**MAINTENANCE**

**Bug Tracking and Resolution**

After deployment, the project was far from over. Maintaining the quiz website required regular monitoring, bug tracking, and improvements based on user feedback. To manage this, we used GitHub Issues to log and track bugs, feature requests, and general improvements.

Whenever we encountered bugs or users reported issues, we would create an issue in the repository. Each issue was categorized (bug, feature, enhancement) and given a priority level. Developers would assign themselves to resolve specific issues, and once a fix was implemented, it would be tested, reviewed, and merged.

Using GitHub Issues helped keep everything organized and allowed us to manage tasks more effectively. We could also track the progress of each bug or feature request, ensuring that everything was addressed in a timely manner. This transparent process made it easy for the entire team to stay on the same page.

**Feedback Loop**

User feedback was another important aspect of maintaining the website. After deployment, we encouraged users to share their experiences and provide suggestions for improvement. We set up a Google Form for users to submit their feedback, which we reviewed periodically.

In addition to this, we monitored the GitHub Discussions page, where users and contributors could post ideas, feature requests, or report bugs. The feedback loop helped us understand how users interacted with the website and what improvements could be made.

We also used analytic tools like Google Analytic to gather data on user behavior. This data allowed us to identify areas where users were struggling or where engagement was low, which informed decisions about further updates and changes.

**Regular Updates and Commits**

Finally, regular updates and commits were essential for maintaining the project’s health. We established a schedule for committing new features, bug fixes, and improvements. New commits were pushed to the repository, and changes were thoroughly tested to ensure the website remained bug-free and high-quality.

With the collaborative nature of the project, every team member was responsible for ensuring that the code-base remained clean and well-documented. Pull requests were reviewed and discussed, and code standards were enforced to maintain a consistent quality of code across the project.

In addition to bug fixes, we regularly updated the website’s content, quiz questions, and features. For example, we added new question categories, implemented a dark mode option for better accessibility, and included social media sharing features. Each update was carefully planned, tested, and deployed to ensure a smooth user experience.

Maintaining Quiz Hub after launch was a continuous and collaborative effort. Our goal wasn’t just to keep the site running—it was to **improve the user experience**, **fix bugs quickly**, and **respond to feedback effectively**. GitHub played a central role in helping us organize, prioritize, and track these efforts.

### **1. Using GitHub Issues for Bug Tracking and Feature Requests**

The GitHub **Issues** tab became our primary maintenance dashboard. Whenever someone encountered a bug or suggested a new feature, we’d log it as an issue.

We used **labels** like:

* bug: for tracking functional or visual glitches
* enhancement: for UI/UX improvements or new quiz categories
* help wanted: when we needed input or code review from teammates
* good first issue: for on-boarding new contributors or junior teammates

#### Real Example:

A tester reported that the **score wasn't updating correctly** when a user skipped a question. We reproduced the bug, logged it under bug, and referenced it in a pull request:

This automatically closed the issue once the PR was merged.

### 2. **Project Boards for Organizing Tasks**

We used **GitHub Projects** to create a Kan-ban board that helped us visualize what stage each task was in:

* 📥 To Do
* 🛠️ In Progress
* ✅ Done

This allowed the team to stay in sync and avoid duplicate work. For example, if someone picked up a styling fix, they’d move the card to "In Progress," preventing others from working on the same task.

The visual nature of project boards made it easier to manage maintenance over time.

SCREENSHOTS:



