MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

MANIPAL HACKATHON 2024

</ PIONEERING PARADIGMS />

ROUND 2
Technical Rule Book

Overview

1. Development Guidelines
2. GitHub Workflow
3. Dataset Usage
4. Bug Bounty Round
5. Final Presentation Round
6. Contact Us

1. Development Guidelines

1. **Original Work**: All code must be developed during the hackathon. Pre-written code is not permitted.
2. **Referencing External Code:** Proper citations must be provided for any referenced code. Code similarity checks will be carried out to ensure compliance.(Refer to page x)
3. Teams are advised to set up emulators on their systems before the commencement of the hackathon.

2. GitHub Workflow

1. **Initial Setup:**
   - A common repository containing a 'README' file template and rulebook will be provided at the beginning of the hackathon. Teams must clone this repository.
   - Teams must follow the README file template.

2. **Branching Strategy:**
   - Teams should create a branch named after their team for development.
   - Sub-branches can be created within the team branch to manage independent workflows.

3. **Commit and Push Protocol:**
   - Regular commits are required to show continuous progress.
   - Teams must push updates to the remote repository at least once every hour.

4. The Repository will be accessible **only** to the required team.

5. Competitors will get **read access** during the Bug-Bounty Round.

3. Dataset Usage

1. Teams are encouraged to use publicly available datasets from platforms like data.gov.in or Kaggle.
2. Generating your own dataset in sensitive areas such as healthcare (e.g. EMG, EEG), is strictly prohibited and will result in appropriate actions being taken.

4. Bug Bounty Round

The Bug Bounty Round will take place after **33** hours of coding, before which teams are expected to have a **complete project**. In this round, teams will test each other's projects for bugs, with the outcome significantly impacting the **overall score**, adding a competitive element focused on code quality and stability. Team that find bugs is here by referred to as *evaluators* and the team whose code is being tested will be referred as *evaluatees*.

1. **Prerequisites:**

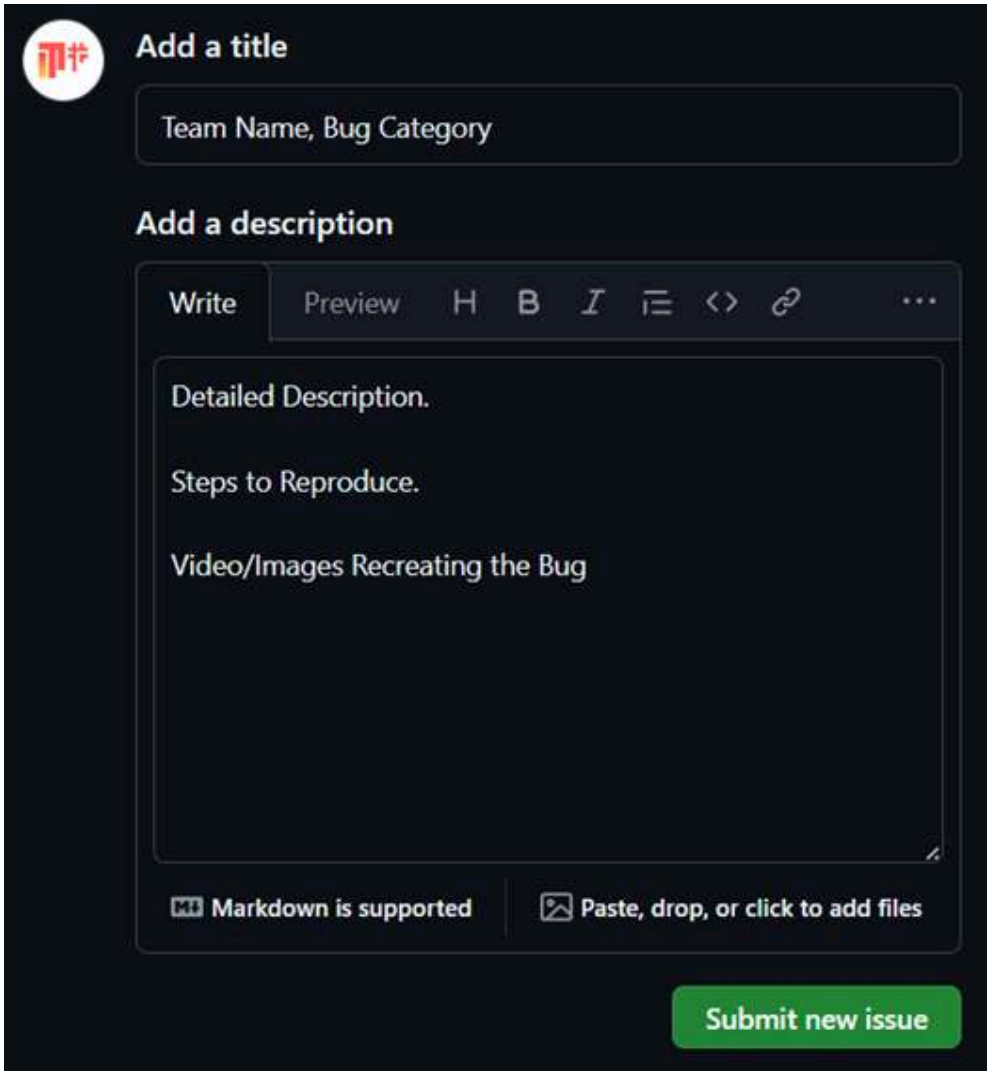
- Websites **must be** deployed online or locally and fully operational
    a. Teams will be on the **same network** to allow local hosting.
    b. 10 points will be awarded to the teams overall score for this.
    c. Teams are **not expected** to give admin access to the competing teams.
- Applications **should be** downloadable, with all APIs **functioning** as intended.
    a. 10 points will be awarded to the teams overall score for this.

4. Bug Bounty Round

- If a team is unable to deploy their website, it is recommended they provide a Docker image along with clear setup instructions in their documentation.
- The Team must **push** their final presentation to their GitHub repository before this round begins.
- Teams are **expected** to have a **buildable** project with clear setup instructions in their GitHub repositories that the opponent team can download and build on their system.
- Teams must inform the organization committee if their project **doesn't meet** these requirements before the commencement of this round, failure to so will result in ***disqualification* from the hackathon.**
- Failure to meet these requirements (with proper notice) will result in a **30% deduction** from the **overall score** and ***disqualification* from *this* round.**

4. Bug Bounty Round

2. Teams will be grouped **randomly** where they will test each other's projects for bugs within a **3-hour** window.
3. During this round, push access will be **disabled** for all teams, preventing any direct changes to their repository.
4. All bugs should be reported as GitHub Issues of the Evaluatees repository in the following format:



4. Bug Bounty Round

5. **Scoring System:**

- This round will have its ***own point system*** where a *maximum of 100 points* can be earned or lost. This will influence the team's cumulative score *(until this round)* by **±30%.**
- Evaluator will **gain** points for **valid** bug reports and Evaluatee will **lose** points for having bugs in their own projects.
- Scores outside this range will be normalized.
- Bonus points will be awarded to the top 3 teams who cross 100 points. (*Details will be shared at the venue*)

4. Bug Bounty Round

6. **Example:**
    - A team earns 80 points overall and scores 50 points in the bug bounty round. After this round, their final score will be 92. Here's how: 50% of 30% equals 15%, and 15% of 80 is 12. Thus, the final score is **80 + 12 = 92**.
    - Similarly, if a team earns 80 points overall and scores -50 points in the bug bounty round. After this round, their final score will be 68. Here's how: 50% of 30% equals 15%, and 15% of 80 is 12. Thus, the final score is **80 - 12 = 68**.

4. Bug Bounty Round

7. **Points System:**

Grading						
Bug Category	Invalid bug	Valid Bug			Total Gain	Total Loss to Opponent Team
		GitHub Issue Format Followed	Proper Bug Description	Valid Reproduction Instructions		
Functional	-5	1	9	10	20	-16
Security	-5	1	9	10	20	-16
UI/UX Minor	-5	1	2	4	7	-5
UI/UX Moderate	-5	1	5	6	12	-9
Data Handling	-5	1	9	10	20	-16
Logical	-5	1	9	10	20	-16
Internationalization and Localization Minor	-5	1	2	4	7	-5
Internationalization and Localization Major	-5	1	2	4	7	-5
Crash	-5	1	9	10	20	-16
Installation and Deployment	-5	1	5	6	12	-9
Error Handling Minor	-5	1	2	4	7	-5
Error Handling Moderate	-5	1	5	6	12	-9
Documentation	-5	1	4	-	5	-5
False Claim	-5	1	1	-	2	-20

4. Bug Bounty Round

- If a bug is reported **more than once** by different evaluators within a **2-minute** window, the latter reports will be **ignored**. However, if it is reported after this window, the evaluators other than the first evaluator will incur a **deduction of 5 points**.
- Any evaluator that wrongly classifies their bug will incur a **deduction of 5 points** from their score **for the bug**. Judges will determine if a bug has been classified correctly or incorrectly
- If an evaluator believes that a bug does not fit into any of the listed categories, they must label it as "**Unknown**".
    - If the judges agree, the evaluator will receive points based on the closest appropriate category that the judges determine fits the bug.
    - If the judges disagree, the evaluator will incur a **deduction of 5 points**. from their score for the bug, and the judges will rectify the bug classification.
- Closing a bug report will incur a **deduction of 5 points.**

4. Bug Bounty Round

Classification of Bugs with Examples:

1. **Functional**
   - Issues affecting the core functionality of the application, causing features to not work as intended.
   - *Example:* Missing or incorrect button actions, non-responsive features, incorrect business logic execution.
2. **Security**
   - Vulnerabilities that expose the system to threats such as data breaches or unauthorized access.
   - *Example:* Unencrypted data transmission, improper user authentication vulnerability to Cross-Site Scripting (XSS), SQL injection risks.
3. **UI/UX Minor**
   - Small visual or usability issues that don't majorly impact the user experience but are noticeable.
   - *Example:* Misaligned icons or text, small touch targets

4. Bug Bounty Round

4. **UI/UX Moderate**
   - More significant user interface problems that can confuse or frustrate users, affecting navigation or performance.
   - *Example:* Overlapping UI elements, confusing navigation, poor scroll performance.
5. **Data Handling**
   - Errors related to data storage, processing, or transfer, resulting in inconsistent or lost data.
   - *Example*: Data loss during form submission, malformed JSON responses, inconsistent data synchronization across devices.
6. **Logical**
   - Bugs related to incorrect decision-making in the code, often due to flawed algorithms.
   - *Example:* Incorrect sorting algorithms, failed conditional logic paths, improper state management

4. Bug Bounty Round

7. **Internationalization and Localization Minor**
   - Small issues related to adapting the app for different languages or regions, typically not affecting functionality.
   - *Example:* Mistranslated labels, non-localized date/time formats.
8. **Internationalization and Localization Major**
   - Critical issues in localization that may break the app or cause significant functionality loss in different regions.
   - *Example:* Failure to load data correctly when set to a certain region due to incorrect date format parsing, preventing key features from functioning, users in certain countries are unable to sign up because the app rejects valid local phone numbers or postal codes specific to their region.
9. **Crash**
   - Bugs that cause the app to crash.
   - *Example:* Opening the app and attempting to upload a large image file causes the app to crash, Crashes During Network Failure.

4. Bug Bounty Round

10. **Installation**
    - Problems related to the setup or installation.
    - *Example:* The installer fails to complete the setup process, showing an "installation failed" error, The app is not launching after installation due to missing system dependencies.
11. **Error Handling Minor**
    - Minor issues where the app doesn't properly notify users about small input errors or vague messages.
    - *Example:* Vague error messages, lack of validation warnings.
12. **Error Handling Moderate**
    - More significant errors where the app fails to handle issues such as network failures or unhandled exceptions.
    - *Example:* No fallback for network failure, failure to handle exceptions.

4. Bug Bounty Round

13. **Documentation**
    - Incomplete, outdated, or inaccurate documentation.
    - *Example:* Missing environment setup instructions, outdated API references.
14. **False Claim**
    - Features or compatibility falsely advertised but not actually available or supported.
    - Example: Features advertised in presentation but not present in project.

5. Final Presentation Round

1. In the Presentation round, the top **10** teams will showcase their projects to all judges and participants.
2. This round will be judged by a panel of judges.
3. **Time Limit:** Each team will have a maximum of **8 minutes** to present their project, followed by a **2 minutes** Q&A session.
4. This will be followed by the Final Evaluation Round.

6. Contact Us

Team Manipal Hackathon 2024

Dillon Almeida: **+91 86686 33682**

Mustafa Haji: **+91 97738 88772**

Rishabh Jain: **+91 87997 71287**