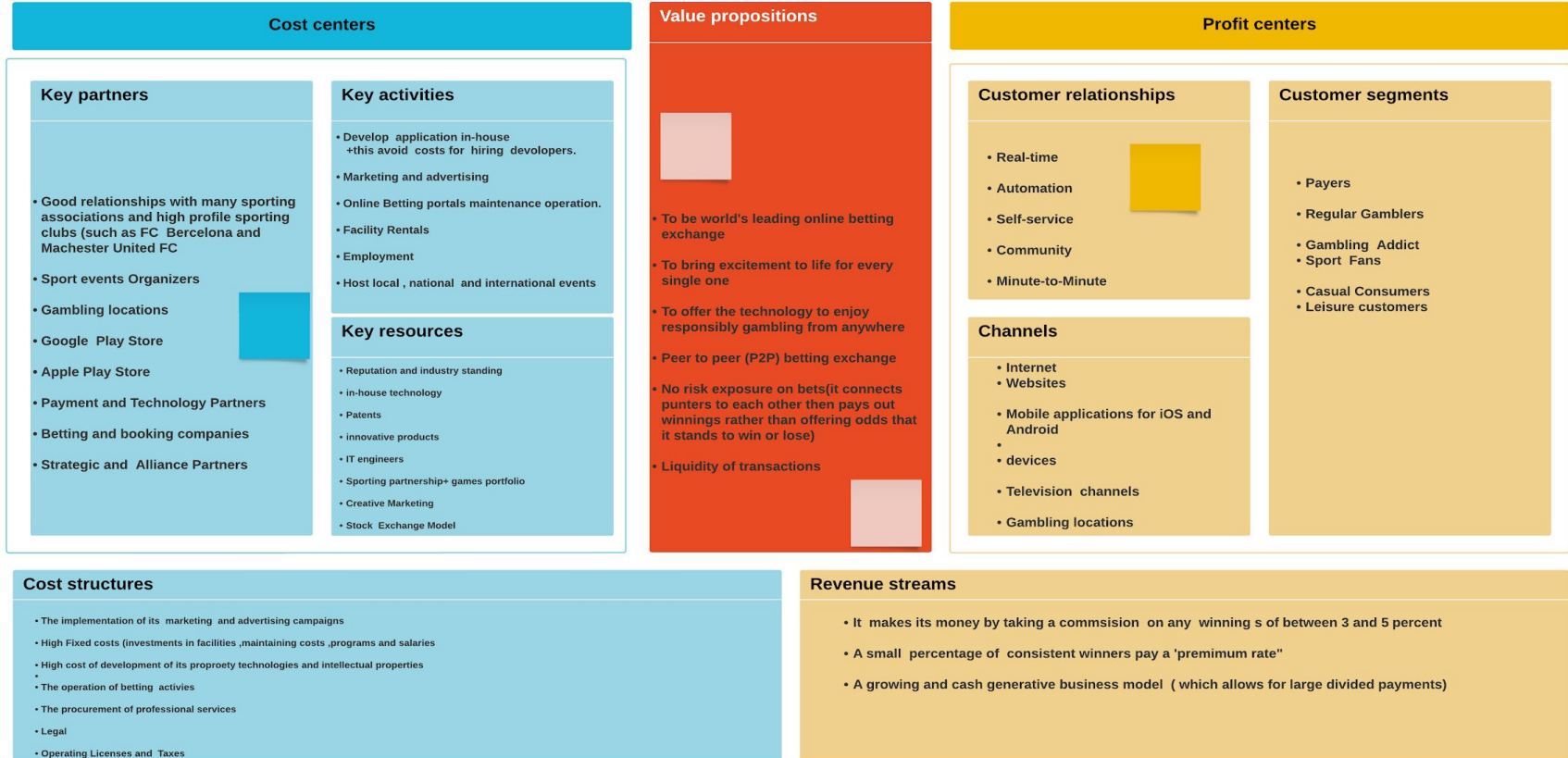


MetaZee Service Advantages



- complete anonymity: a customer does not need to provide any information to open an account and deposit crypto, as well as when you withdraw your funds;
- tools for real-time transaction data;
- crypto information, and analysis;
- providing investors and developers with an easy and safe way to buy, sell and store cryptocurrency;

Business Canvas Model in LucidSpark



Value Proposition Model for MetaZee

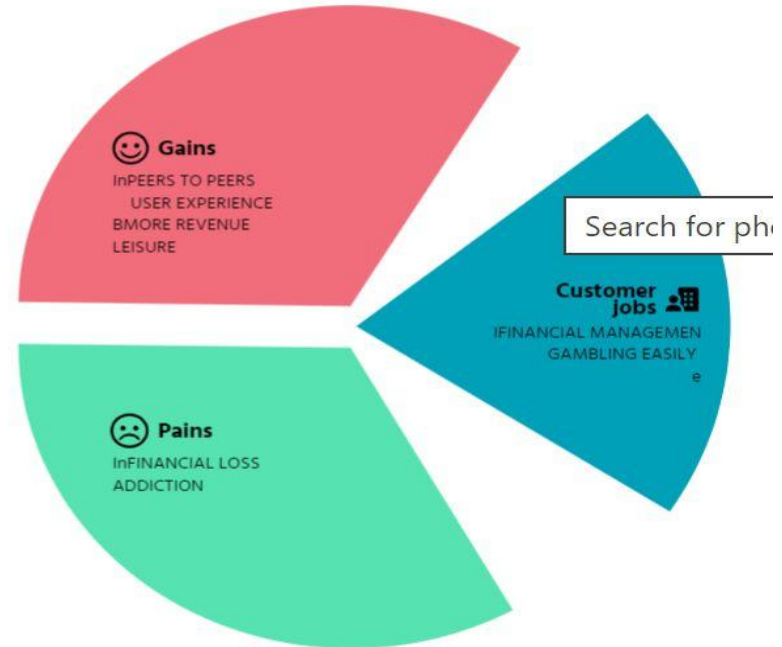
VALUE PROPOSITION CANVAS

METAZEE

VALUE PROPOSITION



CUSTOMER SEGMENT

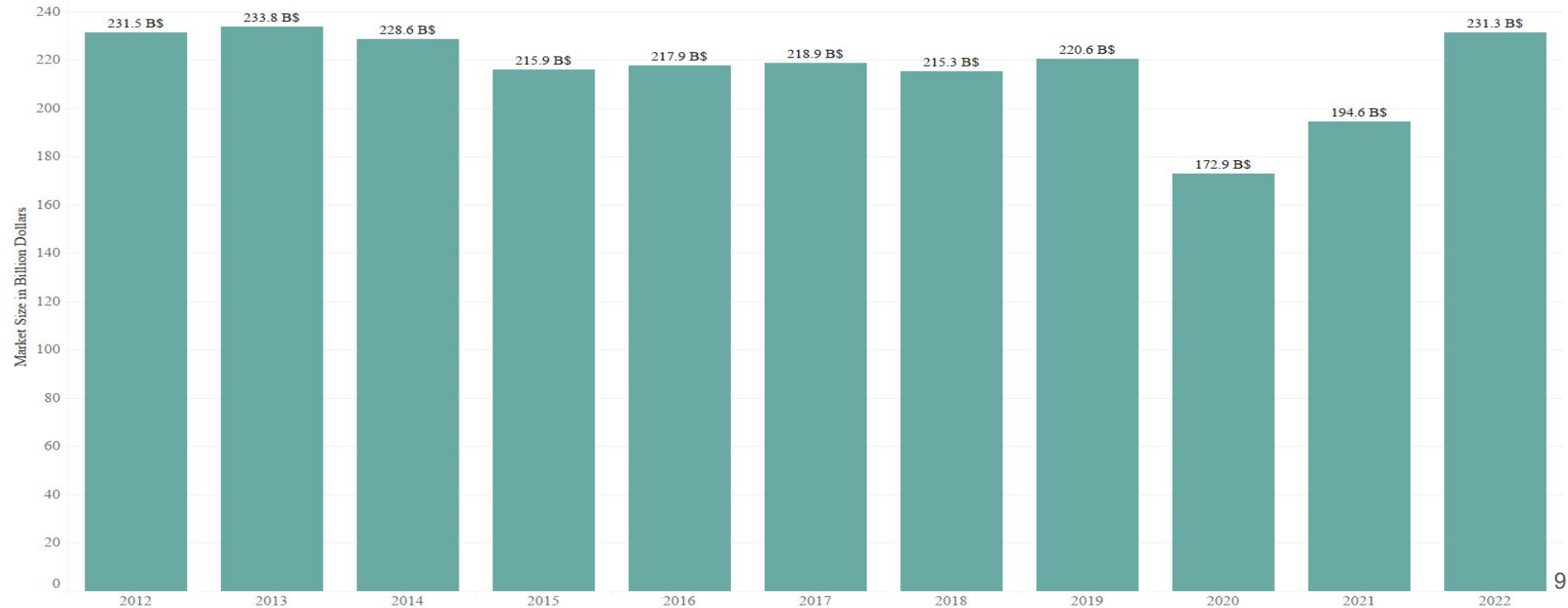


Key industry data on the sports betting sector worldwide as of June 2022



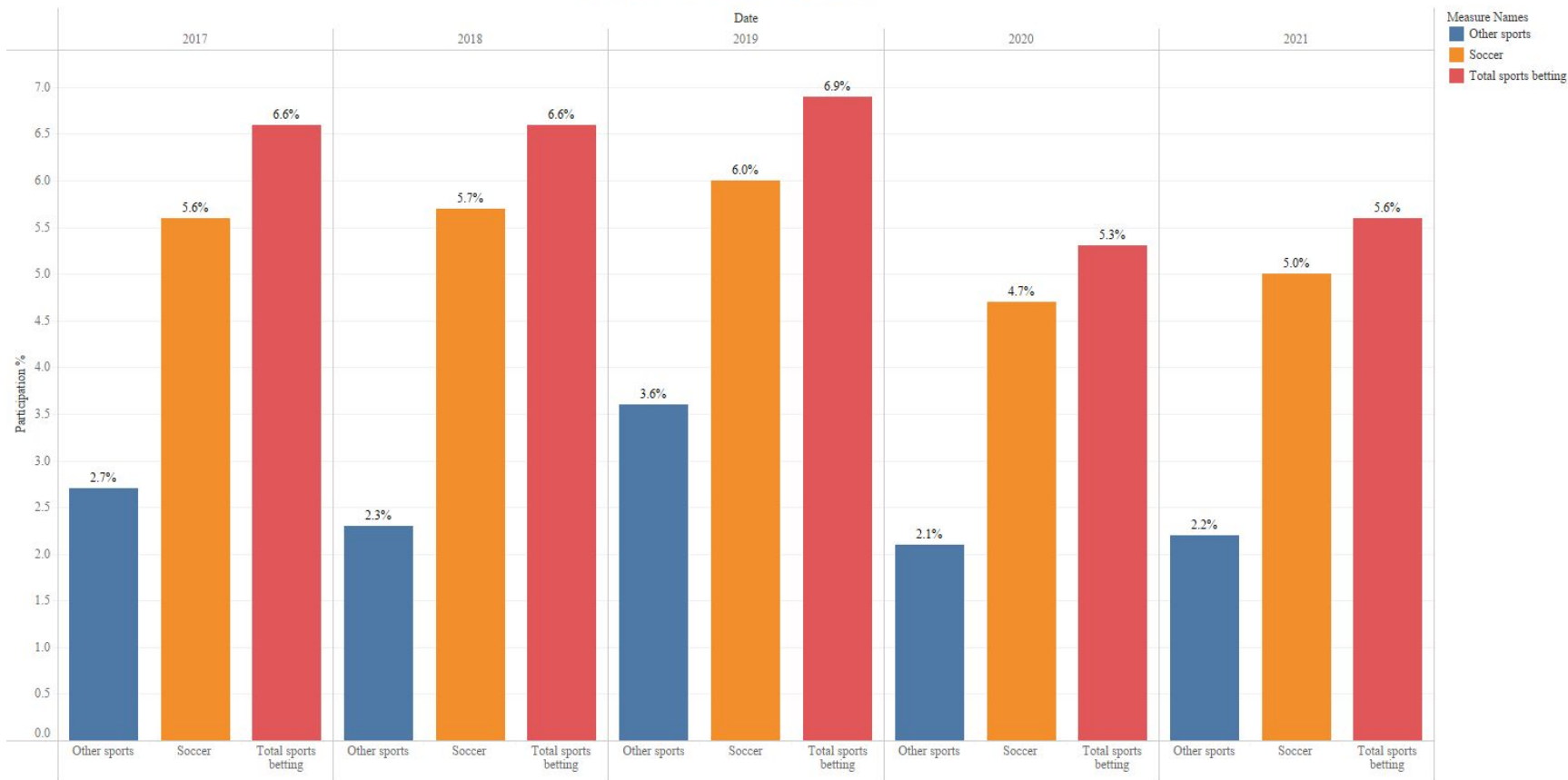
Key data on the global sports betting sector 2022

	2022
Market size (in billion U.S. dollars)	231
Employment	244705
Businesses	25240



Tracking Participation by Years in Tableau

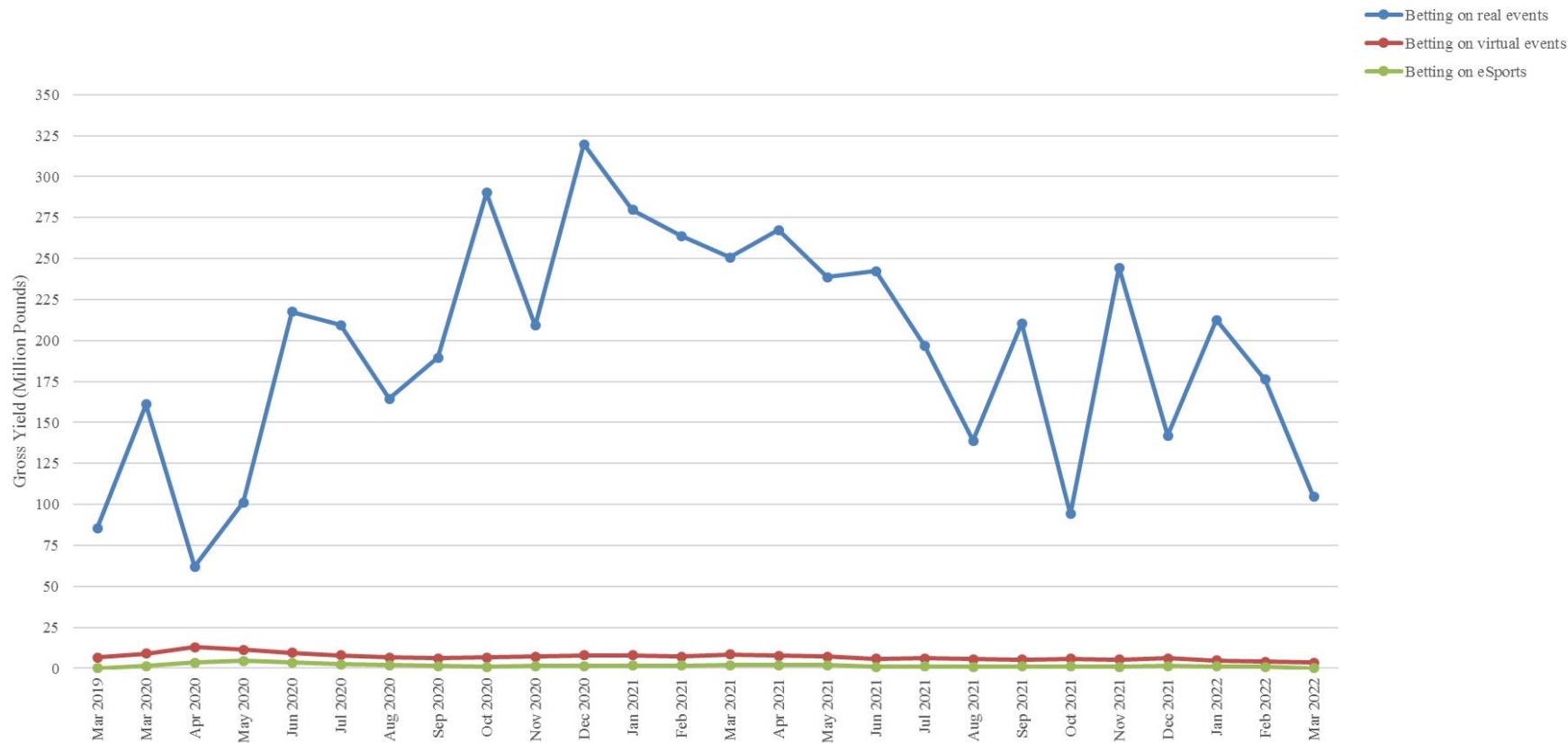
Sports Betting Participation in England Soccer vs Other Sports



Other sports, Soccer and Total sports betting for each Date Year. Color shows details about Other sports, Soccer and Total sports betting.

Monthly Market size in online betting (England)

Gross Yield (in Billion British Pounds) of the online betting market in England



Match prediction results in MetaZee

Dataset:

4506 English Premier League Matches in the past 12 years (2010-2022)

21 variables including:

Date = Match Date (dd/mm/yy)

HomeTeam = Home Team

AwayTeam = Away Team

FTHG = Full Time Home Team Goals

FTAG = Full Time Away Team Goals

FTR = Full Time Result (H=Home Win, D=Draw, A=Away Win)

HTHG = Half Time Home Team Goals

HTAG = Half Time Away Team Goals

HTR = Half Time Result (H=Home Win, D=Draw, A=Away Win)

HS = Home Team Shots

AS = Away Team Shots

HST = Home Team Shots on Target

AST = Away Team Shots on Target

HF = Home Team Fouls Committed

AF = Away Team Fouls Committed

HC = Home Team Corners

AC = Away Team Corners

HY = Home Team Yellow Cards

AY = Away Team Yellow Cards

HR = Home Team Red Cards

AR = Away Team Red Cards

Match prediction results in MetaZee

Random Forest model in Python

```
matches_rolling.index = range(matches_rolling.shape[0])
```

[112]

```
def make_predictions(data, predictors):  
    train = data[data["Date"] < '2022-01-01']  
    test = data[data["Date"] > '2022-01-01']  
    rf.fit(train[predictors], train["Target"])  
    preds = rf.predict(test[predictors])  
    combined = pd.DataFrame(dict(actual=test["Target"], predicted=preds), index=test.index)  
    precision = precision_score(test["Target"], preds)  
    return combined, precision
```

[113]

```
combined, precision = make_predictions(matches_rolling, predictors + new_cols)  
precision
```

[114]

... 0.640625

Match prediction results in MetaZee XGBoost model in Amazon Sagemaker

```
[ ] from sagemaker.serializers import CSVSerializer
```

```
test_data_array = test_data.drop(['FTR_D', 'FTR_H'], axis=1).values #load the data into an array
xgb_predictor.serializer = CSVSerializer() # set the serializer type
predictions = xgb_predictor.predict(test_data_array).decode('utf-8') # predict!
predictions_array = np.fromstring(predictions[1:], sep=',') # and turn the prediction into an array
print(predictions_array.shape)
```

```
(1352,)
```

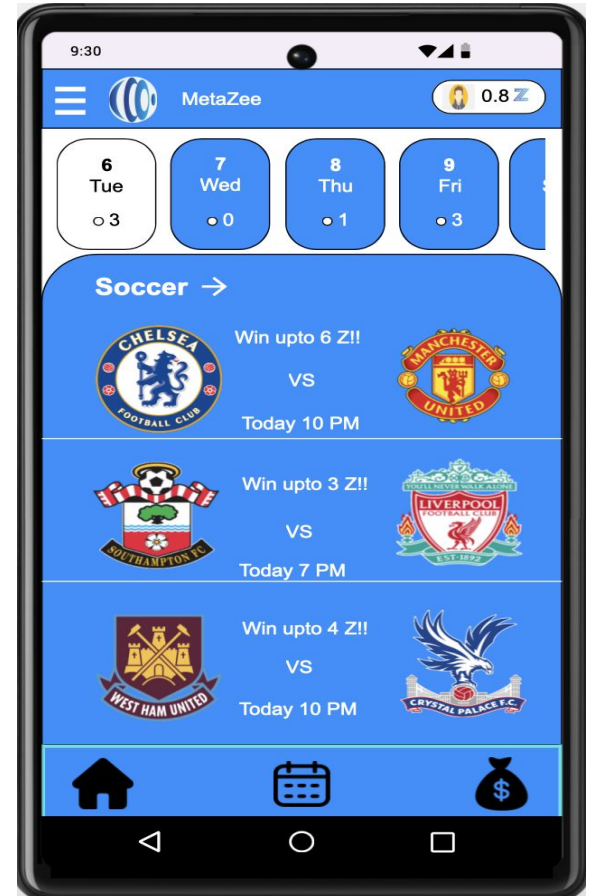
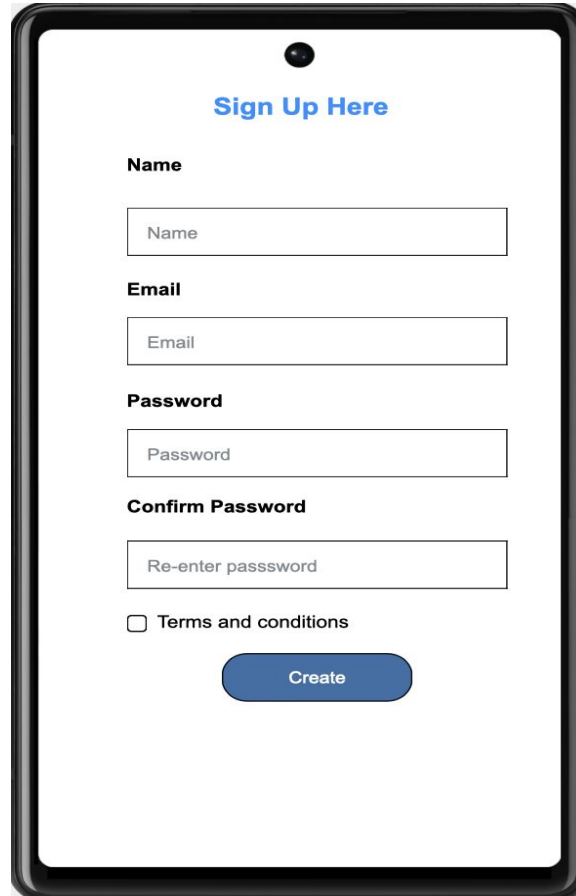
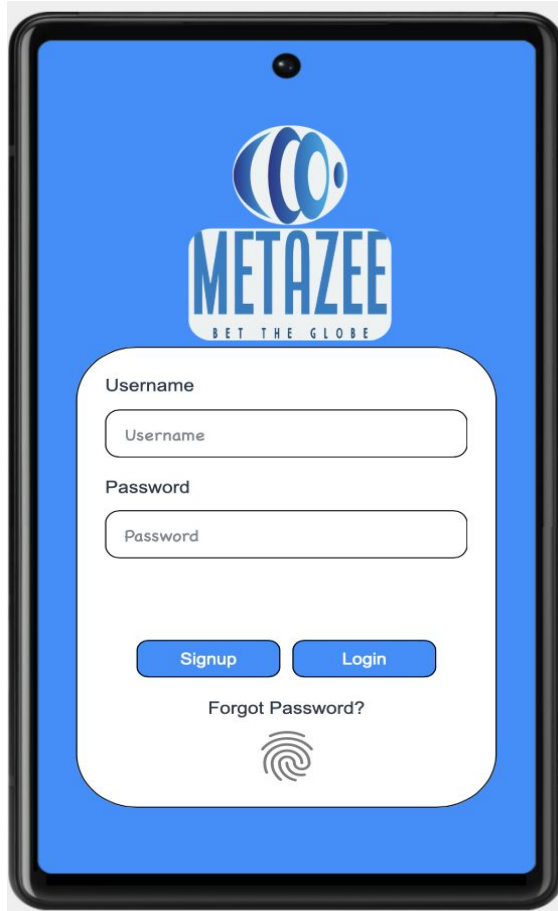
```
cm = pd.crosstab(index=test_data['FTR_H'], columns=np.round(predictions_array), rownames=['Observed'], colnames=['Predicted'])
tn = cm.iloc[0,0]; fn = cm.iloc[1,0]; tp = cm.iloc[1,1]; fp = cm.iloc[0,1]; p = (tp+tn)/(tp+tn+fp+fn)*100
print("\n{0:<20}{1:<4.1f}%\n".format("Overall Classification Rate: ", p))
print("{0:<15}{1:<15}{2:>8}".format("Predicted", "win", "not win"))
print("Observed")
print("{0:<15}{1:<2.0f}% ({2:<}){3:>6.0f}% ({4:<})".format("win", tn/(tn+fn)*100,tn, fp/(tp+fp)*100, fp))
print("{0:<16}{1:<1.0f}% ({2:<}){3:>7.0f}% ({4:<}) \n".format("not win", fn/(tn+fn)*100,fn, tp/(tp+fp)*100, tp))
```



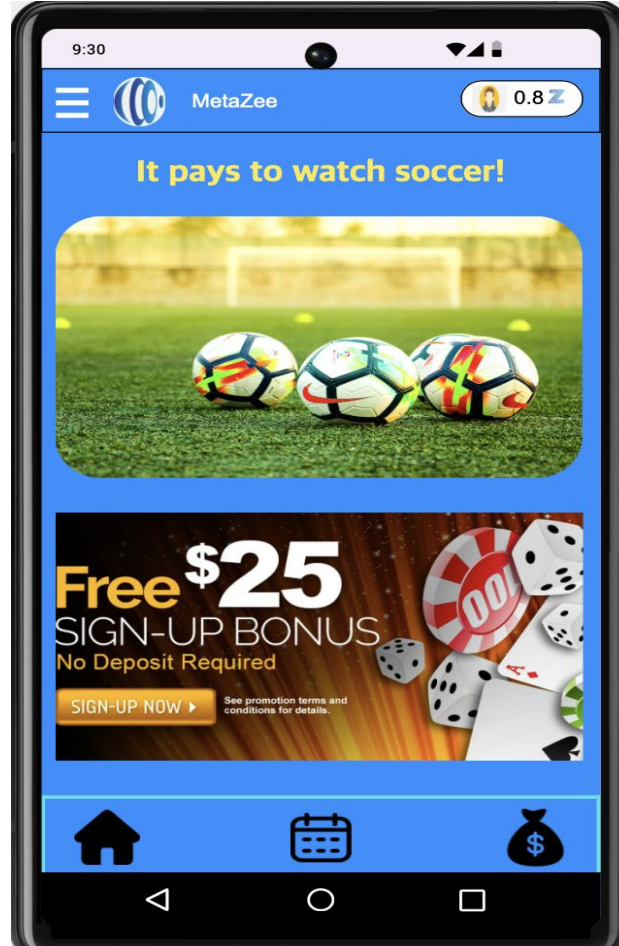
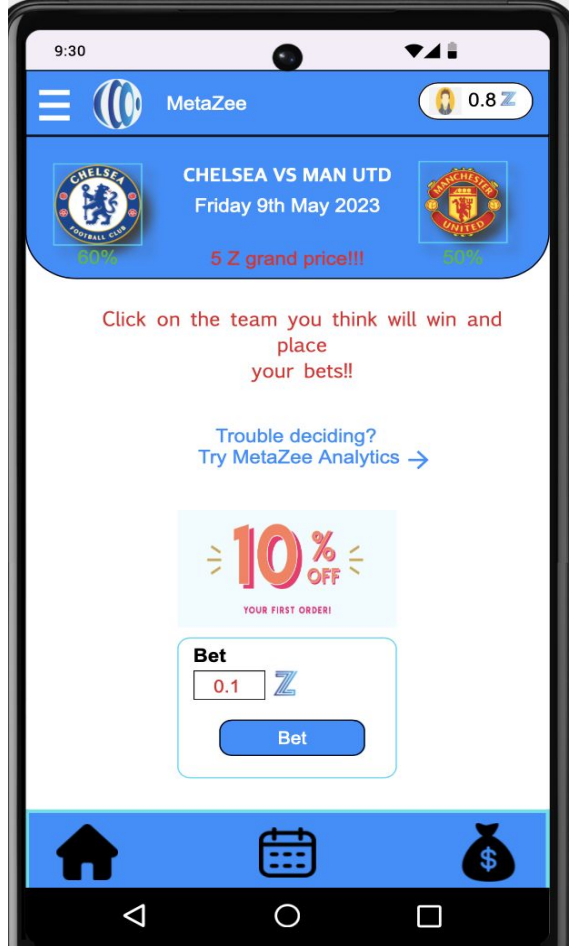
Overall Classification Rate: 84.1%

Predicted	win	not win
Observed		
win	85% (637)	17% (100)
not win	15% (115)	83% (500)

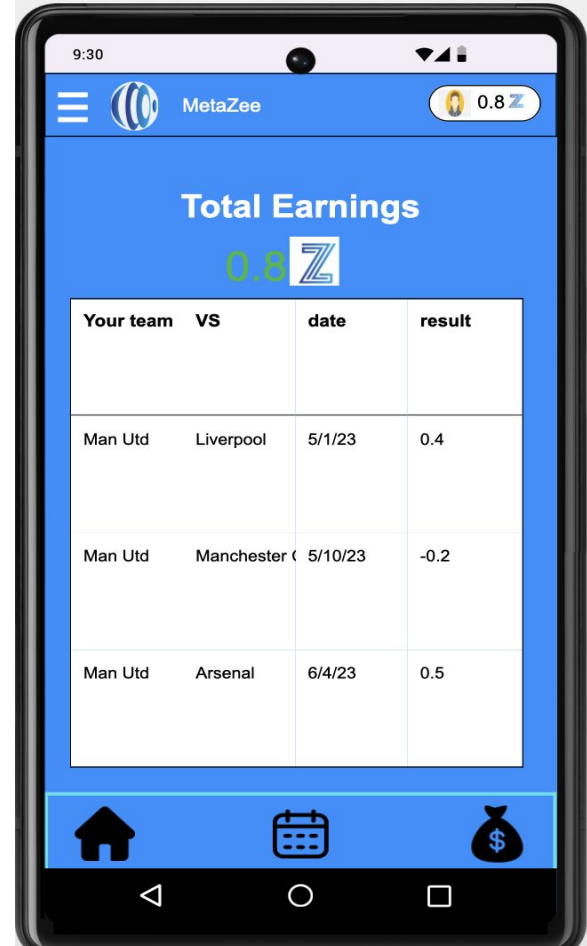
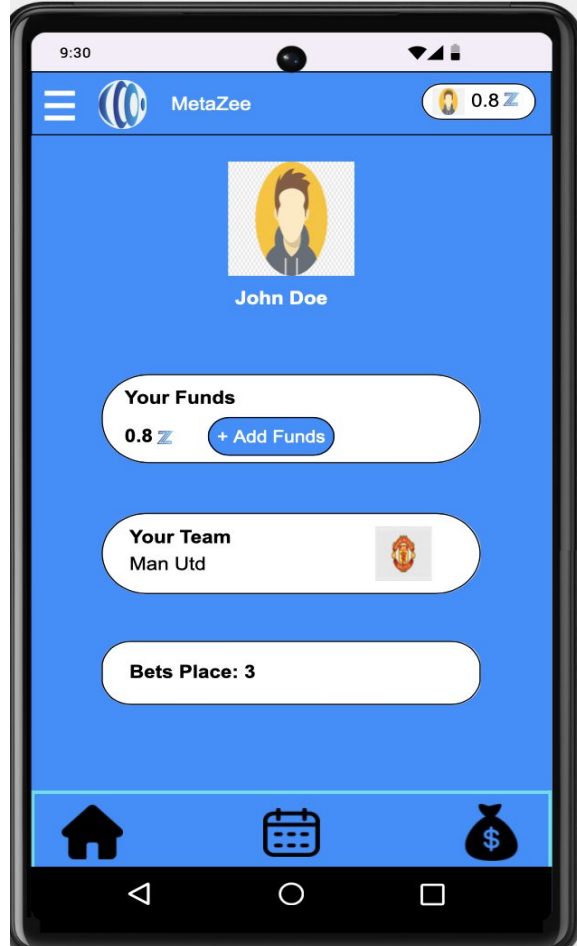
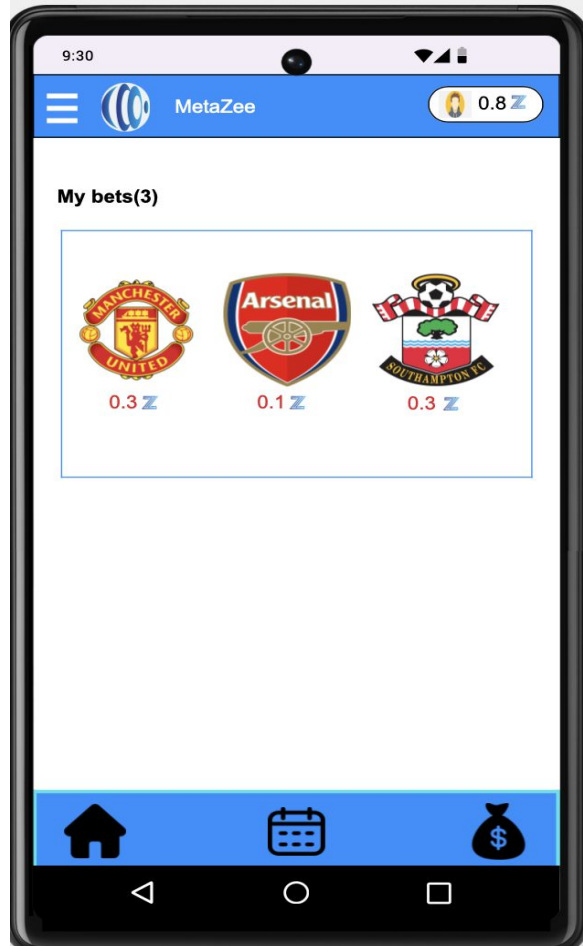
Product Prototype designed in Justinmind



Product Prototype designed in Justinmind



Product Prototype designed in Justinmind



Product Prototype designed in Justinmind



Product Prototype
[Hyperlink](#)

THANK YOU FOR YOUR ATTENTION!

