VIT®

BHOPAL

CSE PROJECT VITYARTHI

Blood Pressure Monitoring Record System

NAME- MANSI KASHYAP

DATE- 25-11-25

REGISTRATION NUMBER - 25BAI10577

INTRODUCTION

Many individuals find it challenging to consistently track and interpret their blood pressure readings at home. This lack of reliable monitoring can lead to missed warning signs, poor health management, and insufficient information for medical consultations. Furthermore, most existing solutions are often either too complex, unavailable, or not suited for everyday users.

This project addresses these issues by delivering a simple, accessible, and user-friendly system to record, categorize, and analyze blood pressure measurements. Using American Heart Association (AHA) guidelines, users can input readings and instantly receive categorization results. By storing historical data, the application provides summary statistics and trends, making health tracking more actionable.

Such a tool enables individuals to better understand their cardiovascular health, improve planning around daily activities and medical appointments, and reduce anxiety about blood pressure management. By empowering users with clear insights and easy-to-use features, the system promotes proactive wellness and encourages routine attention to heart health.

PROBLEM STATEMENT

Many individuals struggle to reliably track and understand their blood pressure readings, which can result in missed signs of hypertension and incomplete information for healthcare management. Without a simple tool to log and categorize readings, people may face difficulty planning daily activities and making informed decisions about their health. Most existing solutions are either too complex, inaccessible, or lack clear feedback for everyday users.

This project seeks to deliver an easy-to-use system that enables users to record, categorize, and analyze their blood pressure readings at home. By utilizing American Heart Association (AHA) guidelines and summarizing personal data, the tool offers personalized insights into cardiovascular health. Users receive instant classification for every measurement and can observe trends over time.

With better data and clear analysis, individuals can monitor their health more effectively, reduce anxiety related to blood pressure management, and support proactive planning for medical appointments or daily routines. This straightforward and reliable solution promotes regular wellness tracking and encourages greater awareness of hypertension risks.

OBJECTIVES

- To develop a tool that accurately records blood pressure readings and dates, allowing users to maintain a personal health history.
- To classify each reading according to American Heart Association (AHA) guidelines, supporting clear understanding of individual cardiovascular status.
- To provide automated, data-driven analysis with personalized summaries, enabling users to monitor average, minimum, and maximum readings over time.
- To support users in personal health management, helping them track trends and better prepare for medical consultations, daily activities, and appointments.
- To raise awareness and promote better heart health by making blood pressure tracking simple and actionable.
- To create a user-friendly interface that can be easily used by individuals, including those with limited technical experience.

FUNCTIONAL REQUIREMENTS

- To allow users to add new blood pressure readings with systolic and diastolic values, and save the date of each measurement.
- To automatically categorize each reading according to American Heart Association (AHA) guidelines, providing instant feedback on blood pressure status.
- To store and display all entered records in a structured and easily accessible list, sorted by date.
- To calculate and show important statistics such as average, maximum, and minimum systolic and diastolic readings based on user data.
- To enable users to view personalized trends and summaries for better health tracking and planning.
- To validate all user input, ensuring data integrity through checks for realistic blood pressure values and proper date formats.
- To provide a friendly and intuitive command-line interface, making the tool simple to use for individuals with any level of technical background

NON FUNCTIONAL REQUIREMENTS

- The system should provide fast performance, allowing users to add and review records without noticeable delay.
- Data entered must remain reliable and consistent, with processes in place to prevent loss or corruption.
- The application should be easy to install, requiring only a standard Python environment with no extra dependencies.
- The user interface must be clear and intuitive, supporting users of all technical backgrounds.
- The tool should maintain privacy and security of personal health data by keeping all records local.
- The system should be robust, handling invalid inputs gracefully and minimizing errors or crashes during use.
- The application must be maintainable, with code that is well-documented and structured for future updates or troubleshooting.

SYSTEM ARCHITECTURE

Overview

The Blood Pressure Monitoring System is a Python-based application that runs locally, providing users with a simple and efficient way to record, categorize, review, and analyze blood pressure readings through a command-line interface.

Components

- Input Module: Captures new blood pressure readings and dates from the user.
- Validation Module: Checks data integrity and ensures realistic values.
- Storage Module: Saves all records in an internal list structure for retrieval and analysis.
- Analysis Module: Calculates statistics (average, max, min) and automatically classifies each reading using American Heart Association (AHA) guidelines.
- Output Module: Displays results, records, and summaries to the user via text-based menus.

Data Flow

1. User enters a blood pressure reading.
2. Input is validated for correctness.
3. Record is stored in the local list.
4. Analysis module evaluates category and statistics.
5. Output module presents information back to the user.

Extensibility

- The modular code design allows easy addition of new analysis types or data exports.
- Storage can be adapted to external files or databases as needed.
- The user interface can be redesigned for graphical or web-based interaction in future versions.
- New health metrics or integration with medical APIs can be added without major changes to core logic.

DESIGN DIAGRAMS

Use Case Diagram

- User interacts with the system to:
  - Add blood pressure readings
  - View all records
  - Analyze statistical summaries
  - Exit the application

Class Diagram

- BPMonitor
  - Attributes: records
  - Methods: add_record(), view_records(), analyze_stats(), get_bp_category(), load_sample_data(), run_cli()
- Record
  - Attributes: sys (systolic), dia (diastolic), date

Sequence Diagram

- User selects 'Add Reading' → System prompts for values → User enters data → System validates → System saves record → System displays category and confirmation

Flowchart

- Start → Show menu → User selects option
  - If Add: Input values → Validate → Save → Categorize → Confirm
  - If View: Retrieve records → Display
  - If Analyze: Compute stats → Display
  - If Exit: Terminate

Workflow Diagram

- Data flows from user input through validation and storage modules; analysis module fetches data and computes stats; output presented via CLI interface. All processes are contained within a local runtime and can be extended for more features or integration.

DESIGN DECISIONS & RATIONALE

Simplicity and Accessibility

- Chose a straightforward command-line interface so users of all technical skill levels can record and review data without difficulty.
- Avoided complex installation requirements to maximize accessibility for diverse users.

Data-Driven Approach

- Incorporated American Heart Association (AHA) guidelines for automatic categorization, providing medically relevant feedback on each reading.
- Designed statistical analysis modules to deliver meaningful summaries such as averages and trends.

Modular Design

- Structured the code into distinct modules for input, validation, storage, analysis, and output.
- This modular approach supports maintainability and future enhancements without affecting existing functionality.

Handle Data Variability

- Implemented robust input validation to filter out unrealistic readings and ensure reliable statistics.
- Support for flexible date inputs enables comprehensive and personalized record-keeping.

Privacy and Security Considerations

- All records are maintained locally, avoiding exposure to cloud risks or third-party access.
- No sensitive information leaves the user's device, supporting privacy for personal health data.

Future Proofing

- Class and method structures are designed to allow easy integration of new features, such as exporting data, connecting to external health APIs, or adding new metrics.
- Plans for adapting the interface or storage layer as user needs evolve without major rework.

IMPLEMENTATION DETAILS

Programming Language and Environment

- Developed in Python 3.x, leveraging its built-in modules for simplicity and compatibility.
- Runs locally on any operating system with a standard Python interpreter—no third-party packages required.

Core Algorithm

- Implements American Heart Association (AHA) guidelines to categorize blood pressure readings based on systolic and diastolic values.
- Aggregates statistical measures (average, min, max) directly from the stored data for real-time analysis.

Input & Output

- Accepts user input for systolic, diastolic values, and date through a command-line interface.
- Provides textual outputs including categorization of readings, tabulated record listings, and summary statistics.

Data Handling

- Stores all blood pressure records in a Python list of dictionaries, ensuring fast access and modification.
- Validates each input for data type and realistic range before storage, ensuring data integrity.

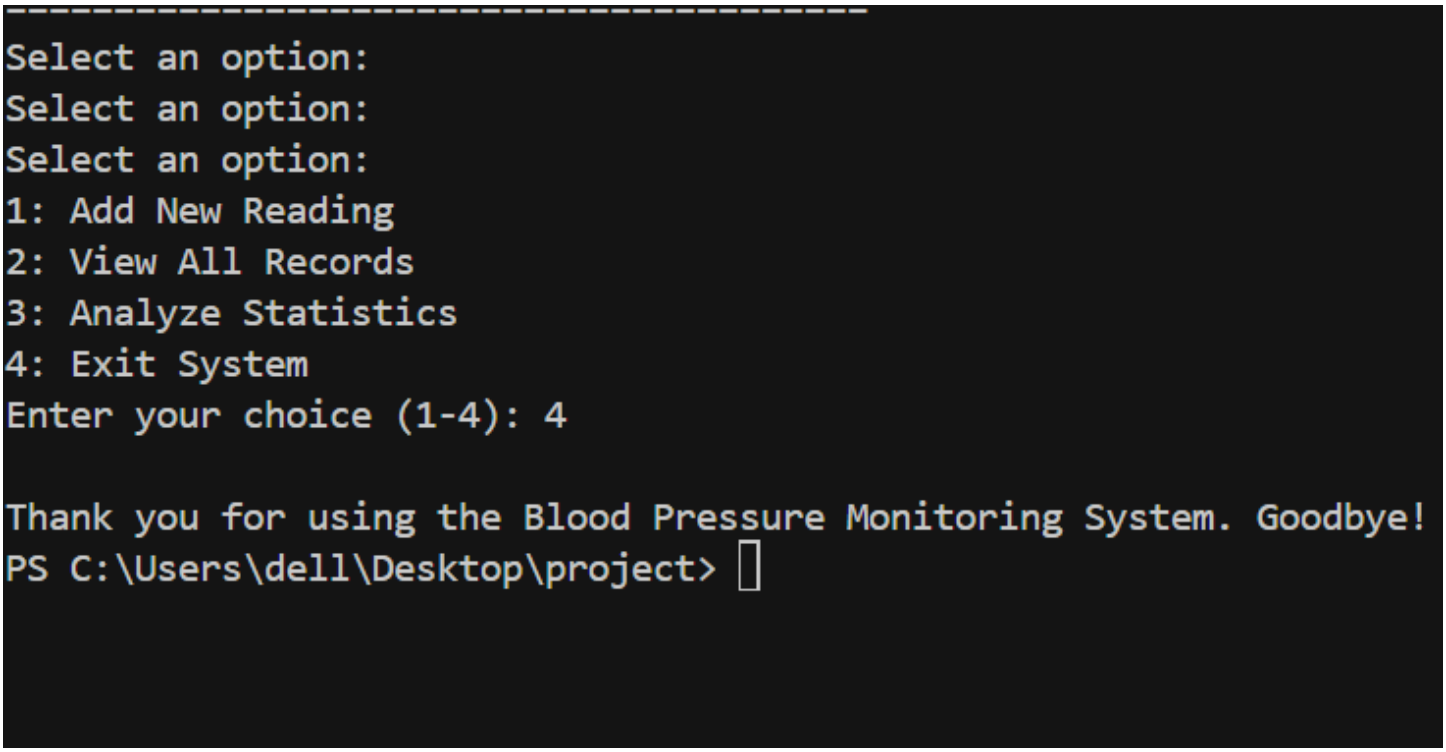
Extensibility & Improvements

- Modular structure allows for the addition of new features such as exporting data to files, integrating charts, or expanding to a graphical or web interface.
- Can be enhanced to support external storage (databases or cloud), health data import/export, or customized alert logic.

SCREENSHOTS / RESULTS

```
✅ Record added successfully on 2025-11-24. Category: Stage 1 Hyper
tension

Select an option:
1: Add New Reading
2: View All Records
3: Analyze Statistics
4: Exit System
Enter your choice (1-4): 2

================================================
         BLOOD PRESSURE MONITORING RECORDS
================================================
Date         Systolic   Diastolic  Category
------------------------------------------------
2025-11-24   120        80         Stage 1 Hypertension
2025-11-24   120        80         Stage 1 Hypertension
2025-10-23   105        65         Normal
2025-10-22   135        90         Stage 2 Hypertension
2025-10-21   118        75         Normal
2025-10-20   125        82         Stage 1 Hypertension
------------------------------------------------

Select an option:
1: Add New Reading
2: View All Records
3: Analyze Statistics
4: Exit System
Enter your choice (1-4): 2
```

```
================================================
         BLOOD PRESSURE MONITORING RECORDS
================================================
Date         Systolic   Diastolic  Category
------------------------------------------------
2025-11-24   120        80         Stage 1 Hypertension
2025-11-24   120        80         Stage 1 Hypertension
2025-10-23   105        65         Normal
2025-10-22   135        90         Stage 2 Hypertension
2025-10-21   118        75         Normal
2025-10-20   125        82         Stage 1 Hypertension
------------------------------------------------

Select an option:
1: Add New Reading
2: View All Records
3: Analyze Statistics
4: Exit System
Enter your choice (1-4): 3

========================================
          ANALYSIS SUMMARY
========================================
Total Readings: 6

Metric          Systolic   Diastolic
----------------------------------
Average:        120.5:<10 78.7:<10
Max:            135        90
Min:            105        65
----------------------------------
Overall Status based on Average: Elevated
========================================
```


Select an option:
Select an option:
Select an option:
1: Add New Reading
2: View All Records
3: Analyze Statistics
4: Exit System
Enter your choice (1-4): 4
Thank you for using the Blood Pressure Monitoring System. Goodbye!
PS C:\Users\dell\Desktop\project>

TESTING APPROACH

Unit Testing

- Test individual methods such as categorization, record addition, statistics calculation, and input validation to ensure each module operates correctly in isolation.

Integration Testing

- Validate interaction between modules (input, storage, analysis, output) by simulating typical user workflows, ensuring seamless end-to-end functionality.

Boundary Testing

- Check system behavior for edge case values like minimum/maximum valid blood pressure readings, dates, and input formats to verify data integrity and error handling.

Manual Testing

- Perform hands-on verification of the command-line interface, adding, viewing, and analyzing records as an actual user would to spot usability and logical errors.

Performance & Usability

- Assess responsiveness for large datasets and confirm the system remains fast, user-friendly, and clear when processing multiple entries or abnormal usage patterns.

Future Automated Testing

- Structure code and logic to support the addition of automated test suites, including regression tests, code coverage analysis, and interface simulation.

CHALLENGES FACED

Input Validation

- Ensuring robust checks for systolic/diastolic value ranges, avoiding both false positives and missed errors.

User Interface Simplicity

- Designing a clear and accessible command-line UI for users with different technical backgrounds.

Data Integrity

- Handling missing, inconsistent, or incorrectly formatted data entries.

Medical Classification

- Implementing accurate categorization based on medical standards while remaining understandable for general users.

Privacy & Security

- Guaranteeing that health data remains private on local machines and is not shared externally.

Scalability

- Structuring code to allow easy future integration with external databases or modules without affecting current performance.

Usability Testing

- Addressing manual entry errors and ensuring outputs are actionable, clear, and suitable for non-medical users.

Workflow Robustness

- Covering edge cases, boundary values, and unusual user actions to prevent crashes and unexpected behavior.

LEARNINGS & KEY TAKEAWAYS

Technical Development

- Enhanced Python programming and modular design skills.

Data Validation

- Gained experience in building strong boundary and input checks.

User Experience

- Learned the impact of simple, clear interfaces for wider accessibility.

Privacy

- Understood best practices for keeping local health data secure.

Extensibility

- Saw how flexible code structures support easy future improvements.

Real-World Impact

- Recognized how accessible health tools empower individuals for better daily wellness.

FUTURE ENHANCEMENTS

Data Export

- Add options to save and share blood pressure records as files or documents.

Graphical Interface

- Develop a user-friendly GUI or mobile app version for improved accessibility.

Cloud Integration

- Enable cloud backup or syncing across devices to ensure data availability and security.

Advanced Analytics

- Integrate trend visualization and personalized alerts for abnormal readings.

API Integration

- Connect with medical APIs or electronic health records for seamless data sharing.

Wearable Device Support

- Expand compatibility to sync readings from smartwatches and health devices.

Machine Learning

- Explore predictive analytics for risk assessment or personalized advice.

REFERENCES

Libraries

- Python standard library (datetime, sys)
- Example: Gradio for Python GUI development

Tutorials and Guides

- GeeksforGeeks: Blood Pressure Analysis GUI Using Gradio in Python
- GitHub blood pressure monitoring projects and open-source frameworks

Sample Datasets

- MIMIC-BP: Curated biomedical signal dataset for blood pressure estimation

API Documentation

- Python official documentation (docs.python.org)
- Medical device API integration resources

Research Papers or Articles

- CDC: Self-Measured Blood Pressure Monitoring—Program Planning, Implementation, and Lessons Learned
- Novel Digital Technologies for Blood Pressure Monitoring and Management

Development Tools

- Python 3.x Interpreter
- Visual Studio Code, Jupyter Notebook, Git