CS4530 Final Project: HuskyFlow Community Engagement

Group 907: Giles, Jade, and Maxwell

Our Feature:

In the original stack overflow, users can ask and answer questions, but there was no motivation for them to do so. We identified three key problems:

- **No recognition:** users were not able to see their own standing or be recognized for their contributions
- **No assessment:** there was no structured way to test a users knowledge and verify their credibility
- **No achievements:** there was no system to reward a users' milestones or accomplishments

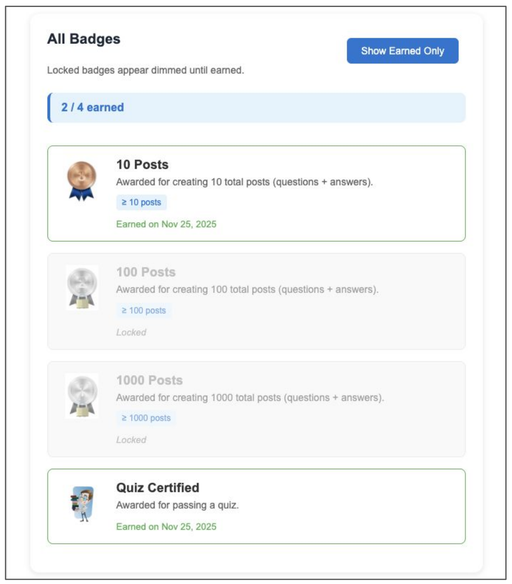
This lead our team to develop three interconnected features:

- **Leaderboards:** Ranked lists that show top contributors based on answers and upvotes. Users can filter by time period and/or by tag. Users are able to see where they stand globally, or within a specific topic.
- **Quizzes:** Assessments with various question types, implementing auto-grading, time limits, and attempt limits. Admins are able to create quizzes and users are able to attempt them and review their answers.
- **Badges:** An achievement system for rewarding user milestones (e.g. 10, 100, 1,000 posts) and quiz completions. Displays earned badges on the user's profile with badge description, criteria, and date earned.

Design Decisions:

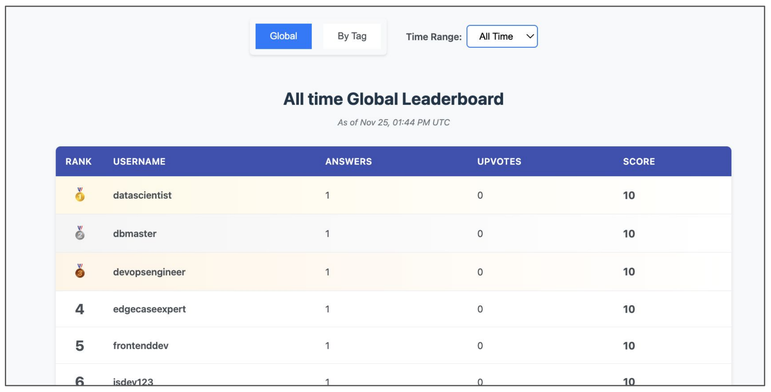
- **Compute on read:** Leaderboards are MongoDB aggregations over existing Questions/Answers/Tags; time filters use **`askDateTime/ansDateTime`,** tag filtering is an extra **`$match`** (no new tables).
- **Stable, UI-ready API:** Leaderboards return **rank, username, answers, upvotes, score** plus **timeRange, tagId, computedAt** so the client can label views without storing snapshots.
- **Generic badges:** Split into **BadgeType** (definition: name/icon/description/kind/criteria) and **BadgeAward** (userId, badgeTypeId, achievedAt, meta); unique (userId, badgeTypeId) makes awards **idempotent**.
- **Display-ready endpoints:** **`GET /api/badges/me`** returns key/name/icon/description/criteria/earnedAt; **`GET /api/badges/types`** powers an "All badges" legend.
- **Flexible architecture: quizquestions** collection uses **questionType** field, enabling versatile quiz structure. **Many-to-one** relationship with questions allows for question re-use and dynamic quiz construction.
- **API first design:** quizzes have an **API driven** management design, allowing administrators to access an array of **CRUD** endpoints for managing **quizzes** and **quizquestions** collections.

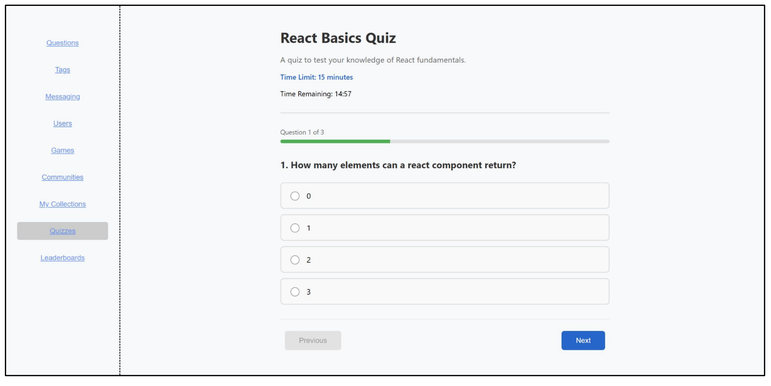
Demo and Source:



[Link to Demo Site]

[Link to Github]





Leaderboard, Quiz, and Badges features are shown above

Future Work:

- **Create new badge types**
- **Badge progress is tracked and displayed**
- **Admins can revoke badges if earned through rule violations or cheating**
- **Leaderboards can reset for Seasonal/Temporary Events**
- **Leaderboards can be filtered for specific groups, like students or new users**
- **A leaderboard scoring formula is available for the public to view/download**
- **Quizzes can detect basic cheating behaviors**
- **quizAttempt collection for scalable data handling**
- **Interface for public user to propose quizzes and/or quiz questions.**