

-
- Title: Heart Attack Risk Checkup System
 - Subtitle: Python Application for Individual Heart Risk Assessment
 - Author(s): AMAN NAWAZ
 - Date: 24-11-2025
-

1. Introduction

This Python project provides a simple, interactive terminal application to help users assess their risk level for a heart attack based on basic clinical and lifestyle inputs. The system aims to increase awareness and encourage early self-screening for cardiovascular health.[ncbi.nlm.nih+1](#)

2. Problem Statement

Cardiovascular disease is a leading cause of death globally. Many people lack immediate access to basic risk screening. This tool aims to address that gap by enabling users to estimate their heart attack risk using routine, self-reported data and highlighting the need for early medical intervention where appropriate.[betterhealth+1](#)

3. Functional Requirements

- Collect user input for age, symptoms, clinical history, and lifestyle factors.
 - Calculate a heart attack risk score using a rule-based approach.
 - Categorize risk levels: Low, Moderate, High.
 - Provide outcome messages and recommendations.
 - Store user results in an Excel file for future reference.
-

4. Non-functional Requirements

- User-friendly console prompts and error handling.
 - Data privacy by storing results locally.
 - Portable across platforms supporting Python and required packages.
 - Scalable for modifying the risk logic or expanding data fields.
-

5. System Architecture

- Input Layer: User prompt and input validation.
 - Processing Layer: Risk calculation logic.
 - Output Layer: Result display and Excel storage.
-

6. Design Diagrams

Use Case Diagram

- Actors: User
- Use Cases: Input Data, Review Result, Save to Excel

Workflow Diagram

- Start → Input Data → Calculate Risk → Display Result → Save Data → End

Sequence Diagram

- User requests check → System prompts for data → User inputs data → System calculates and displays result → System saves to file

Class/Component Diagram

- Main module: heart_attack_check()
- Subcomponents: Input management, scoring logic, record storage

ER Diagram (for storage)

- Table: heart_risk_data
 - Fields: Age, Chest Pain, Systolic BP, Diastolic BP, Cholesterol, Diabetes, Family History, Physical Activity, Obesity, Smoking, Risk Level

7. Design Decisions & Rationale

- Chosen rule-based scoring for simplicity and transparency.
- Pandas for data manipulation and Excel I/O.
- Console interface for maximum portability.

8. Implementation Details

- All code provided is in Python.

- Uses pandas for Excel management.
 - os for file existence checks.
 - Try-except blocks ensure correct input types and user experience.[geeksforgeeks+1](#)
-

9. Testing Approach

- Manual input test cases for each risk category.
 - Validation of Excel output structure.
 - Error handling for invalid data.
-

10. Challenges Faced

- Balancing simplicity with risk factor detail.
 - Validating risk scoring against accepted medical guidelines.[tools.acc+1](#)
 - Ensuring data integrity in Excel records.
-

11. Learnings & Key Takeaways

- Importance of clear clinical criteria in automated health screening.[ncbi.nlm.nih](#)
 - Effective user messaging for health apps.
 - Python's strength in rapid prototyping and data manipulation.
-

12. Future Enhancements

- GUI or web interface for broader reach.
 - Integration with medical databases or APIs for validation.
 - Secure anonymous data aggregation for population-level insights.
 - More nuanced scoring with additional clinical data.
-

13. References

- Heart risk assessment standards[ncbi.nlm.nih+2](#)
 - Python and Pandas documentation[w3schools+1](#)
 - Cardiac risk guidelines from ACC/AHA[ncbi.nlm.nih](#)
-

This outline covers all major sections you requested. For the actual PDF, each section can be expanded with code snippets, sample terminal outputs, and annotated diagram images generated using UML tools or drawing platforms. Citations are collected from up-to-date web resources and technical references to ensure accuracy.

1. <https://healthify.nz/health-a-z/h/heart-risk-assessment>
2. <https://www.jeeviacademy.com/heart-diseases-prediction-machine-learni/>
3. <https://www.ncbi.nlm.nih.gov/books/NBK537146/>
4. <https://www.scribd.com/document/521054269/Heartdisease>
5. <https://tools.acc.org/ascvd-risk-estimator-plus/>
6. <https://www.geeksforgeeks.org/machine-learning/ml-heart-disease-prediction-using-logistic-regression/>
7. <https://www.ncbi.nlm.nih.gov/books/NBK554923/>
8. <https://www.geeksforgeeks.org/python/pandas-step-by-step-guide/>
9. <https://www.betterhealth.vic.gov.au/health/conditionsandtreatments/heart-disease-and-stroke-your-risk-score>
10. https://www.w3schools.com/python/pandas/pandas_dataframes.asp