General Instruction

1. Student Instruction

Basic Rules

1. Each team must consist of 1–4 students.
2. Every team must submit via GitHub Repository Link
    - Working Project (Frontend + Backend)
    - Final Presentation (PPT and Demo)
3. Project must be:
    - Original
    - Developed during the hackathon timeline
4. AI tools (ChatGPT, Copilot, etc.) are allowed for guidance, but:
    - You must understand your code.
    - You must explain your logic during judging.

Timeline Discipline

- No late submissions.
- Final Git commit time will be verified.
- Last-minute ZIP submissions without proper repository history will be rejected.

Project Expectations

Your project must include:

- Working UI
- Proper Backend/API
- Database integration
- Deployment (recommended and Preferable)

2. Git Repository & Team Development Rules

Mandatory Requirements

1. Repository must be created by the team leader within 24 hours of releasing the problem statement.
2. Repository name format: NavKalpana-TeamCode
   Example: NavKalpana-RICR-NK-0001
3. All team members must:
    - Must be the contributor in repo
    - Use individual GitHub accounts
    - Contribute via commits
    - Avoid single-person commits for team projects

Commit Rules

- Minimum 10 meaningful commits.
- Avoid a single bulk commit at the end.
- Commit messages must be clear.
    - Good Commit Example:
        - Added login authentication
        - Integrated payment API
        - Fixed cart validation bug

- Bad Commit Example:
  - Update
  - Final
  - Added
  - done

Bad Commit Messages will lead to negative marking

Branching Rule (Recommended)

- main – Final stable version
- All the Members must Create Separate Branches which further merges to the Main Branch
  - Example:
    - feature/login
    - feature/register
    - bug/update etc.

3. Recommended Project Folder Structure

Repository Name Example: *NavKalpana-RICR-NK-0001*

Standard Structure

- NavKalpana-TeamCode
  - frontend
  - backend
  - docs
    - problem-statement.pdf
    - architecture-diagram.png
    - api-documentation.md
    - presentation.pptx
  - README.md

4. README.md Must Include

- Project Title
- Team Members & Roles
- Problem Statement
- Tech Stack Used
- Installation Steps
- API Endpoints
- Screenshots
- Future Improvements

5. What Judges Will Verify

Technical Verification

- Git commit history
- Contribution by all members
- Code understanding

- Folder structure clarity
- Backend API functionality
- Database integration
- Error handling

Innovation Check
Judges will evaluate:
- Whether it solves a real problem
- Practical usability
- Scalability potential
- Meaningful use of AI (if included)

Viva Round Questions
Each member may be asked:
- Explain your module.
- Explain database schema.
- Explain API flow.
- Explain deployment steps.
- Explain one technical challenge you faced.

6. Important Warning
Immediate disqualification may occur if:
- Code is copied from the internet without modification
- Purchased templates are used without proper logic implementation
- Repository is created only on the final day
- Only UI is presented without functionality
- Fake data is shown without backend logic (unless clearly defined prototype)
- Only one member answers all questions

7. Bonus Considerations
- Clean UI/UX
- Proper Authentication
- Role-based access control
- AI/ML Integration (if required)
- Deployment (Vercel, Render, AWS, Azure)
- Clean code practices

Final Note
The hackathon is not only about coding. It is about teamwork, structured thinking, problem solving, and clear communication. Build a meaningful solution and be prepared to explain it confidently.

CardioShield AI
AI-Powered Early Cardiovascular Risk Stratification Platform

1. Executive Summary

CardioShield AI is an explainable, low-cost, deployable AI platform designed for early risk prediction of cardiovascular diseases (CVD) — specifically heart attacks and strokes — in underserved and resource-constrained populations.
The system enables:
• High-accuracy 5–10 year CVD risk prediction
• Point-of-care screening using low-cost clinical inputs
• Real-time explainable risk scoring
• Bias-aware model calibration for South Asian populations
• Offline-capable deployment in rural health settings
CardioShield AI operates as a structured health intelligence loop:
Input → Clean → Engineer → Predict → Calibrate → Explain → Stratify → Act
This platform bridges the gap between early screening and specialist intervention.

2. Product Vision

To reduce premature cardiovascular mortality in underserved populations by delivering:
• Affordable AI-based risk screening
• Clinically interpretable risk explanations
• Bias-aware decision support
• Scalable population-level screening
• Deployment-ready architecture for rural clinics
CardioShield AI is not a diagnostic replacement.
It is an early-warning risk stratification system.

3. Target Users

Primary Users:
• Rural Primary Health Centers (PHCs)
• Community Health Workers
• Government screening programs
• Insurance risk stratification teams
• Telehealth providers
End Beneficiaries:
• Adults aged 30–65
• High-risk South Asian populations
• Low-income & underserved communities

4. System Architecture

CardioShield AI consists of:
PART 1 – Core Risk Stratification Engine (~70%)
PART 2 – Advanced Clinical Intelligence, Ethics & Deployment Layer (~30%)

PART 1 – CORE RISK STRATIFICATION ENGINE (70%)
This layer handles the predictive intelligence backbone.

5. Data Input & Preprocessing Engine

5.1 Dataset Link GitHub

https://github.com/ricrbhopal/PythonDataSets/blob/main/CardioShieldDataSet.csv

5.2 Supported Inputs (Low-Cost Only)

Features:

1. Age | Objective Feature | age | (days)
2. Height | Objective Feature | height | (cm) |
3. Weight | Objective Feature | weight | (kg) |
4. Gender | Objective Feature | gender | categorical code |
5. Systolic blood pressure | Examination Feature | ap_hi |
6. Diastolic blood pressure | Examination Feature | ap_lo |
7. Cholesterol | Examination Feature | cholesterol | 1: normal, 2: above normal, 3: well above normal |
8. Glucose | Examination Feature | gluc | 1: normal, 2: above normal, 3: well above normal |
9. Smoking | Subjective Feature | smoke |
10. Alcohol intake | Subjective Feature | alco |
11. Physical activity | Subjective Feature | active |
12. Presence or absence of cardiovascular disease | Target Variable | cardio |

5.3 Data Cleaning & Robust Preprocessing

Outlier detection
Missing value imputation
Normalization
Noise simulation (rural measurement variability)

5.4 Imbalance Handling Strategy

Problem:
High-risk cases are rare → false negatives deadly.

- over sampling / under sampling
- Cost-sensitive learning
- Model Performance

Evaluation must explicitly justify:
Synthetic sample risks
False negative trade-offs

6. Feature Engineering Engine

Advanced transformations:

- Pulse pressure (SBP − DBP)
- BMI risk categories
- Age × BP interaction

• Glucose × BMI interaction
• Composite risk indices
Rationale must align with clinical logic.

7. Model Development Engine
Participants must train and compare ≥4 models:
• XGBoost
• LightGBM
• Neural Network
• TabNet
• Ensemble model

7.1 Performance Targets
ROC-AUC ≥ 0.92
Recall ≥ 0.85 (critical)
Precision ≥ 0.80
Strong PR-AUC
Why and which class Recall prioritized over accuracy.

7.2 Calibration Engine
Raw probabilities calibrated using:
Platt Scaling
Isotonic Regression
Output categories:
Low Risk
Moderate Risk
High Risk
Clinically interpretable thresholds.

8. Risk Stratification Output Layer
Each patient receives:
Predicted 5–10 year risk probability
Risk category
Confidence interval
Top contributing risk factors

9. Explainability Engine (Mandatory)
9.1 Global Explainability
Feature importance
SHAP summary plots

9.2 Individual-Level Explanation
SHAP values
Counterfactuals

System must answer:
“Why is this patient high-risk?”
Example output:
“High systolic BP (+32%), elevated glucose (+18%), and smoking (+12%) significantly increase predicted risk.”
Medical alignment mandatory.

10. Core Clinical Dashboard
Displays:
Risk score
Risk category
Top risk drivers
Recommended preventive steps
Referral suggestion
Clean UI for clinicians.