**AI Project**



**Session 2023-2027**

**Submitted By:**

Muhammad Harris Amin (2023-CS-160)

Muhammad Tayyab Amir (2023-CS-101)

**Supervised by:**

Muhammad Kabir Ahmed

**University of Engineering and Technology, Lahore**

**Advanced Health Data Visualization Tool**

**Comprehensive Documentation**

**Table of Contents**

1. [Overview](#overview)
2. [System Architecture](#system-architecture)
3. [Installation](#installation)
4. [User Interface](#user-interface)
5. [Visualization Categories](#visualization-categories)
6. [Technical Implementation](#technical-implementation)
7. [Development Guide](#development-guide)
8. [Future Enhancements](#future-enhancements)
9. [Troubleshooting](#troubleshooting)
10. [References](#references)

**1. Overview**

The Advanced Health Data Visualization Tool is a comprehensive software application designed to provide insightful visualizations of health-related data. This application shows the power of Python's data science libraries (NumPy, Pandas, Matplotlib, Seaborn) with a user-friendly PyQt5 graphical interface to help users explore and analyze health metrics.

The application is designed for healthcare professionals, researchers, and data analysts who need to visualize relationships between various health parameters such as BMI, heart rate, exercise habits, sleep patterns, and lifestyle factors.

**Key Features:**

* Interactive data visualizations across five categories
* Collapsible sidebar for easy navigation
* Customizable visualization panels
* Export functionality for reports and presentations
* Comprehensive analysis of health metrics correlations

**2. System Architecture**

The application follows a modular architecture with clear separation of concerns:

**Core Components:**

1. **Data Layer**: Pandas for data management and preprocessing
2. **Analysis Layer**: NumPy for numerical operations
3. **Visualization Layer**: Matplotlib and Seaborn for generating plots
4. **Presentation Layer**: PyQt5 for the graphical user interface

**Component Diagram:**

┌────────────────┐ ┌───────────────┐ ┌────────────────┐

│ │ │ │ │ │

│ Data Sources │────▶│ Data Manager │───▶│ Visualization │

│ (Dataset) │ │ (Pandas) │ │ Engine │

│ │ │ │ │ │

└────────────────┘ └───────────────┘ └────────┬───────┘

│

▼

┌───────────────┐ ┌────────────────┐

│ │ │ │

│ User Input │◀────│ UI Manager │

│ Handler │ │ (PyQt5) │

│ │ │ │

└───────────────┘ └────────────────┘

**3. Installation**

**System Requirements:**

* Python 3.7 or higher
* Windows, macOS, or Linux operating system
* 4GB RAM minimum (8GB recommended)
* 500MB free disk space

**Dependencies:**

* NumPy (1.20.0+)
* Pandas (1.3.0+)
* Matplotlib (3.4.0+)
* Seaborn (0.11.0+)
* PyQt5 (5.15.0+)

**Installation Steps:**

1. **Create a virtual environment:**

bash

python -m venv healthviz-env

1. **Activate the virtual environment:**
   * Windows: healthviz-env\Scripts\activate
   * macOS/Linux: source healthviz-env/bin/activate
2. **Install required packages:**

bash

pip install numpy pandas matplotlib seaborn PyQt5

1. **Clone or download the application:**

bash

git clone https://github.com/username/health-visualization-tool.git

1. **Run the application:**

bash

python main.py

**4. User Interface**

The application features a modern, intuitive interface designed for ease of use and productivity:

**Main Components:**

1. **Header Bar**: Contains the application title, logo, and navigation controls
2. **Collapsible Sidebar**: Provides categorical access to all visualizations
3. **Visualization Panel**: Displays the selected visualization
4. **Toolbar**: Offers interaction with the current visualization (zoom, pan, save)
5. **Footer**: Provides access to help and exit options

**UI Features:**

* **Toggling Sidebar**: Use the ≡ button to show/hide the sidebar for more visualization space
* **Collapsible Categories**: Click on category headers to expand/collapse visualization options
* **Interactive Plots**: Many visualizations support hovering for additional information
* **Export Options**: Save visualizations as images for reports or presentations
* **Help Documentation**: Access comprehensive guidance through the help button

## 5. Visualization Categories

The application organizes visualizations into five categories, each designed to address specific analytical needs:

**5.1 Basic Distributions**

Visualizations that show the distribution of single variables:

* **Age Distribution**: Histogram showing the age distribution of subjects
* **BMI Distribution**: Distribution of Body Mass Index across the dataset
* **Sleep Hours Distribution**: Analysis of sleep patterns
* **Daily Steps Distribution**: Distribution of activity levels via step count
* **Heart Rate Distribution**: Distribution of resting heart rates

**5.2 Comparative Analysis**

Visualizations that compare metrics across different groups:

* **BMI by Gender**: Box plot comparing BMI distribution between genders
* **Exercise by Smoker Status**: Compare exercise habits between smokers and non-smokers
* **Sleep by Age Group**: Analysis of sleep patterns across different age groups
* **Alcohol Consumption by Gender**: Compare drinking habits between genders
* **Heart Rate by Diabetic Status**: Compare heart rates between diabetic and non-diabetic individuals

**5.3 Correlation Analysis**

Visualizations that explore relationships between multiple variables:

* **Health Metrics Heatmap**: Correlation matrix of key health indicators
* **Lifestyle vs Vitals Correlation**: How lifestyle choices correlate with vital signs
* **Clustermap of All Variables**: Hierarchically clustered heatmap of correlations
* **Age vs Health Indicators**: How health metrics change with age
* **Exercise Impact Analysis**: Correlation between exercise and other health metrics

**5.4 Multivariate Analysis**

Visualizations exploring relationships between multiple variables simultaneously:

* **Daily Steps vs BMI**: Scatter plot examining relationship between activity and BMI
* **BMI vs Age**: How BMI changes with age
* **Alcohol vs Heart Rate**: Impact of alcohol consumption on heart rate
* **Sleep vs Exercise**: Relationship between sleep and physical activity
* **Calorie Intake vs Weight**: How caloric intake relates to body weight

**5.5 Advanced Visualizations**

Complex visualizations that provide deeper insights:

* **FacetGrid: Health Metrics by Gender**: Comparison of multiple metrics split by gender
* **Health Radar Chart**: Multi-dimensional comparison of health metrics
* **Comprehensive Health Dashboard**: Overview of multiple health indicators
* **Risk Factors Sunburst Chart**: Hierarchical representation of health risk factors
* **3D Health Analysis**: Three-dimensional analysis of age, BMI, and heart rate

**6. Technical Implementation**

**6.1 Data Processing**

The application uses Pandas for data handling:

* Data loading from CSV files
* Data cleaning and normalization
* Feature extraction (e.g., systolic/diastolic from blood pressure)
* Categorical binning (e.g., age groups, BMI categories)

**6.2 Visualization Implementation**

The visualization module leverages Matplotlib and Seaborn:

* **Figure Management**: Each visualization function returns a Matplotlib figure
* **Color Schemes**: Consistent color palettes for clarity and accessibility
* **Interactive Elements**: Hover tooltips and dynamic elements
* **Layout Control**: Tight layout management for optimal display

**6.3 UI Implementation**

The PyQt5 implementation follows these design patterns:

* **Model-View Architecture**: Clear separation between data and presentation
* **Event-Driven Programming**: User interactions trigger appropriate responses
* **Responsive Design**: Layouts adapt to window resizing
* **Collapsible Widgets**: Custom implementation of collapsible sections

**Key UI components:**

* **QMainWindow**: Main application window
* **QSplitter**: Resizable division between sidebar and visualization panel
* **QStackedWidget**: Manages switching between different visualizations
* **CollapsibleSection**: Custom widget for expandable/collapsible categories

**7. Development Guide**

**7.1 Project Structure**

health-visualization-tool/

├── main.py # Application entry point

├── Dataset.py # Data loading and preprocessing

├── visualization.py # Visualization functions

├── data/

│ └── health\_data.csv # Sample health dataset

├── assets/

│ ├── icons/ # UI icons

│ └── styles/ # CSS styling files

└── docs/

└── documentation.md # This documentation

**8. Future Enhancements**

Planned features for future versions:

1. **Data Import/Export**: Allow users to load their own datasets
2. **Custom Visualization Creation**: User-defined visualization options
3. **Statistical Analysis**: Add advanced statistical tests and modeling
4. **Predictive Analytics**: Incorporate machine learning models for health prediction
5. **Report Generation**: Automated report creation with key findings
6. **User Profiles**: Save and load user preferences and settings
7. **Collaborative Features**: Share visualizations and insights with team members

**9. Troubleshooting**

**Common Issues and Solutions:**

1. **Application fails to start**:

* Verify all dependencies are installed correctly
* Check Python version compatibility
* Ensure no port conflicts with other applications

1. **Visualizations not displaying**:

* Check for error messages in the console
* Verify the data file exists and is properly formatted
* Ensure visualization functions are properly implemented

1. **UI responsiveness issues**:

* Complex visualizations may take time to render with large datasets
* Consider using data sampling for improved performance
* Optimize visualization functions for speed

1. **Export functionality problems**:

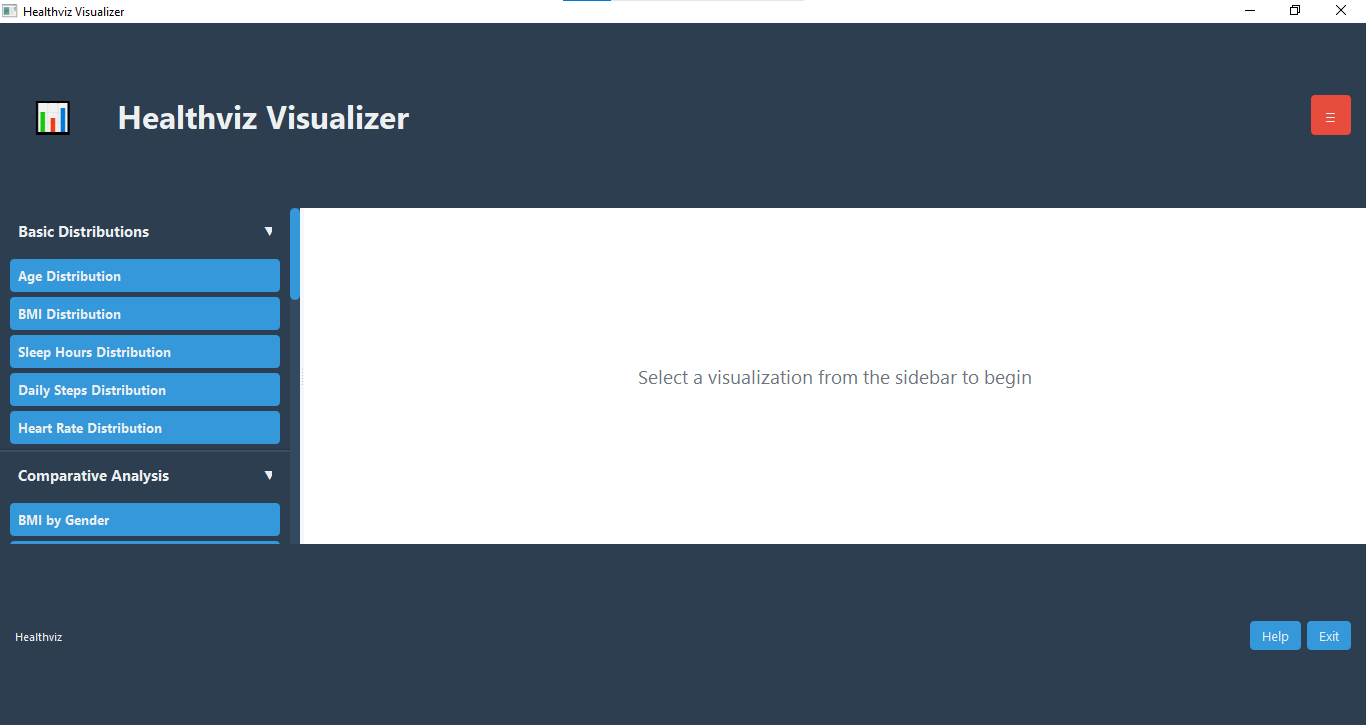
* Ensure write permissions to the destination folder
* Check for adequate disk space
* Verify Matplotlib backend compatibility with your system

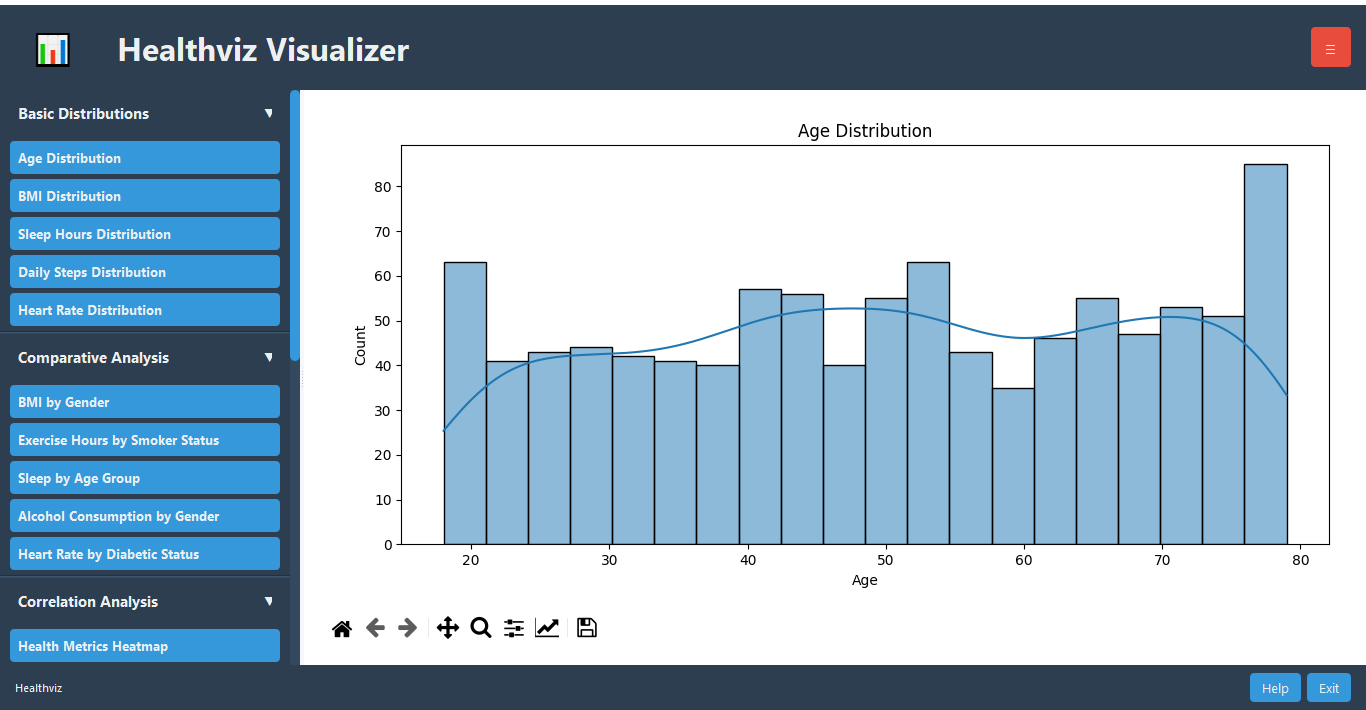
**10. References**

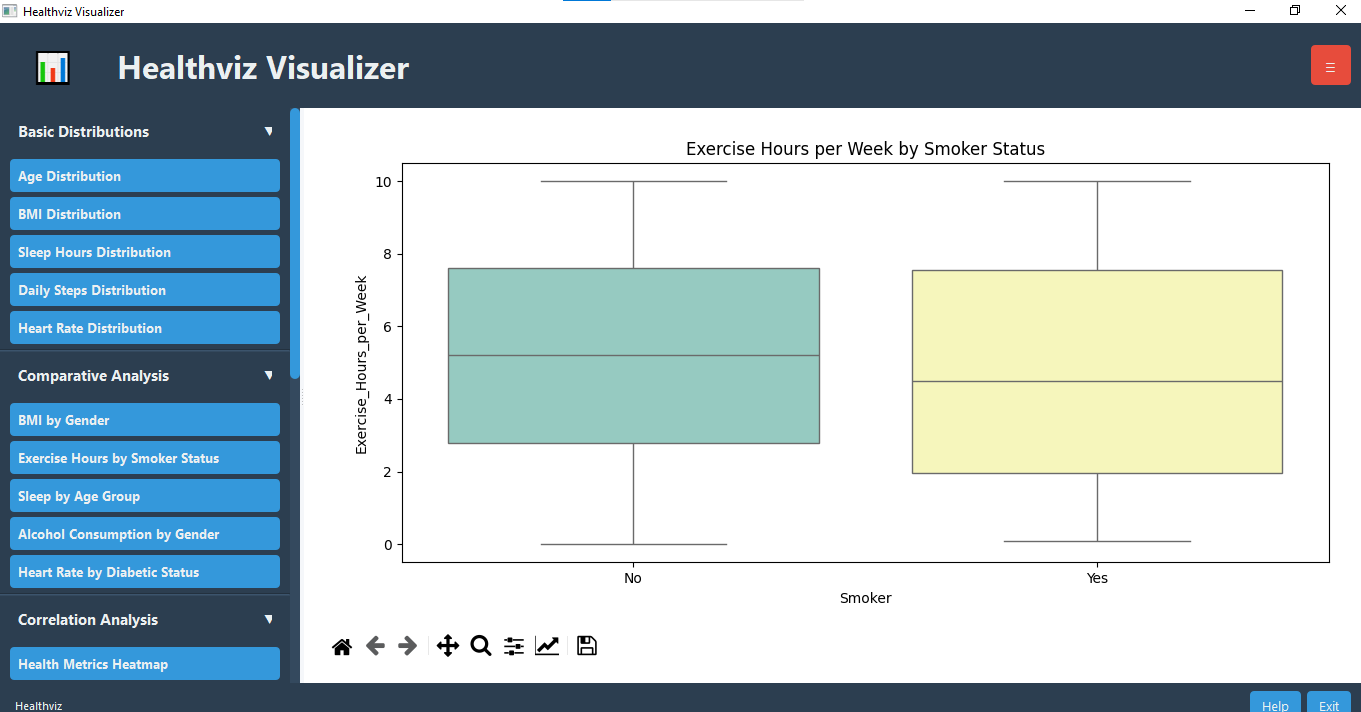
**Libraries Documentation:**

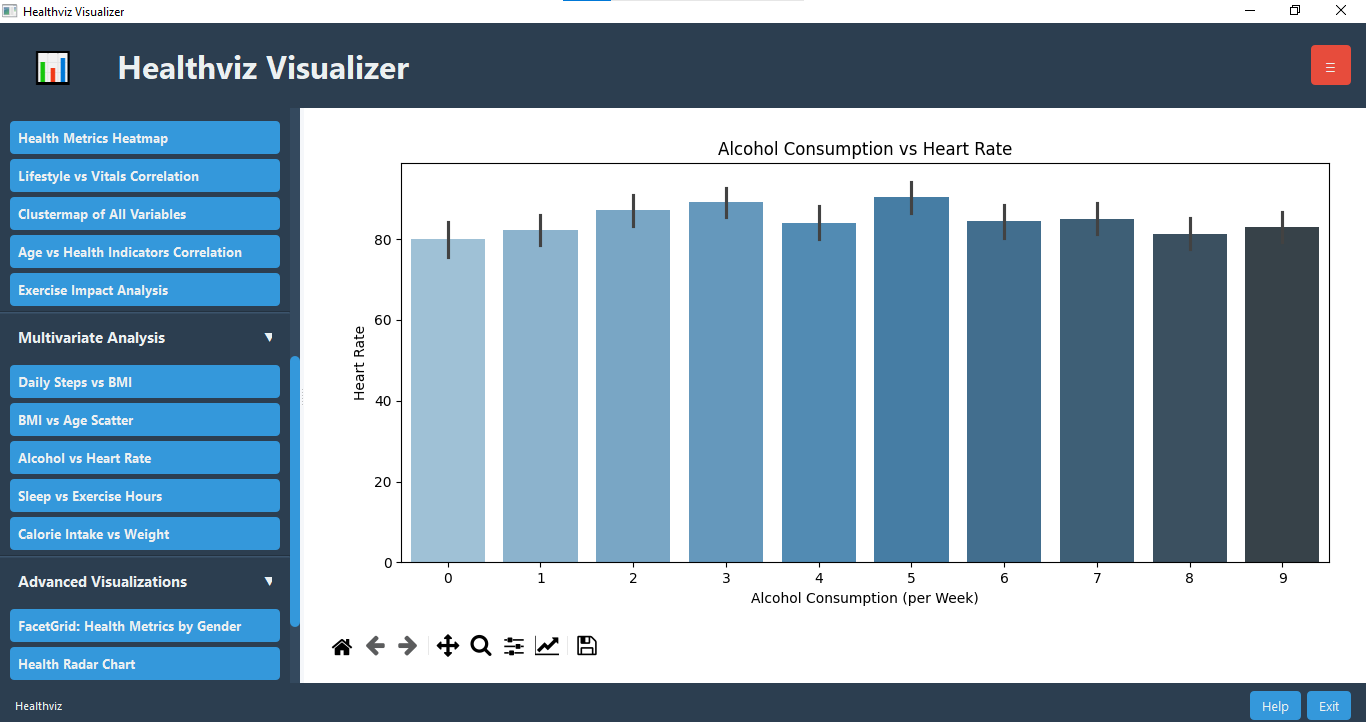
* [NumPy Documentation](https://numpy.org/doc/)
* [Pandas Documentation](https://pandas.pydata.org/docs/)
* [Matplotlib Documentation](https://matplotlib.org/stable/contents.html)
* [Seaborn Documentation](https://seaborn.pydata.org/)
* [PyQt5 Documentation](https://doc.qt.io/qt-5/)

**11.Wireframes**

****

****

****

****

