Project Overview

• Project Name: HackAZ
• Team Members:
– Kanit Mann
– Tanishk Singh

• Description:
TBD

• Objectives / Goals:
– State the key objectives for the solution.
– Define what success looks like for this hackathon.

2. Timeline & Milestones

• Hackathon Schedule:
– Day 1: Key tasks or goals
– Day 2: Key tasks or goals
– Day 3: Key tasks or goals (if applicable)

• Milestones:
– Milestone 1 (e.g., project idea finalized)
– Milestone 2 (e.g., prototype completed)
– Milestone 3 (e.g., testing and polishing)

• Deadlines:
– 23rd March 2025

3. Role Breakdown & Responsibilities

Since you're a team of two, clearly define who does what.

• Kanit Mann:
– Main responsibilities (frontend, design, research, etc.)
– Secondary/support responsibilities

• Tanishk Singh:
– Main responsibilities (backend, API integration, etc.)
– Secondary/support responsibilities

- Communication Methods:

– Preferred communication tools (Slack, WhatsApp, etc.)
– Frequency of check-ins

---

4. Brainstorm & Ideation

- Initial Ideas:

– List all initial concepts or problem areas identified.
– Document discussions around feasibility and impact.

- Idea Validation:

– Research or quick user interviews (if applicable).
– Links to relevant articles, documents, or data used for validation.

- Final Project Direction:

– What idea did you choose and why?
– Criteria used to make the final decision (e.g., impact, complexity, time constraints).

---

5. Technical Stack & Tools

- Programming Languages & Frameworks:

– Language (e.g., Python, JavaScript, etc.)
– Frameworks (e.g., React, Flask, Node.js, etc.)

- Libraries & Dependencies:

– List of libraries/packages and their purposes.

- Collaboration Tools:

– Version control (Git, GitHub/Bitbucket, etc.)
– Project management (Trello, Jira, Notion)

- Any other required services (cloud hosting, APIs, databases, etc.):

– Service name, credentials (if needed), usage instructions.

---

6. Design & Architecture

- System Architecture Diagram:

– Simple sketch or formal diagram of the system flow (frontend, backend, database, external APIs).

- UI/UX Overview (if applicable):

– Wireframes, mockups, or quick sketches.
– Color schemes, layouts, user flow.

- Key Components:

– Highlight crucial parts (e.g., authentication module, data processing, etc.).

7. Development Progress & Daily Logs

Record progress at regular intervals. For a multi-day hackathon, you might have daily logs:

- Day 1:

– Goals set.
– Accomplishments.
– Roadblocks/issues and how they were addressed.
– Personal contributions (Name 1 & Name 2).

- Day 2:

– Goals set.
– Accomplishments.
– Roadblocks/issues and how they were resolved.
– Personal contributions (Name 1 & Name 2).

- Day 3 (if applicable):

– Goals set.
– Accomplishments.
– Roadblocks/issues and how they were resolved.
– Personal contributions (Name 1 & Name 2).

Add more days or time slots as needed. Be sure to differentiate individual tasks between the two members.

8. Testing & Quality Assurance

- Testing Strategy:

– Unit tests, integration tests, manual testing approach, etc.

• Test Cases & Results:
– Outline scenarios tested (e.g., typical usage, edge cases, error conditions).
– Document pass/fail status.

• Bugs & Fixes:
– Any critical issues discovered during testing.
– Steps taken to fix them.

---

9. Final Deliverables

• Demo / Presentation Material:
– Link to demo video or presentation slides if applicable.
– Keep a brief summary of the key points you plan to present.

• Code Repository Link:

- https://github.com/kanitmann01/hackaz_team_wildhackers
- https://github.com/users/kanitmann01/projects/1

• Deployed App (if applicable):
– URL for live demo.
– Instructions or credentials if needed to log in/test.

---

10. Personal Reflections & Contributions

Use this section for each team member to reflect on what they contributed and learned:

• Kanit Mann
– Contributions (summarize tasks owned, code written, features developed).
– Key learnings (technical skills, teamwork, time management).
– Challenges faced and how they were overcome.

• Tanishk Singh
– Contributions (summarize tasks owned, code written, features developed).
– Key learnings (technical skills, teamwork, time management).
– Challenges faced and how they were overcome.

---

11. Lessons Learned & Future Improvements

• Overall Lessons:
– What went well?
– What could be improved for the next hackathon or future projects?

• Next Steps (if continuing project post-hackathon):
– Features to add or refine.
– Potential collaborations or expansions of the idea.

• Acknowledgments:
– Mentors, sponsors, or other teams who assisted.