

Florida International University
Knight Foundation School of Computing and Information Sciences

Capstone Project Documentation

Project Title: TapIn MMA

Team Members: Christian Garcia, Kevin Ferrer, Bruno Gouveia

Product Owner/Team lead: Christian Garcia

Instructor/Faculty: Masoud Sadjadj, Florida International University (FIU)

Executive Summary (Abstract)

TapIn MMA is a web application created to deliver fast, data-driven insights into Mixed Martial Arts. Fans, analysts, and casual viewers often struggle with scattered fighter information across multiple websites and APIs. TapIn MMA solves this by aggregating fighter profiles, fight histories, events, and statistical data into one streamlined platform.

The system integrates the Octagon API, Tapology API (via RapidAPI), The Odds API, Mailjet API, and Nominatim geocoding to generate fighter profiles, matchup data, event cards, odds, and gym geolocation. Users can filter fighters, compare them head-to-head, explore statistics through visual charts, view upcoming fight cards, and use a prediction model that provides a simple, rules-based win probability.

Built using Streamlit, Python, Plotly, Pandas, and Folium, TapIn MMA prioritizes usability and speed. Its clean layout, responsive UI, and interactive visualizations make navigating MMA data intuitive and engaging. The application demonstrates strong software engineering practices through requirements analysis, modular design, testing procedures, and user-centered functionality.

Table of Contents

Executive Summary (Abstract)	2
Problem & Requirements	4
System Overview & Architecture	5
Purpose of New System	6
User Stories	7
Project Plan	10
Hardware and Software Requirements	11
Sprints Plan	12
System Design	15
Architectural Patterns	18
System and Subsystem Decomposition	20
Design Patterns	22
System Validation	23
Ethical, Legal, Societal Impact	25
Implementation Notes	26
Limitations & Future Work	27
Appendix A – Diagrams	28
Appendix B – Screenshots	29
Appendix C – Sprint Review Reports	32
References	33

Problem & Requirements

Accessing accurate and organized MMA data is often difficult due to fragmentation across websites, forums, social media pages, and independent databases. Fans searching for fighter statistics, fight history, performance analytics, or upcoming events frequently switch between platforms to compile basic information. Even existing paid MMA analytics tools provide incomplete or cluttered interfaces.

TapIn MMA addresses this problem by providing a centralized MMA data platform. The application offers fighter search and filtering, detailed fighter profiles, win/loss metrics, interactive statistics, head to head comparisons, event listings, geolocation of gyms, and odds integration. It also includes a lightweight fight-prediction model that uses rule based logic.

The goal of TapIn MMA is to simplify how fans engage with MMA data by creating a clean, accurate, and interactive MMA dashboard that is accessible to all users.

System Overview & Architecture

Before TapIn MMA, users depended on inconsistent data sources such as Tapology, UFC.com, ESPN, Sherdog, Reddit, and various API-driven sites. These platforms often require manual searching, provide incomplete data, or are cluttered with ads and navigation friction.

Key issues with the current MMA information ecosystem:

- No single platform consolidates fighter stats, fight history, and events.

Many sites lack API access or restrict data behind paywalls.

- Fighter data is inconsistent between different sources.
- Users cannot easily compare fighters side-by-side.
- There are few available fight predictive algorithms available.
- Existing dashboards do not offer prediction logic or interactive visualizations.
- No integrated gym geolocation tools exist for MMA.

As a result, the typical user experience is fragmented and time consuming.

Purpose of New System

TapIn MMA aims to create a unified, reliable, and interactive MMA data dashboard that:

- Aggregates fighter data from multiple APIs.
- Provides fast search and filtering.
- Displays fighter profiles with detailed stats.
- Allows head-to-head comparisons.
- Visualizes fight metrics through charts.
- Shows upcoming events and fight cards.
- Displays event locations via geocoding.
- Predicts fight outcomes using weighted heuristic models.
- Offers optional betting odds from third-party providers.
- Send notifications using Mailjet.

The system provides an intuitive experience for MMA fans, analysts, bettors, and beginners who want an accessible, data-driven overview.

User Stories

The following user stories represent the completed functionality of the TapInMMA application. Each story includes the subtasks that were carried out during implementation. All listed stories were fully completed during development.

1. Betting Info

As a user, I want to see betting odds for upcoming fights so that I can understand how the market views each matchup.

Subtasks Completed:

- Researched and selected a reliable betting odds API
- Set up the API connection and tested sample requests
- Created data structures to store and organize odds
- Displayed odds directly on fight cards in the events tab
- Added a “Last Updated” indicator to avoid confusion with stale data
- Implemented a fallback message when the odds API is unavailable
- Converted provided odds into implied probabilities for clarity

2. Fight Predicting Software / Algorithm

As a user, I want the site to provide a fight prediction with a confidence score so that I can get insights beyond raw statistics.

Subtasks Completed:

- Decided which fighter data would be used in predictions
- Built feature extraction methods (reach, age, record, recent streaks, etc.)
- Implemented a simple rule-based prediction model
- Added a prediction function that returns both predicted winner and confidence
- Displayed the “TapIn Pick” prediction inside the comparison tool
- Added a toggle to enable or disable predictions

- Added a brief disclaimer explaining prediction limitations

3. Email Notifications for Upcoming Events

As a user, I want to receive email notifications about upcoming events so that I don't miss fights I want to watch or attend.

Subtasks Completed:

- Chose and configured an email provider (Mailjet / SMTP)
- Added an email signup section with proper consent language
- Created reusable email templates for event notifications
- Built the process to automatically send notifications when new events are listed
- Implemented unsubscribe logic
- Added filtering options to send notifications based on user preferences (fighters or weight classes)
- Added retry logic and error logging for failed email sends

4. Background Themes

As a user, I want to choose between different background themes so that I can personalize the look and feel of the site.

Subtasks Completed:

- Designed and implemented light mode and dark mode
- Added a theme toggle switch to the interface
- Implemented saving of user preferences (using Streamlit state)
- Applied themes across all main pages and UI elements
- Added several MMA-inspired theme options
- Performed color visibility and contrast testing to ensure usability

5. Regional Event Selector

As a user, I want to select a geographic region to see upcoming MMA events so that I can stay informed about fights near me.

Work Completed:

- Finalized region dropdown selection
- Connected filtered region queries to the event API
- Improved filtering behavior and handling of empty results

6. Event Details Display

As a user, I want to view event details like the main event, weight class, city, and broadcast information so that I can plan to watch or attend.

Work Completed:

- Connected all event data fields to the backend
- Displayed complete fight card information in a clean, readable layout
- Finalized mobile and desktop responsive layout design

Pending User Stories

There are no remaining user stories pending completion.
All stories planned for development were completed successfully.

Project Plan

The development of TapIn MMA followed an iterative agile methodology. Work was divided into sprints to ensure features were designed, implemented, tested, and reviewed systematically.

Sprints included:

- Fighter Search & Filter Development
- Comparison Tool
- Statistics Engine & Visualizations
- Upcoming Events Integration
- Prediction Model Integration
- Betting Odds Integration
- Final UI/UX refinement and performance testing

Hardware & Software Requirements

Software

- Python 3.10+
- Streamlit
- Pandas
- Plotly
- Requests
- RapidAPI / Octagon API access
- Nominatim geocoding
- Folium
- Mailjet or Gmail SMTP
- GitHub

Hardware

- Laptop or desktop with at least:
 - 4 GB RAM
 - 2 CPU cores
 - Stable internet connection

Sprints Plan

This section summarizes the sprint-by-sprint development cycle of TapIn MMA, aligned with the structure presented in the team's final presentation. Each sprint reflects iterative progress toward core features, API integration, UI development, and final deliverables.

Sprint 1:

Sprint Goal

Establish foundational research and set up the initial structure of the application.

Completed Tasks

- Conducted background research on MMA data sources and APIs.
- Identified feasibility constraints and available free-tier services.
- Began early-stage development and set up the core project structure.

Sprint Review

This sprint established the project direction and prepared the environment needed for future development.

Sprint 2:

Sprint Goal

Begin functional UI development and connect early frontend components with backend logic.

Completed Tasks

- Continued front-end development including core layout and initial tab structure.
- Established communication between UI components and backend helper functions.
- Began developing the first version of the predictive algorithm.

Sprint Review

UI and backend interactions were successfully prototyped, enabling more advanced features in later sprints.

Sprint 3:

Sprint Goal

Expand functionality and integrate external APIs into the system.

Completed Tasks

- Completed most front-end elements.
- Integrated APIs for:
 - Betting Information
 - Fighter Comparison
 - Upcoming Events
- Completed at least one full API implementation end-to-end.
- Developed a prototype version of the prediction algorithm.

Sprint Review

This sprint solidified the system's backbone, enabling real data to flow through the UI for the first time.

Sprint 4:**Sprint Goal**

Achieve full baseline functionality and introduce enhancements beyond core requirements.

Completed Tasks

- Reached baseline functionality for all major system features.
- Added optional enhancements, including:
 - Newsletter/email notifications
 - Background themes for UI customization
- Continued refining the prediction algorithm for improved accuracy.

Sprint Review

This sprint ensured all required features were operational while adding valuable extra features.

Sprint 5:**Sprint Goal**

Strengthen system reliability, performance, and user experience.

Completed Tasks

- Performed extensive testing across all tabs and APIs.
- Fixed issues, refined UI layouts, and optimized API call efficiency.
- Completed full refinement and testing of the predictive algorithm.

Sprint Review

The system became stable, efficient, and ready for final polishing.

Sprint 6:**Sprint Goal**

Finalize the entire application and prepare all deliverables.

Completed Tasks

- Finalized all system features and ensured stability.
- Completed documentation, poster materials, and presentation preparation.
- Polished UI elements and performed end-to-end validation.
- Conducted final stability checks before release.

Sprint Review

The project reached full completion, meeting all capstone requirements and preparing the system for demonstration.

System Design

This section explains how TapIn MMA is organized internally and how its main components interact. Although the entire project runs inside one Streamlit file, the code is structured into clear subsystems that separate UI, logic, data handling, and background tasks.

Overall Architecture

TapIn MMA follows a simple layered architecture:

- **Presentation Layer:**
Streamlit UI, tabs, forms, charts, and customization controls.
- **Application Logic Layer:**
Helper functions for filtering fighters, computing predictions, converting betting odds, formatting values, and managing themes.
- **Data Integration Layer:**
Adapters for the Octagon API, The Odds API, Tapology (RapidAPI), Nominatim geocoding, and Mailjet email delivery.
- **Background & Persistence Layer:**
APScheduler for optional daily digests and a CSV file storing subscriber emails.

Streamlit handles reactive reruns on every user interaction, and cached functions keep data loading fast.

1. Presentation Layer (UI Tabs)

The interface is organized into five tabs:

- **Fighter Comparison** – Choose weight class and compare two fighters.
- **Upcoming Events** – Browse events by region and view a simple geo-map.
- **Betting Info** – Display odds and implied probabilities.
- **Fight Predictor** – Run the rules-based scoring model.
- **Settings** – Choose themes, background colors, and access admin tools.

Key UI features:

- A disclaimer gate using session state.
- Selectboxes, toggles, expanders, charts, and admin-only controls.
- Simple, responsive layouts using Streamlit columns.

2. Application Logic Layer

Core logic sits behind the UI tabs:

- Themes: A small THEMES dictionary and CSS injection apply user-selected styles.
- Fighter Comparison: Filters fighters by weight class, shows stats, images, and a Plotly record chart.
- Prediction Model: A rule-based scoring system uses wins, losses, reach, height, and age to output a confidence percentage.
- Odds & Probability: Converts American odds to implied probability and warns if data is stale.
- Email Logic: Handles subscription, unsubscription, and digest preview formatting.

Session state controls acceptance of terms, theme selection, and admin authentication.

3. Data Integration Layer (APIs + Caching)

TapIn MMA integrates several external services through dedicated helper functions, ensuring efficient data retrieval, caching, and standardized responses across all modules.

- Octagon API: Loads fighter data and caches it for one hour.
- The Odds API: Fetches American odds and normalizes them into a Pandas DataFrame.
- Tapology (RapidAPI): Returns event data by region.
- Nominatim: Converts event city names into latitude/longitude for mapping.
- Mailjet: Sends notification emails using the v3.1 API.

All API functions use `try/except` blocks and provide fallbacks if calls fail.

Caching (`@st.cache_data`) reduces API usage and keeps the app responsive.

4. Background & Persistence Layer

- CSV Storage: Email subscribers are stored in a simple CSV file.
- Background Scheduler: If APScheduler is available, the app creates a daily 9 AM job to fetch events and send digests automatically.
- If scheduling fails or APScheduler is missing, the app simply continues without background jobs.

Component Interaction Summary

1. User accepts the disclaimer → tabs unlock.
2. Streamlit reruns the app on every interaction but returns cached data instantly.
3. UI calls helper functions (fighters, events, odds).
4. Logic processes the data (probabilities, prediction scores, geocoding).
5. Output is rendered through charts, tables, and interactive controls.
6. Email features allow subscribing/unsubscribing and (for admins) sending digests.

Design Goals Achieved

- Clear separation of concerns between UI, logic, and data handling.
- Graceful failure for all external API calls.
- Fast performance via caching and lightweight data processing.
- Easy extensibility for future ML models or additional tabs.
- Simple deployment with no full backend or database required.

Architectural Patterns

TapIn MMA applies several lightweight architectural patterns to keep the system organized, maintainable, and stable despite relying on multiple external APIs.

1. Layered Pattern (UI → Logic → Data)

The app uses clear separation between:

- **Presentation Layer:** Streamlit UI components, tabs, forms, charts.
- **Application Logic Layer:** helper functions for scoring, conversions, formatting, filtering.
- **Data Integration Layer:** API calls to Octagon, Tapology (RapidAPI), The Odds API, Nominatim, and Mailjet.
- **Persistence Layer:** CSV based email list + APScheduler background jobs.

This structure keeps UI code simple while isolating business logic and external data handling.

2. Adapter Pattern (API Wrappers)

Functions such as

`fetch_upcoming_events()`, `fetch_odds_from_api()`, `load_octagon_fighters()`, and `send_email_mailjet()`

act as adapters that convert inconsistent API responses into standardized Python structures. This allows the UI to stay consistent even if external APIs change formatting.

3. Caching Pattern

Streamlit's `@st.cache_data` decorator caches results for:

- fighter lists
- event schedules
- geocoding
- betting odds

Caching reduces API calls, avoids rate-limit failures, and speeds up page loads.

4. Reactive Pattern

Streamlit reruns the script whenever:

- inputs change
- buttons are pressed
- theme updates
- tabs switch

This creates an observer-like behavior, where UI elements automatically react to state changes in `st.session_state`.

5. Strategy Pattern (Prediction Algorithm)

The fight prediction logic is implemented as a scoring strategy, where the system:

- extracts features
- applies weighted rules
- normalizes scores
- chooses a winner

This can be easily swapped for a machine learning model later without changing the UI.

6. Scheduler Pattern

TapIn MMA uses APScheduler to optionally send daily email digests. This is a background task pattern separate from the main UI thread.

System and Subsystem Decomposition

TapIn MMA is organized into several focused subsystems, each responsible for a clear portion of the application's functionality.

1. User Interface Subsystem

- Built entirely with Streamlit.
- Renders tabs, comparison layouts, event lists, betting tables, charts, forms, and maps.
- Uses session state to control themes, admin access, and disclaimer acceptance.

2. API Aggregation Subsystem

- Contains wrappers for all external services:
 - Octagon API (fighters)
 - Tapology via RapidAPI (events)
 - The Odds API (betting odds)
 - Nominatim (geocoding)
 - Mailjet (email delivery)
- Ensures consistent response formatting and error handling.

3. Fighter Data Processing Subsystem

- Parses fighter height, reach, age, and records.
- Standardizes data formats and converts units (e.g., height to cm).
- Supports the comparison and prediction features.

4. Comparison & Prediction Subsystem

- Computes side-by-side metrics (wins, losses, draws, reach differences, etc.).

- Runs the rule-based scoring model to produce confidence percentages and the TapIN Pick.
- Includes helpers for normalization and probability formatting.

5. Events Subsystem

- Retrieves upcoming events for selected regions.
- Cleans, formats, and prepares schedule data for both list view and mapping.
- Handles missing or partial information safely.

6. Geocoding & Map Subsystem

- Uses Nominatim to convert city names into coordinates.
- Generates map visualizations with Plotly's `scatter_geo`.
- Falls back gracefully if geocoding fails or rate-limits occur.

7. Email Notifications Subsystem

- Stores subscriber emails in a CSV file.
- Sends event digests through Mailjet.
- Supports subscribing, unsubscribing, and admin-triggered digests.
- Optional APScheduler job sends daily summaries automatically.

Design Patterns

TapIn MMA uses several lightweight patterns that match its code structure and Streamlit-based workflow:

MVC Pattern (Conceptual)

- **Model:** Data returned from APIs.
- **View:** Streamlit UI tabs and components.
- **Controller:** Helper functions handling filtering, scoring, odds logic, and formatting.

Adapter Pattern

- All API calls are abstracted into dedicated helper functions.
- Each function adapts inconsistent external API formats into unified Python structures.

Strategy Pattern (Prediction Logic)

- The fight predictor uses a scoring strategy that can be replaced later with a machine-learning strategy without touching the UI.

Factory Pattern (Charts)

- Plotly charts are created through small, reusable generation functions (bar charts, scatter_geo, etc.).

Observer Pattern (Reactive UI)

- Streamlit automatically reruns whenever input changes, mimicking an observer system reacting to state updates.

System Validation

Testing and validation were performed throughout development and finalized during the last sprint:

Unit Tests

- Checked API wrapper functions for correct formatting and safe error handling.
- Verified helper functions (height conversion, probability math, prediction scoring).

Integration Testing

- Tested end-to-end fighter comparison, event browsing, prediction output, and odds display.
- Ensured tabs worked together without state conflicts.

Performance Testing

- Validated fast load times using Streamlit's caching.
- Confirmed heavy tabs (Betting, Events) remain responsive under repeated use.

API Stress & Reliability Tests

- Checked behavior during rate limits or failed API calls.
- Ensured fallbacks and user-friendly error messages worked correctly.

Final Sprint Validation

- All major features passed:
 - Fighter comparison
 - Event region filtering
 - Betting odds display
 - Prediction tool

- Email notifications
 - Theming and admin mode
- UI was polished and refined, and the system met all functional and non-functional requirements.

Ethical, Legal, Societal Impact

Ethical Considerations

- **Prediction Accuracy:** The system provides suggested predictions but does not guarantee accuracy; users must understand the predictions are advisory only.
- **Responsible Use:** The system does not promote gambling; odds are shown for informational purposes, and the UI avoids language encouraging betting behavior.

Legal Considerations

- **Betting Data:** The project uses third-party APIs (The Odds API). These are used under educational fair-use; no real betting transactions are supported.
- **Email Privacy:** Only minimal personal information (email addresses) is stored locally. No emails are shared or sold. Users may unsubscribe at any time.
- **API Licensing:** All APIs used are either free-tier or allow educational usage. Attribution and usage guidelines are followed.

Societal Impact

- Improves user understanding of combat sports statistics.
- Demonstrates how AI-assisted insights can make data more accessible.
- Avoids harmful impacts by ensuring predictions are not presented as financial advice.

Implementation Notes

- Repository structure organized into modules (UI, data access, services); separate layers for routing, business logic, and data storage.
- Uses Node.js/Express for backend APIs and MongoDB for data persistence; frontend built with Streamlit/React and Plotly for visualization.
- Implementation follows design patterns (Observer, Adapter) and uses helper functions to call external APIs (OpenAI, Octagon, Nominatim, The Odds).
- Code emphasizes readability and maintainability: consistent naming conventions, modular functions, and error handling across modules.

Deployment & Operations

- Application is deployed on a cloud platform (e.g., Heroku or AWS) with continuous integration for seamless updates.
- Monitoring and logging are implemented to track uptime and errors; maintenance tasks include database backups and performance tuning.
- Future operations can include scaling the backend services, enhancing automated testing, and improving user support processes.

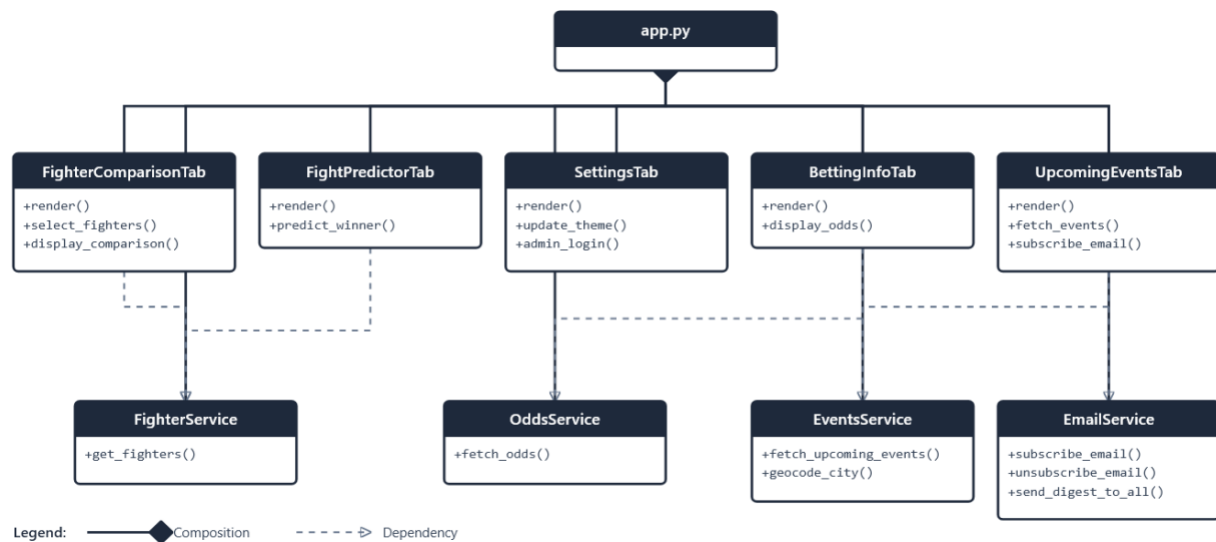
Limitations & Future Work

- Predictions depend on limited available data; they do not guarantee accuracy and may not capture underdog factors.
- The app currently lacks real-time user accounts or payment integration; user management features are minimal.
- Future work includes improving model accuracy with richer datasets, expanding to other combat sports, and enhancing the UI/UX with personalization and accessibility considerations.

Appendix

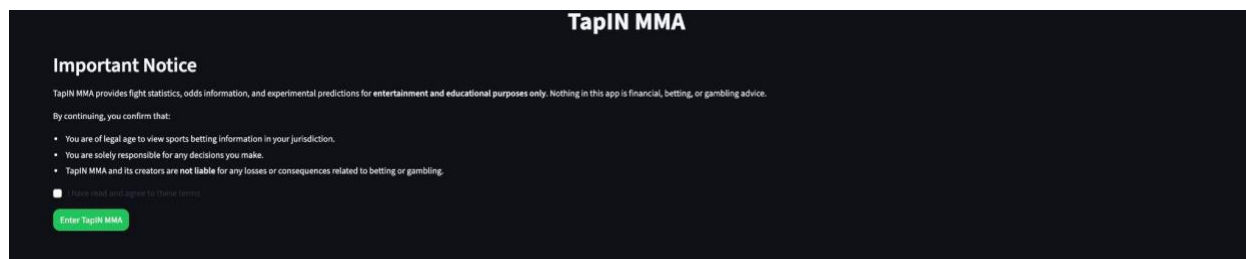
Appendix A – Diagrams

Object diagram:

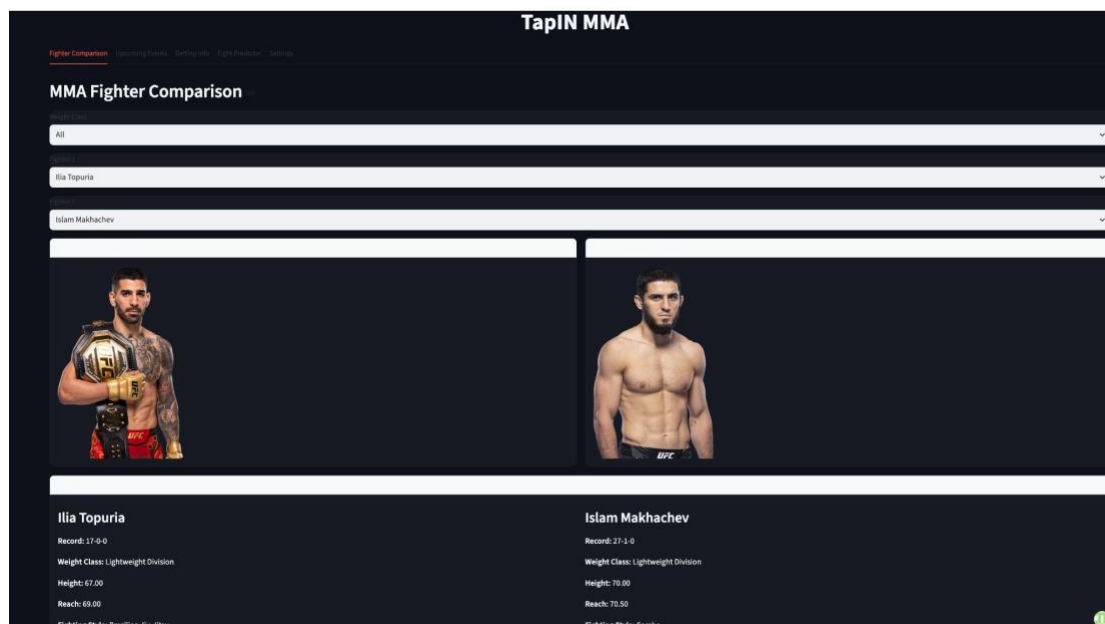


Appendix B – Screenshots

Landing Page



Fighter Comparison



Upcoming Events

TapIN MMA

Fighter ComparisonUpcoming EventsBetting InfoFight PredictionSettings

Upcoming MMA Events by Region

US - Ohio

> AFP 104 - Saturday, December 6, 6:00 PM ET

> Caged Thunder 36: Fight Night at the Civic - Saturday, December 13, 6:00 PM ET

> Live MMA at Penguin City 10 - Saturday, December 13, 7:00 PM ET

> Carnage Fight Series 5: Fights Before Chrl... - Saturday, December 20, 7:00 PM ET

> Ohio Combat League 42 - Saturday, December 20, 7:00 PM ET

> EGO Fight Series 6: Reloaded - Saturday, January 31, 6:00 PM ET

> Caged Thunder 37 - Saturday, January 31, 6:00 PM ET

Get Notified of Upcoming Fights

Notification System

Get Notified of Upcoming Fights

Get notified of upcoming fights

I'd like to receive emails about upcoming MMA events.

> Unsubscribe

Betting Info

TapIN MMA

Fighter ComparisonUpcoming EventsBetting InfoFight PredictionSettings

Betting Odds

Prices are American odds, implied probabilities, MMA odds pulled from The Odds API.

Last Fetched: 2025-12-03T18:40:20.531485+00:00

Filter > Matchup

All

Matchup	Book	Fighter	Odds	Implied Prob	Book Last Update
0 Malik Basahel vs Mahmoud Ataf	BetOnline.ag	Mahmoud Ataf		900 10.0%	2025-12-03T18:40:02Z
1 Malik Basahel vs Mahmoud Ataf	BetOnline.ag	Malik Basahel	-1500 93.8%		2025-12-03T18:40:02Z
2 Malik Basahel vs Mahmoud Ataf	Bovada	Mahmoud Ataf	900 10.0%		2025-12-03T18:39:37Z
3 Malik Basahel vs Mahmoud Ataf	Bovada	Malik Basahel	-1800 94.7%		2025-12-03T18:39:37Z
4 Malik Basahel vs Mahmoud Ataf	FanDuel	Mahmoud Ataf	790 11.2%		2025-12-03T18:40:20Z
5 Malik Basahel vs Mahmoud Ataf	FanDuel	Malik Basahel	-1500 93.8%		2025-12-03T18:40:20Z
6 Yanis Ghemmouri vs Islam Reda	BetOnline.ag	Islam Reda	-105 91.2%		2025-12-03T18:40:02Z
7 Yanis Ghemmouri vs Islam Reda	BetOnline.ag	Yanis Ghemmouri	-135 57.4%		2025-12-03T18:40:02Z
8 Yanis Ghemmouri vs Islam Reda	Bovada	Islam Reda	105 48.6%		2025-12-03T18:39:37Z
9 Yanis Ghemmouri vs Islam Reda	Bovada	Yanis Ghemmouri	-125 55.6%		2025-12-03T18:39:37Z

Fight Predictor

TapIN MMA

[Fighter Comparison](#) [Upcoming Events](#) [Betting Info](#) [Fight Predictor](#) [Settings](#)

Fight Predictor

Baseline rule-based model with a TapIN Pick. Toggle predictions below.

☒ Enable Predictions

Disclaimer: Experimental model -- not betting advice.

Select Weight Class

All

Fighter 1

Islam Makhachev

Fighter 2

Dricus Du Plessis

Predict the Winner

Islam Makhachev Confidence: 54.54%

Dricus Du Plessis Confidence: 45.46%

TapIN Pick: Islam Makhachev

Settings

TapIN MMA

[Fighter Comparison](#) [Upcoming Events](#) [Betting Info](#) [Fight Predictor](#) [Settings](#)

Settings

Choose App Background Color

Theme

Select Theme

Dark

Admin

Admin Username

Login as Admin

Logout Admin

Appendix C – Sprint Review Reports

[*Sprint 1*](#)

[*Sprint 2*](#)

[*Sprint 3*](#)

[*Sprint 4*](#)

[*Sprint 5*](#)

[*Sprint 6*](#)

References:

- [1] MongoDB, “*MongoDB Node.js Driver Documentation.*” Accessed: Dec. 3, 2025. [Online]. Available: <https://www.mongodb.com/docs/drivers/node/current/>
- [2] Octagon API, “*Octagon API Documentation.*” Accessed: Dec. 3, 2025. [Online]. Available: <https://api.octagon-api.com>
- [3] OpenAI, “*OpenAI API Documentation.*” Accessed: Dec. 3, 2025. [Online]. Available: <https://platform.openai.com/docs>
- [4] OpenStreetMap, “*Nominatim API Documentation.*” Accessed: Dec. 3, 2025. [Online]. Available: <https://nominatim.openstreetmap.org>
- [5] Plotly, “*Plotly Express Documentation.*” Accessed: Dec. 3, 2025. [Online]. Available: <https://plotly.com/python/plotly-express/>
- [6] RapidAPI, “*Topology API (Unofficial).*” Accessed: Dec. 3, 2025. [Online]. Available: <https://rapidapi.com>
- [7] Streamlit, “*Streamlit Documentation.*” Accessed: Dec. 3, 2025. [Online]. Available: <https://docs.streamlit.io>
- [8] The Odds API, “*Sports Betting Data API.*” Accessed: Dec. 3, 2025. [Online]. Available: <https://the-odds-api.com/>