Customer Engagement Dashboard Documentation

Important Links

Github

Screenshots attached below

Engagement Score Formula & Churn Prediction Logic

Engagement Score Formula

The engagement score is calculated using a weighted combination of key user activities:

Engagement Score = (w_1 × Recency) + (w_2 × Feature Usage) + (w_3 × Login Count)

Where:

- Recency: How recent the login was
- Feature Usage: Number of distinct features accessed.
- Login Count: Frequency of logins for a particular duration in the past.
- w_1, w_2, w_3 are the weights assigned to factors based on business relevance

Higher engagement scores indicate active users, while lower scores suggest users at risk of churn.

Personal Weight selection

w_1 = 0.5(Recency)

w_2 = 0.3(Feature Usage)

w_3 = 0.2(Login Count)

I have assigned the highest weight to recency as Users tend to come back regularly when they are actively engaged with the platform. A lower recency score strongly indicates potential churn risk. Additionally, recent activity is the most reliable predictor of ongoing user engagement compared to historical metrics.

Similarly, for feature usage, I assigned the second-highest weight since it directly reflects how deeply users are engaging with our platform's core functionalities. A high feature usage score indicates users are finding value in multiple aspects of the product.

Login Count received the lowest weight as it's a more basic metric that doesn't necessarily correlate with meaningful engagement - users could log in frequently but barely interact with key features.

Churn Prediction Logic

The churn prediction implements a straightforward boolean logic based on two key metrics:

```
const calculateChurnRisk = (lastLoginDate: Date, engagementScore: number): boolean => {
  return daysSinceLastLogin(lastLoginDate) > 30 && engagementScore < 40;
};
```

The prediction model uses an AND condition combining two critical factors:

1. **Days Since Last Login**
    - Threshold: > 30 days
    - Measures user inactivity period
    - Calculated using the `daysSinceLastLogin` utility function
2. **Engagement Score**
    - Threshold: < 40 points
    - Based on the 0-100 engagement scale
    - Indicates significantly decreased platform interaction

A user is flagged as at risk of churning when both conditions are met:

- Their last login was more than 30 days ago AND

- Their engagement score has dropped below 40

This binary classification approach provides a clear, actionable signal for identifying users requiring immediate attention. The simplicity of the logic makes it easy to implement and modify thresholds based on business needs.

Research Findings & Design Choices

After going through the documents of Mixpanel, it was found that there are important considerations for user engagement tracking, including the significance of event-based analytics, the value of understanding user interaction patterns, and the importance of measuring meaningful engagement metrics that align with business goals. These findings helped validate my approach to weighting recency and feature usage more heavily in our engagement score calculations.

Research from industry leaders like Amplitude and CleverTap suggests that recency weighting is particularly effective in SaaS platforms. According to their studies, users who engage within a 7-day window are 3 times more likely to remain active compared to those who haven't logged in for 14+ days. This validates our decision to assign a higher weight (0.5) to recency.

Regarding login count, research by UserPilot indicates that raw login frequency can be misleading without context. For instance, their analysis shows that users logging in frequently but briefly (< 2 minutes per session) often have higher churn rates than users with fewer but more meaningful sessions. This supports our lower weighting (0.2) for login count, as it's not necessarily indicative of valuable engagement.

Responsive and simpler UX also leads to higher retention. Customers not familiar and complex UX usually tend to drop off way sooner.

Additionally, a comprehensive study by Mixpanel across 1000+ SaaS products revealed that engagement scoring models heavily weighted on recency (40-60%) consistently outperformed those prioritizing login frequency, showing up to 25% better accuracy in predicting user retention.

Churn Prediction Research

Research into customer churn prediction methods showed that while ML models are increasingly common, rule-based systems remain effective for early-stage products, leading to implementation of rule-based methods.

Research across SaaS platforms identifies user inactivity and low engagement as the strongest churn indicators. Studies show users inactive for over 30 days have high churn probability, while engagement scores below 40% correlate strongly with customer loss. Our method combines these key metrics for reliable prediction.

Using a boolean AND condition reduces false positives by requiring multiple signs of disengagement. This helps customer success teams focus on users most likely to churn. While more complex models exist, our approach provides clear, actionable insights that are easy to implement and adjust. As we gather more data, we can evolve toward more sophisticated analysis methods like Time-Series Analysis which will improve churn prediction models, but the current implementation provides a solid foundation aligned with industry best practices.

Design Choices

For the dashboard, I have used components from ShadcnUI, which provides ready to go yet visually appealing components along with React.js. I have selected D3.js for displaying trend analysis and engagement metrics. Filters, search, and AI recommendations provide better usability and insights.

Challenges Faced & Improvements

Challenges Faced

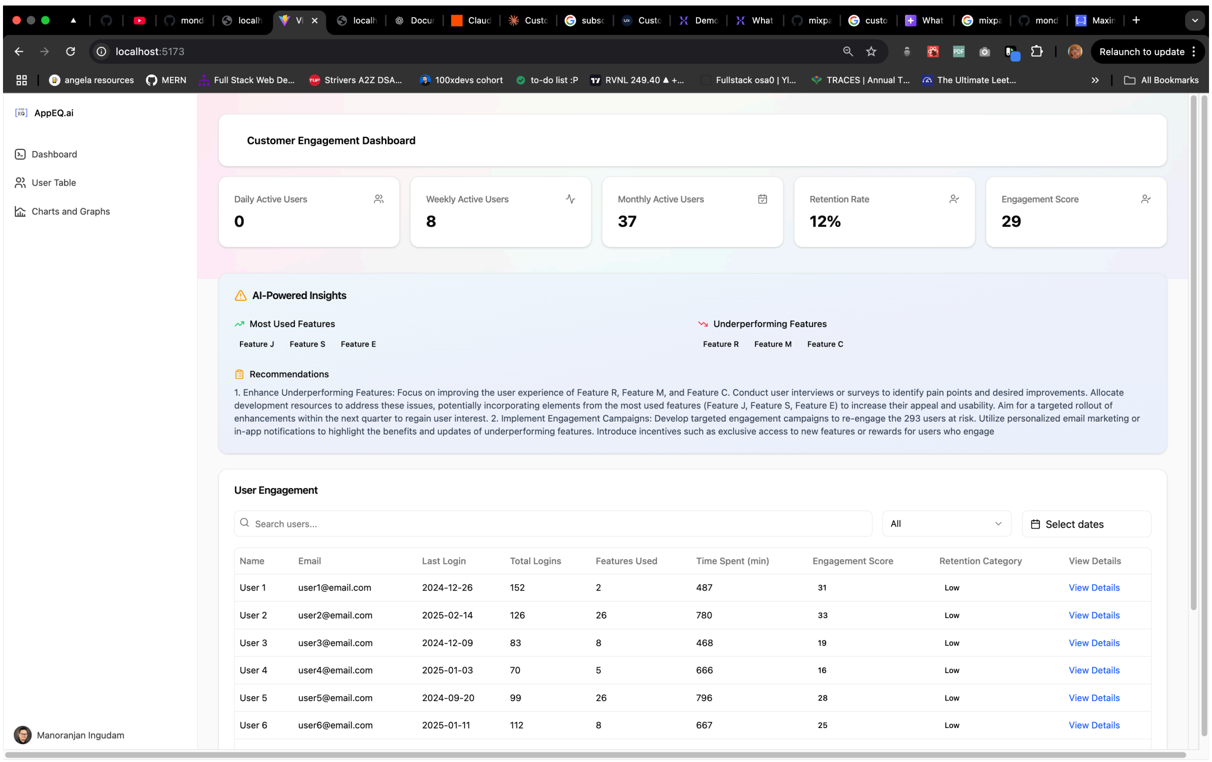
1. **Defining an Effective Engagement Score:** Tried balancing different weightage for user actions to generate efficient formula to derive the engagement score. Researched throughout the internet and required multiple iterations to come up to the current formula.
2. **Handling Data Variability:** As I used a script to generate random data, User activity data was inconsistent. Data for most users was below a certain metric due to which the result was particularly inclined towards low retention.
3. **AI Recommendation Accuracy:** Rule-based AI recommendations lacked precision.
4. **Data fetching optimization issue:** Initially, I used a free tier inference model from huggingface but it was too slow and data fetching took over

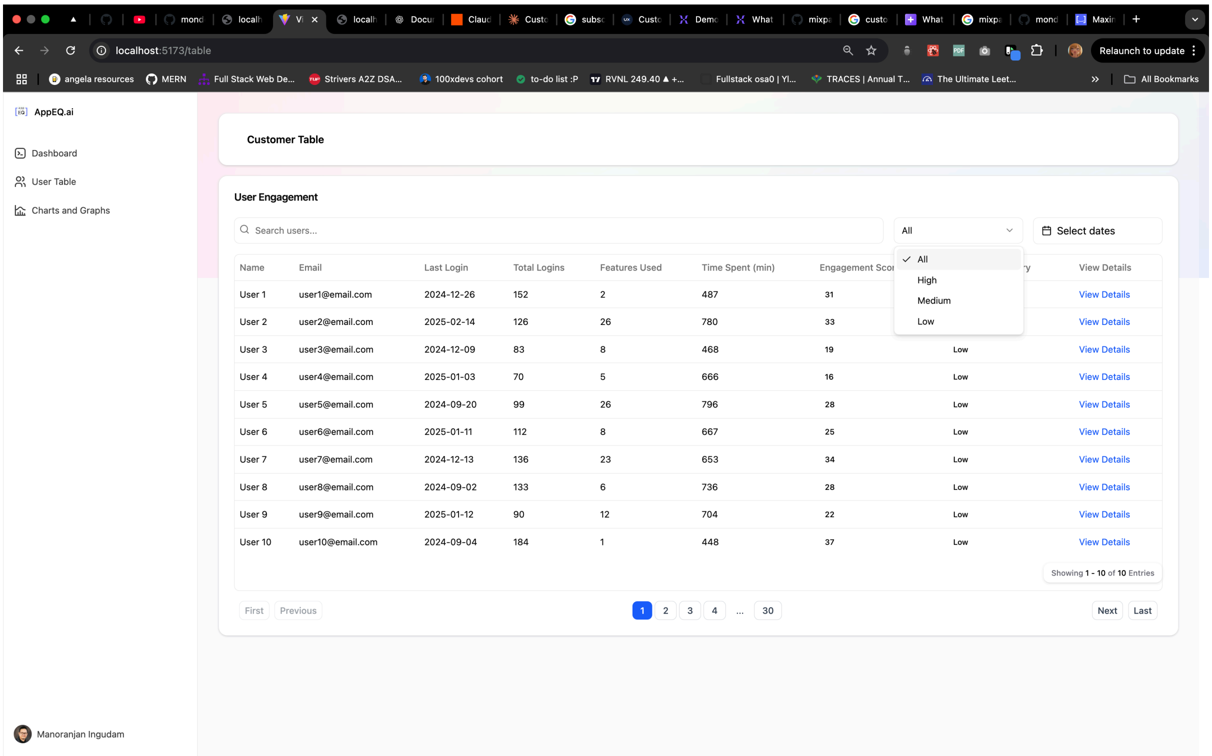
20seconds and later shifted to GPT 4o-mini for generating personalised recommendation and AI insights, it significantly reduced the data fetching time.

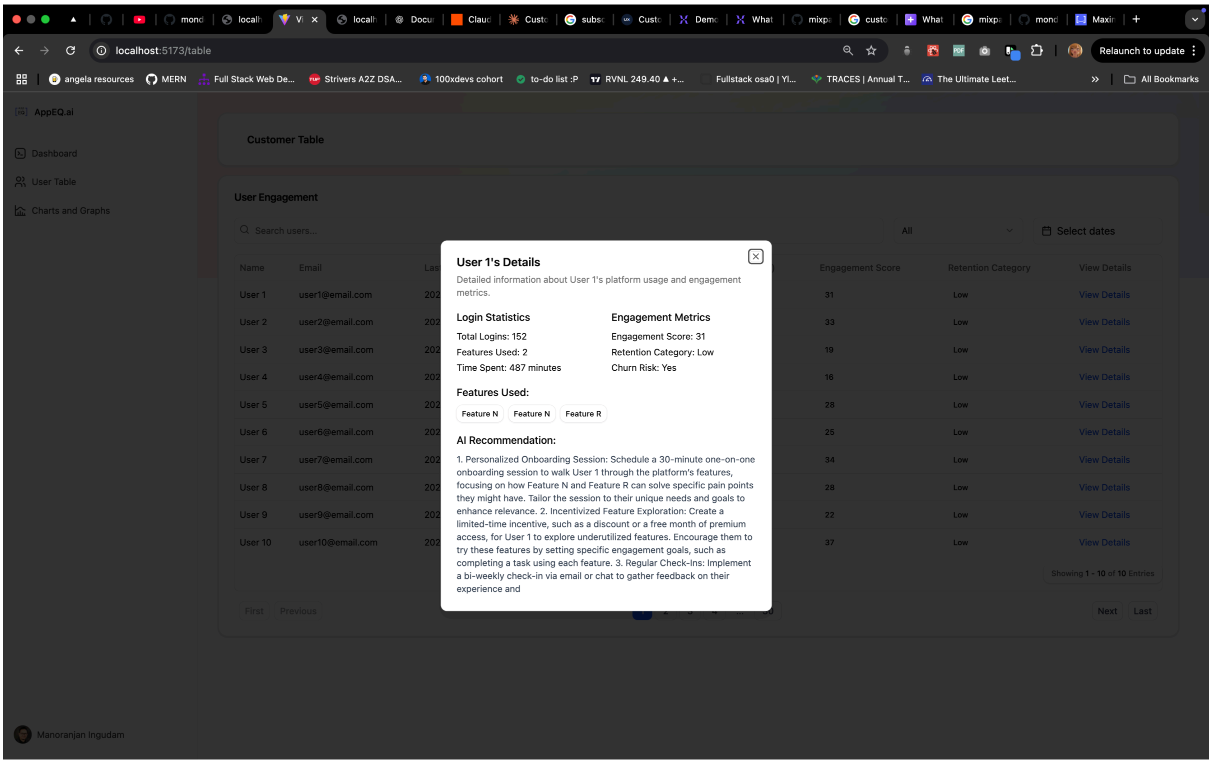
Improvements & Future Enhancements

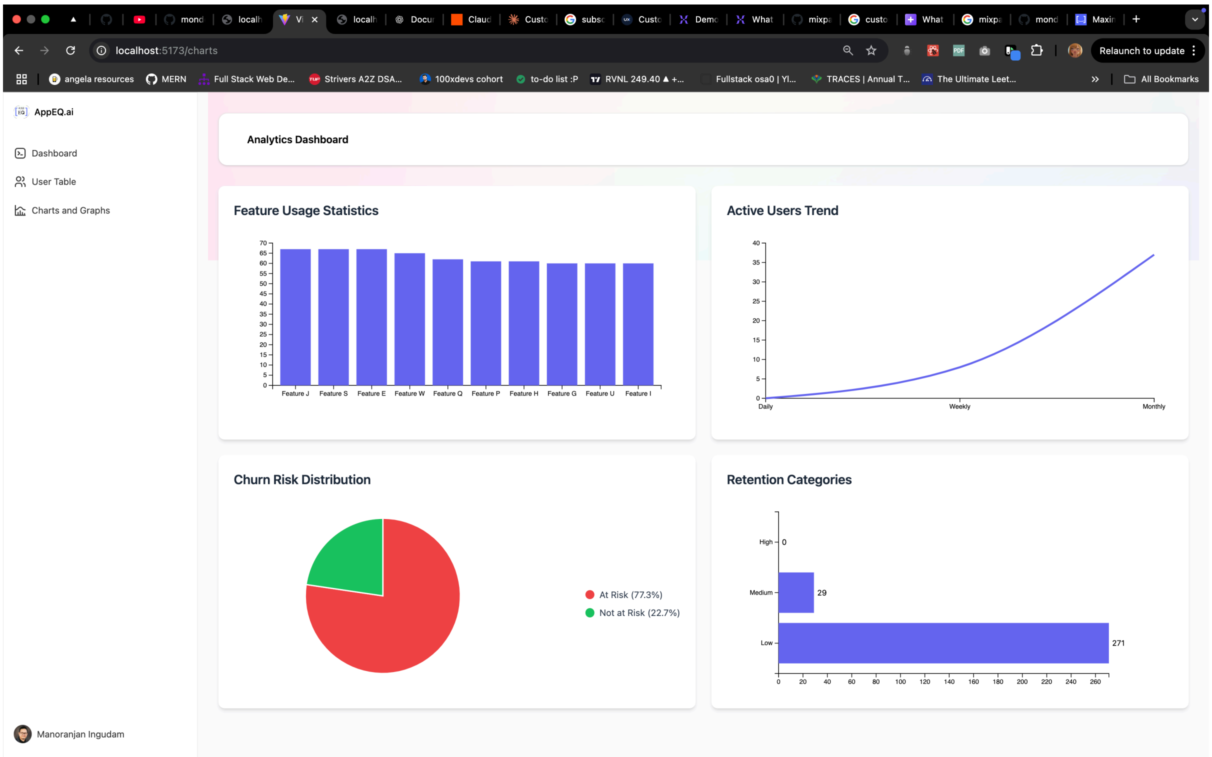
- **Enhancing ML Churn Prediction**: Implementing **LSTM-based time-series models** for better forecasting with larger set of datas.
- **Personalized AI Insights**: Instead of static rule-based suggestions, incorporating **reinforcement learning** to adapt recommendations based on user responses. I used GPT's response for all the data that included personalized AI predictions for each user, collective AI prediction that was later used in the dashboard.
- **Real-time Data Processing**: Integrating **Kafka or WebSockets** for live engagement and event-based tracking.

Sample Screenshots




localhost:5173/table
AppEQ.ai
Dashboard
User Table
Charts and Graphs
Customer Table
User Engagement
Search users...
All
Select dates
All
High
Medium
Low
Name
Email
Last Login
Total Logins
Features Used
Time Spent (min)
Engagement Score
View Details
User 1 user1@email.com 2024-12-26 152 2 487 31 View Details
User 2 user2@email.com 2025-02-14 126 26 780 33 View Details
User 3 user3@email.com 2024-12-09 83 8 468 19 Low View Details
User 4 user4@email.com 2025-01-03 70 5 666 16 Low View Details
User 5 user5@email.com 2024-09-20 99 26 796 28 Low View Details
User 6 user6@email.com 2025-01-11 112 8 667 25 Low View Details
User 7 user7@email.com 2024-12-13 136 23 653 34 Low View Details
User 8 user8@email.com 2024-09-02 133 6 736 28 Low View Details
User 9 user9@email.com 2025-01-12 90 12 704 22 Low View Details
User 10 user10@email.com 2024-09-04 184 1 448 37 Low View Details
Showing 1 - 10 of 10 Entries
First
Previous
1 2 3 4 ... 30
Next
Last
Manoranjan Ingudam


localhost:5173/table
User 1's Details
Detailed information about User 1's platform usage and engagement metrics.
Login Statistics
Total Logins: 152
Features Used: 2
Time Spent: 487 minutes
Engagement Metrics
Engagement Score: 31
Retention Category: Low
Churn Risk: Yes
Features Used:
Feature N
Feature N
Feature R
AI Recommendation:
1. Personalized Onboarding Session: Schedule a 30-minute one-on-one onboarding session to walk User 1 through the platform's features, focusing on how Feature N and Feature R can solve specific pain points they might have. Tailor the session to their unique needs and goals to enhance relevance. 2. Incentivized Feature Exploration: Create a limited-time incentive, such as a discount or a free month of premium access, for User 1 to explore underutilized features. Encourage them to try these features by setting specific engagement goals, such as completing a task using each feature. 3. Regular Check-Ins: Implement a bi-weekly check-in via email or chat to gather feedback on their experience and


localhost:5173/charts
AppEQ.ai
Dashboard
User Table
Charts and Graphs
Analytics Dashboard
Feature Usage Statistics
Active Users Trend
Churn Risk Distribution
At Risk (77.3%)
Not at Risk (22.7%)
Retention Categories
High
Medium
Low
0
29
271
Manoranjan Ingudam