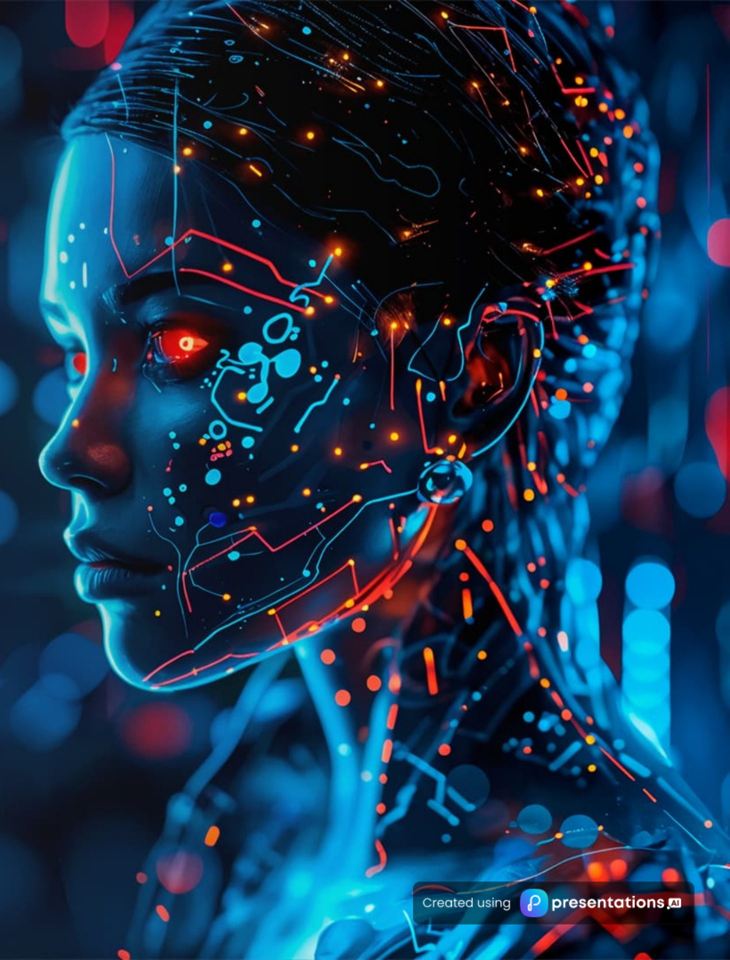
# AI-Powered Heart Risk Prediction with Blockchain and Generative AI for Personalized Health Plans

Aryan Sawant - 21101A0008
Shivraj Pawar - 21101A0017
Vishal Devkate - 21101A0019
Ashish Verma - 21101A0011



# Introduction to the Project

- **Project Overview**

  An end-to-end system for heart disease risk prediction using ML, Blockchain, and Generative AI.

- **Heart Disease Risk Prediction**

  Utilizes an optimized Random Forest model for high-accuracy predictions (99%).

- **Blockchain Integration**

  Employs Polygon blockchain for secure, transparent storage of prediction records.

- **Data Visualization**

  Features interactive dashboards with Streamlit and Plotly for historical trend analysis.

- **Generative AI Health Plans**

  Generates personalized health plans based on prediction outcomes using Google Gemini 2.0 Flash.

# Overview of Technologies Utilized

| | Purpose |
|---|---|
| Python | AI Model Training & Inference, API Dev, Data Processing, Streamlit App |
| Flask | API Development (for AI model endpoint) |
| Scikit-learn | Machine Learning (RandomForestClassifier, StandardScaler) |
| Pandas & Numpy | Data Manipulation |
| Pickle | Saving/Loading Trained ML Model |
| Google Gemini 2.0 Flash | Generative AI for personalized health plans |
| Optuna | Hyperparameter Optimization (used during model training) |
| Solidity | Smart Contract Development (`HeartRiskPrediction.sol`) |
| Hardhat | Blockchain Development Framework & Deployment |
| Ethers.js | Used within Hardhat scripts for deployment |
| Alchemy | Node provider for interacting with Polygon Amoy Testnet |
| Polygon PoS (Amoy Testnet) | Blockchain for storing full AI prediction records |

# AI Model for Heart Risk Prediction

### Objective of the AI Model

To predict heart disease presence using medical parameters.

### Dataset Selection

Used heart disease datasets like heart.csv with key health indicators.

### Data Preprocessing

Handled missing values and standardized features with StandardScaler.

### Model Training & Tuning

Multiple ML models trained, optimized Random Forest selected for high accuracy.

### Model Saving for Deployment

Saved best model and StandardScaler for consistent API use.

### Prediction Output

API returns prediction and probability of heart disease detection.

# Deploying the AI Model as an API

### Objective of API Deployment

Provide scalable, real-time access to heart disease prediction.

### API Development Process

Utilize Flask to create a web application for model deployment.

### Defining the Endpoint

Establish a /predict endpoint for accepting JSON data via POST requests.

### Loading Models

Integrate StandardScaler and RandomForestClassifier from saved .pkl files.

### Input Data Format

Accept 13 features in JSON format for predictions.

### Processing Input Data

Scale input features using StandardScaler before prediction.

### Making Predictions

Utilize the model to predict class and probability of heart disease.

### Output Formatting

Structure the prediction results into a JSON response format.

### API Deployment for Testing

Run Flask locally at http://127.0.0.1:5000 for development.

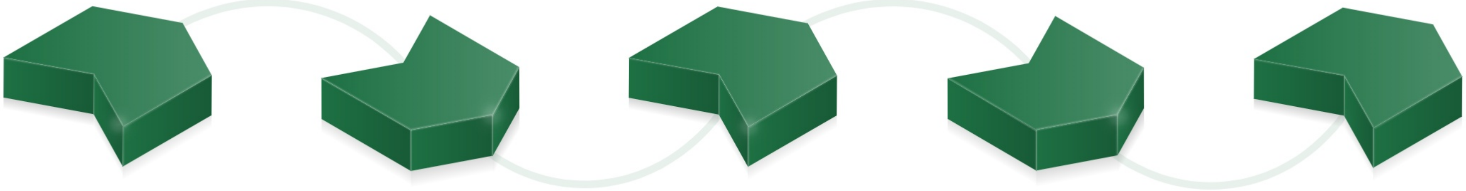
### Example JSON Request

Provide an example JSON request to the API for clarity.

### Example JSON Response

Show a sample output response from the API for heart disease detection.

# Blockchain Integration for Secure Storage



### Why Blockchain?

Ensures data integrity and immutability, providing tamper-proof records linked to user addresses for transparency.

### Why Polygon PoS?

Faster and cheaper transactions make it ideal for frequent data logging on the Amoy Testnet without real costs.

### Hardhat Development Environment

Used for compiling, testing, and deploying smart contracts in the blockchain setup process.

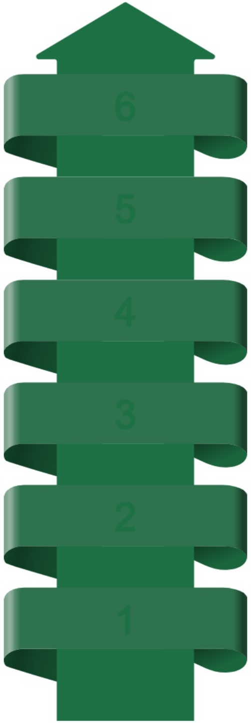
### Alchemy RPC Node

Facilitates connection to the Polygon Amoy network for deploying scripts and application connectivity.

### Smart Contract Implementation

Stores predictions using a structured contract for secure and traceable data storage on the blockchain.

# Data Visualization & Trend Analysis



### Interactive Dashboard

A user-friendly interface built with Streamlit for visualizing heart health data.

### Data Retrieval Process

Uses web3.py to fetch historical prediction data from the blockchain smart contract.

### Historical Data Table

Displays a comprehensive table of all past heart health predictions in a formatted view.

### Risk Trend Visualization

Plotly line chart illustrating prediction outcomes over time, showing trends in heart health risk.

### Feature Distribution Charts

Various Plotly charts displaying distributions of significant health features like Age and Cholesterol.

### Enhanced UI Styling

Custom CSS applied for improved aesthetics and user experience throughout the dashboard.

# Generative AI for Personalized Health Plans

Integrating AI for Tailored Health Recommendations

- **Objective of Generative AI**

  Generate tailored health recommendations based on heart disease prediction outcomes.

- **Why Choose Gemini 2.0 Flash?**

  Balances speed and contextual understanding for responsive health advice.

- **AI Prediction Process**

  Utilizes Flask API for heart disease risk prediction and returns textual outcomes.

- **Streamlit App Functionality**

  Converts text prediction into numeric values for further analysis and action.

- **Action Based on Prediction**

  Generates prompts based on prediction outcomes for tailored health advice.

- **Response for Heart Disease Detected**

  Creates urgent, detailed action plans for patients with detected heart disease.

- **Response for No Heart Disease**

  Generates preventive health plans focusing on maintenance and lifestyle choices.

- **Prompt Example Logic**

  Demonstrates structured prompts for both positive and negative predictions.

- **Generated Health Plan Structure**

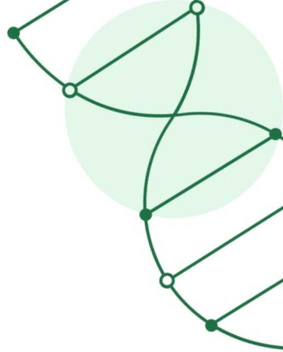
  Advice includes medical actions, lifestyle changes, and follow-up recommendations.

- **User Engagement in Streamlit**

  Displays generated advice in a user-friendly info box format.

# End-to-End System Workflow Overview

Comprehensive steps of health data processing



### User Inputs Health Data

Users enter 13 medical parameters through the Streamlit interface, initiating the process.

### API Call for Prediction

Streamlit sends the collected data to the Flask API for prediction.

### AI Predicts Heart Risk

Flask API preprocesses the input and predicts heart risk using a Random Forest model.

### Display Prediction & Advice

The prediction outcome and personalized health plan are displayed to the user via Streamlit.

### Blockchain Stores Record

Users can store their prediction data on the blockchain by clicking a button.

### View Historical Trends

Users can view past predictions and visualizations of their health data trends.

# Conclusion

- **High Accuracy in Risk Prediction**

  Utilizes an optimized Random Forest model achieving 99% accuracy for reliable predictions.

- **Immutable Record Keeping**

  Employs Polygon blockchain for tamper-proof and timestamped prediction records.

- **Data Transparency for Users**

  Users can access and review their entire prediction history stored securely on the blockchain.

- **Actionable Health Insights**

  Personalized health plans are generated based on specific prediction outcomes.

- **Interactive Visualization Dashboard**

  Rich dashboard with Plotly charts allows analysis of historical trends and data distributions.

- **Enhanced User Experience**

  Features a well-styled Streamlit interface with custom CSS for better usability.