

CS4530 Final Project:

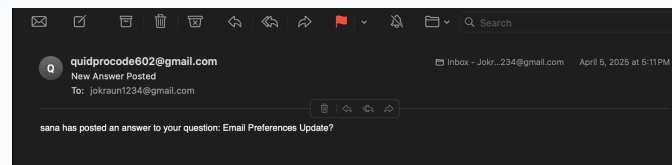
{¢} QuidProCode

Answers worth rewarding.

Our Features:

While demoing the class release FakeStackOverflow, we observed that users had little motivation to answer questions. We used this fact to explore the ways we could increase user incentive. In our release of QuidProCode, an improvement of FakeStackOverflow, we introduced three core features: CodeCoins, badges and notifications, and a new game, ConnectFour.

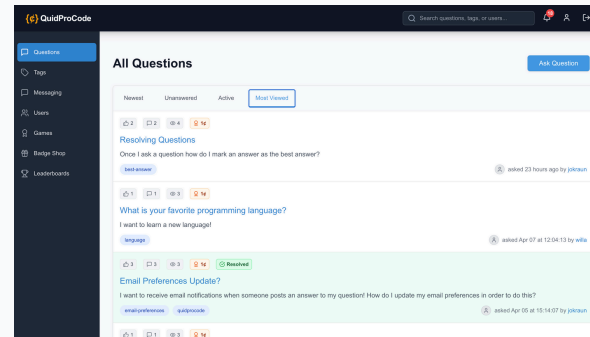
When users interact with application features, they're awarded CodeCoins which can be spent to post bountied questions, purchase badges, etc. Badges continue this reward system, with a verified badge that can be used to obtain a premium on answered questions. ConnectFour and the staked games option develop more interesting game dynamics. Finally, to keep our users updated and engaged, a real-time notification system was implemented, as well as a user option to enter their email address to receive email notifications pertaining to their questions while offline.



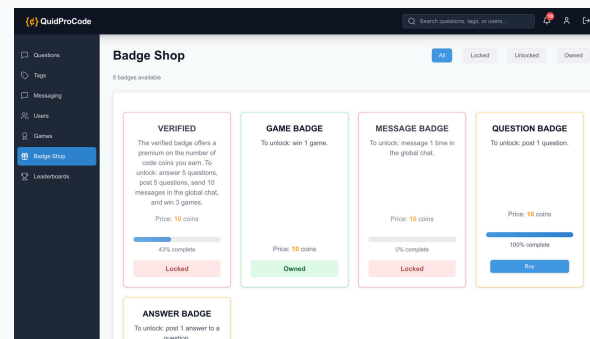
Sample email received when an answer is posted to your question.

Future Work:

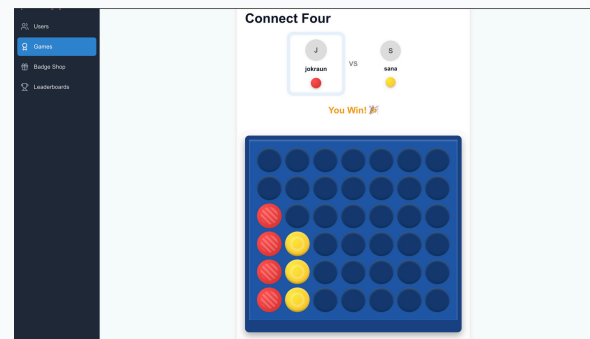
We have a lot of room for improvement in our application. Firstly, we would love to Implement more benefits for verified users once they unlock and purchase verified badges. For example, we'd like to Implement specialized online events for verified users to join that is unique to them, exchanging knowledge and sharing Improvement Ideas for QuidProCode. Further, we would like to Incorporate user cookies to display personalized adds on the website, with the option for users to pay a subscription to access a premium, no-ad QuidProCode. By boosting our application's revenue, we will be emboldened to create new and Improved features to build a better user experience.



Questions can now be "Resolved" when a user selects the "Best Answer."



Badge Shop where users can purchase unlocked badges.



New Connect Four Game w/ staked game optionality.

Tech Stack & Design:

Backend:

Firstly, we began by defining the required changes to our database schema. This involved the implementation of our CodeCoins currency for user wallets, bounties and bestAnswer fields for questions, and a new Notification type so that users were able to see notifications even after being offline for them. We defined new routes such as:

```
POST: '/resolveQuestion'
PATCH: '/addBountyToQuestion'
GET: '/getNotifications/:username'
DELETE: '/deleteNotification'
PATCH: '/purchaseBadge'
GET: '/getUserBadges/:username'
```

After implementing API endpoints, we got to work on creating WebSocket events for all of our new features. This ensures Socket events are emitted to the correct users when questions are answered, notifications are sent, and even when game rooms are created or game moves are made.

Frontend:

On the frontend, we created a massive overhaul of the UI design, ensuring a clean and sleek look throughout the application, alongside colorful logos and images that enhance the visuals of QuidProCode. We correctly implemented React hooks alongside components so that the user interface is completely connected, allowing for seamless interaction between UI components for all of our new features.

Source and Demo:

Deployed Website Link:

<https://cs4530-s25-602-client.onrender.com>

GitHub Repository:

<https://github.com/neu-cs4530/spring25-team-project-spring25-project-group-602>