

4.2 Non-Functional Requirements

User	Requirement	Definition
Usability	User-Friendly Interface	Information such as the following made easily accessible: <ul style="list-style-type: none"> • Rating submission • Search • User account management
	Intuitive Navigation	Implement a logical flow in the application architecture, ensuring that users can navigate from one function to another seamlessly.
Performance	Fast Loading Times	Optimize both front-end and back-end components to ensure quick loading of the web pages. Use efficient coding practices, minimize HTTP requests, and leverage browser caching.
	Optimized for High Traffic	Load balancing, using efficient databases with fast read and write capabilities, and potentially incorporating microservices to handle different aspects of the platform.
Reliability	High Availability	Aim for 99.9% uptime by using reliable hosting services with excellent track records.
	Regular Updates and Maintenance	Schedule regular maintenance and updates outside of peak hours. Provide users with advance notice of downtime via the platform and communication channels like email or social media.
Security	Data Encryption	Use strong encryption protocols such as TLS for data in transit between users' devices and your servers. Ensure that data at rest is also encrypted using robust algorithms to protect sensitive user data and feedback.
	Secure Handling of User Data	Follow best practices for security, such as implementing OWASP security principles, to prevent common vulnerabilities such as SQL injection, XSS, and CSRF attacks.
	Authentication and Authorization	Implement strong authentication mechanisms, including the option for two-factor authentication (2FA) for added security.