AI Solutions Project Document (Based on UN SDGs)

1. Project Title

AI-Powered Health Monitoring and Early Risk Detection (SDG 3)

2. SDG Focus

Goal: SDG 3 – Good Health & Well-Being

Problem: Many individuals lack access to continuous health monitoring tools that can detect early signs of cardiovascular and respiratory risks. Traditional check-ups are infrequent, leaving potential health abnormalities unnoticed.

Challenge Addressed: "How can we provide real-time, AI-driven health risk detection using easily available sensor data from wearable devices?"

3. AI Approach

Software Engineering Skills Applied

- **Automation:** Automated preprocessing, anomaly detection, and disease prediction.
- **Testing:** Validation of machine learning models using accuracy metrics, manual test batches, and API endpoint tests.
- **Scalability:** Modular architecture with separate ML models, an API (Flask), and frontend integration.
- **Deployment:** Backend deployed on Google Cloud Run; frontend hosted on Firebase.

Technical Solution

An AI-powered pipeline that: 1. Collects health metrics such as heart rate, SpO2, and activity levels. 2. Uses ML-based anomaly detection (*Isolation Forest, Random Forest*) to detect irregular patterns. 3. Predicts disease risk levels using a trained Random Forest model. 4.

Generates personalized lifestyle recommendations. 5. Displays results via an accessible React web application.

4. Tools & Frameworks

AI / ML

- Scikit-learn
- NumPy, Pandas

Software Engineering Tools

- Flask (API)
- Google Cloud Run (container deployment)
- Firebase Hosting (frontend)
- Docker
- Git & GitHub

Data Sources

- Public biosignal datasets
- Synthetic wearable data for training

5. Deliverables

Code

- Python scripts for preprocessing, model training, and evaluation.
- Flask backend API with `/predict_all` endpoint.
- React frontend for visualization.

Deployment

- Dockerized backend running on Google Cloud Run.
- React frontend deployed on Firebase.

6. Ethical & Sustainability Checks

Bias Mitigation

- Balanced dataset used for cardiovascular risk classification.
- Evaluated model for overfitting and class imbalance.

Environmental Impact

- Lightweight ML models chosen (Random Forest, Isolation Forest).
- Efficient preprocessing to reduce compute costs in the cloud.

Scalability

- Designed to operate in low-resource settings.
- Works even with minimal wearable data (HR + SpO2 only).

7. Project Outline

Phase: Ideation

- Researched SDG 3 challenges.
- Identified gaps in real-time monitoring.
- Proposed AI-driven anomaly and disease prediction.

Phase: Development

- Built preprocessing pipeline.
- Trained anomaly detection models.
- Trained disease risk Random Forest model.
- Implemented Flask API.

Phase: Testing

- Validated models on test batches.
- Performed bias analysis.
- Conducted local API tests before deployment.

Phase: Deployment

- Dockerized and deployed backend on Cloud Run.
- Connected React frontend.
- Deployed frontend on Firebase.

Phase: Monitoring

- Logs stored for each prediction.
- Future extensions include model drift detection.

8. Machine Learning Models (Detailed)

8.1 Anomaly Detection Models

Two models detect abnormal patterns: - **Random Forest Classifier** (binary: normal vs anomaly) - **Isolation Forest** (unsupervised outlier detection)

These models were trained using biosignal features: - Heart Rate (HR) - SpO2 - Rolling mean, std, variance - HR differences - SpO2 trends

8.2 Disease Risk Prediction Model

A Random Forest model that predicts: - **Low Risk** or **High Risk** cardiovascular status

It uses the same engineered features but is trained using labelled disease-risk datasets.

9. Model Training Code Snippets

9.1 Isolation Forest Training

```
import pandas as pd
from sklearn.ensemble import IsolationForest
from sklearn.preprocessing import StandardScaler
import joblib

# Load training data
df = pd.read_csv("training_data.csv")
features = ["HR", "SpO2", "HR_MEAN", "HR_STD", "HR_DIFF", "HR_VAR", "SPO2_DIFF",
"SPO2_TREND"]

# Scale data
scaler = StandardScaler()
scaled = scaler.fit_transform(df[features])
joblib.dump(scaler, "physionet_scaler.joblib")

# Train model
iso = IsolationForest(contamination=0.05)
iso.fit(scaled)

# Save
joblib.dump(iso, "physionet_isolation_forest.joblib")
```

9.2 Random Forest Anomaly Model

```
from sklearn.ensemble import RandomForestClassifier
import joblib

rf = RandomForestClassifier(n_estimators=200)
rf.fit(df[features], df["label"])  # label: 0 = normal, 1 = anomaly
joblib.dump(rf, "physionet_random_forest.joblib")
```

9.3 Disease Risk Model Training

```
from sklearn.ensemble import RandomForestClassifier
import joblib

# Train risk model
risk_model = RandomForestClassifier(n_estimators=300)
risk_model.fit(df[features], df["disease_risk"])  # 0=low, 1=high

joblib.dump(risk_model, "disease_rf_model.joblib")
```

10. Conclusion

This AI-powered health monitoring system aligns with SDG 3 by enabling early detection of health risks using machine learning. The system is: - Lightweight - Cloud-deployed - Real-time - Scalable

Future improvements: - Integrate wearable IoT streaming - Add deep learning for ECG data - Integrate real doctors into reporting loop