

24-25J-109

# PATIENT-CENTERED MOBILE APPLICATION FOR COMPREHENSIVE DIABETES SELF-MANAGEMENT AND OPTIMIZATION

Progress Review 2



# OUR MEMBERS



**Supervisor**  
Dr.Junius Anjana



**Co-Supervisor**  
Ms.Gaya Thamali Dassanayake



Gunawardhana D.H.M.G



Kajeevan J



De Silva L.K.N



Wimansa P.P.S.D.

# OUR EXTERNAL SUPERVISORS



**Dr. Ramesh Kumar Thevraj**  
Senior Registrar in Endocrinology  
National Hospital Sri Lanka



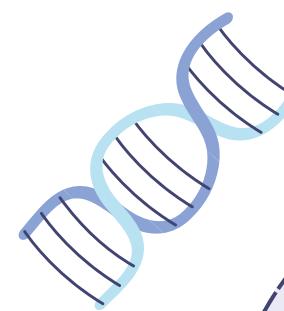
**DR.SOMESWARAPILLAI PATHANTHAN**  
GRADE I MEDICAL OFFICER, NATIONAL  
HOSPITAL, SRILANKA.

# INTRODUCTION

- Integrated mobile app for comprehensive management.
- Real-time data integration from devices.
- Feedback system to track progress and adjust recommendations.
- User-friendly interface to enhance patient engagement and education.



# RESEARCH PROBLEM

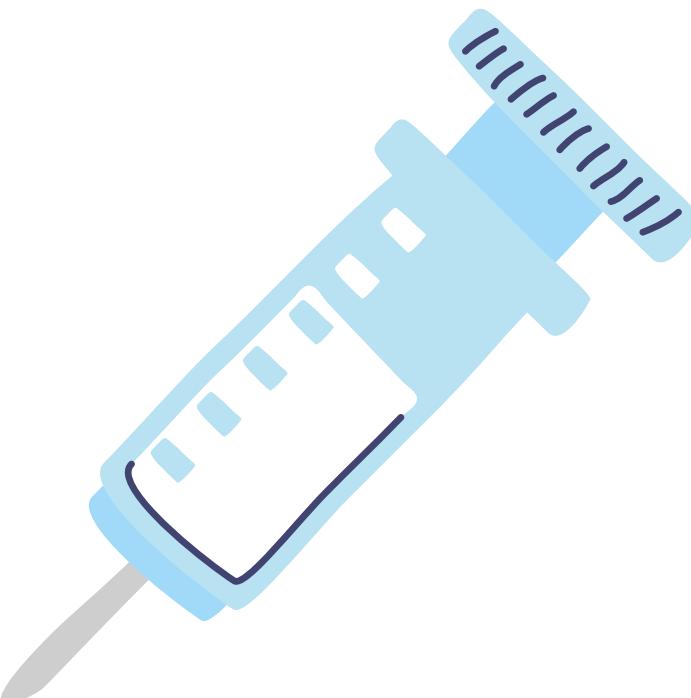


## Insulin Optimization

How can advanced machine learning algorithms be utilized to adjust insulin dosage effectively, minimizing the risks of hypo- and hyperglycemia?

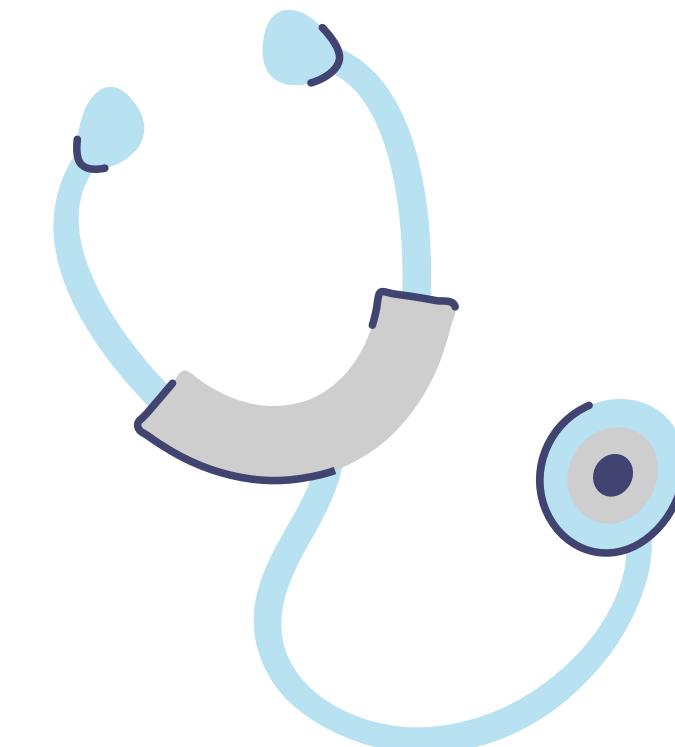
## Side Effect Management

How can a system effectively monitor and manage side effects like hypoglycemia, weight gain, and cardiovascular issues?

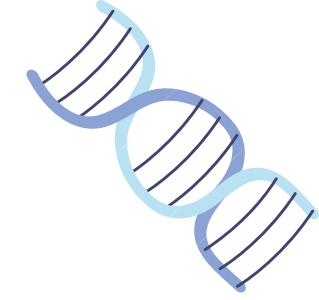
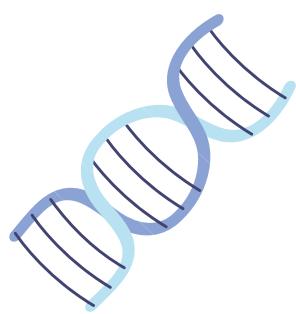


## Simplified Food Tracking

How can automated systems ensure accurate and user-friendly tracking of food intake for diabetes patients?



# RESEARCH GAP

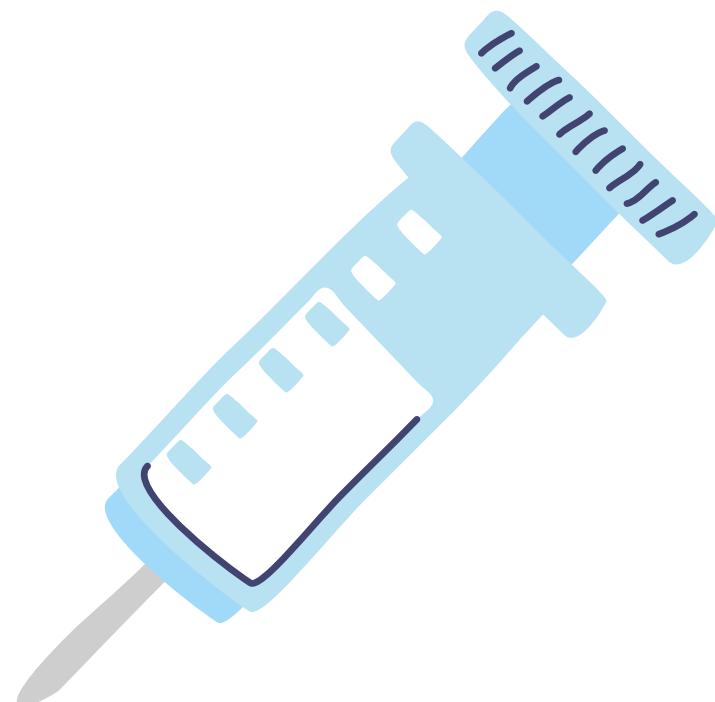


Limited Personalization

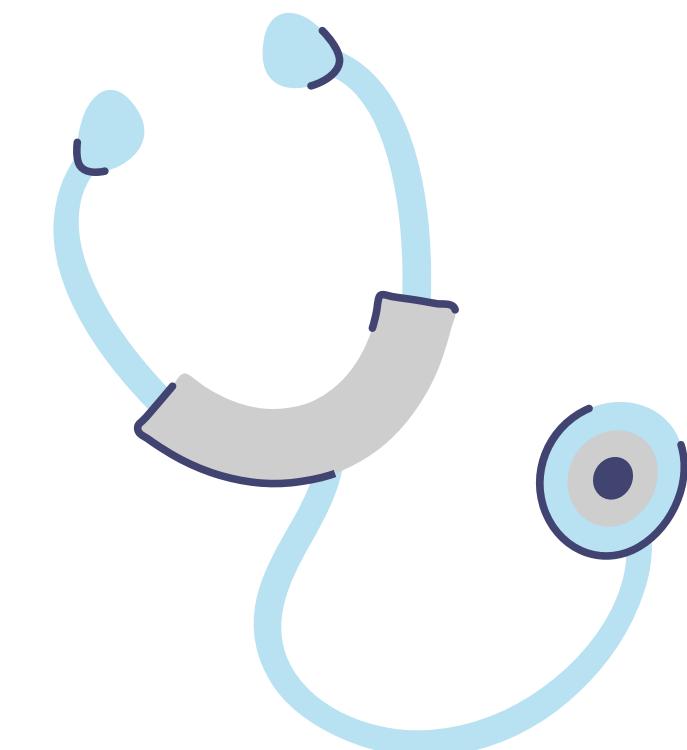
Data Overload Challenges

Side Effect Management

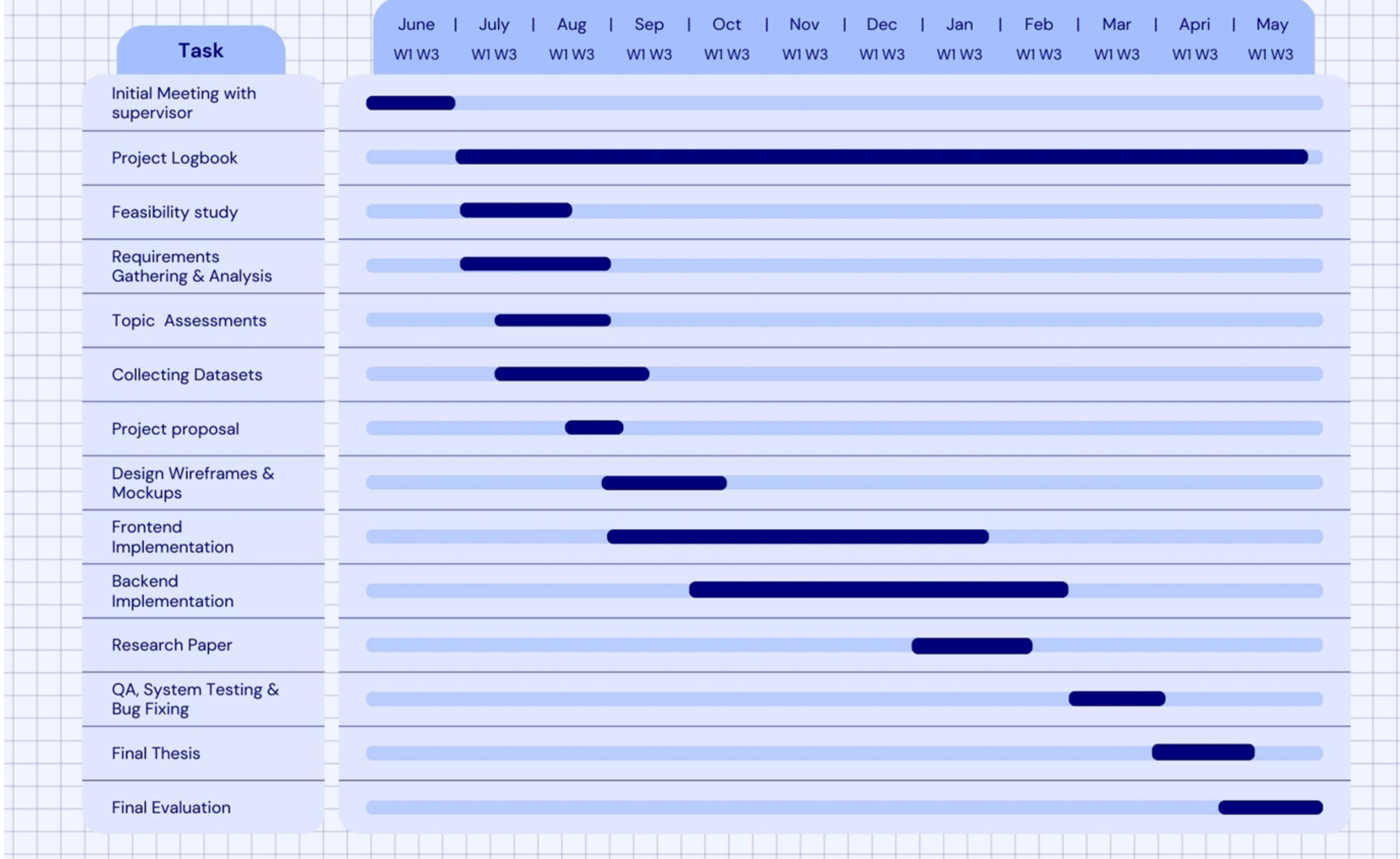
Inadequate Food Tracking Tools



Glycemic Variability Insights



# Project Timeline: Gantt Chart



# Commercialization Plan

## 1. Target Market & Value Proposition

- Targeting Type 1 & Type 2 diabetes patients, healthcare providers, and clinics.
- Real-time blood glucose prediction, insulin optimization, and personalized care using AI.

## 2. Product Development

- Initial mobile app with cloud-based AI models.
- Future integration with wearables and medical devices.

## 3. Marketing Strategy

- Educational campaigns, social media influencers, and healthcare conferences.
- Targeted advertising to patients and providers.

## 4. Scaling & Expansion

- Strategic partnerships with tech and pharmaceutical companies.

## 5. Customer Support & Engagement

- 24/7 support and community building.

## 6. Financial Projections

- Initial Funding: Seek investment through venture capital, crowdfunding, or partnerships with healthcare organizations to fund product development and marketing.



**IT21289798 | GUNAWARDHANA D.H.M.G.**

**SPECIALIZATION - DATA SCIENCE**

**PERSONALIZED INSULIN AND MEDICATION**

**PREDICTION WITH PATIENT ENGAGEMENT**

# INTRODUCTION

This innovative system leverages AI, time-series analysis, and real-time data processing to predict blood glucose levels and optimize insulin dosing, revolutionizing diabetes care. Built with React Native, Python, and Cloud Computing.



# RESEARCH PROBLEM

Managing diabetes requires precise blood glucose predictions and insulin dosing. Current methods lack personalization and real-time accuracy, leading to suboptimal care.



# RESEARCH SOLUTION

Develop a machine learning-powered system for accurate glucose prediction and insulin optimization using time-series analysis and real-time data, enabling better health outcomes and decision-making.

# RESEARCH GAP

## Patient-Centered Design

Limited tools that provide personalized insights tailored to individual patient profiles and conditions.

## Decision Support

Insufficient integration of actionable recommendations for healthcare providers and patients to improve treatment decisions.

## Data Integration

Lack of systems that effectively combine diverse data sources (e.g., glucose levels, insulin doses, carbohydrate intake, and lifestyle factors).

## Technological Integration

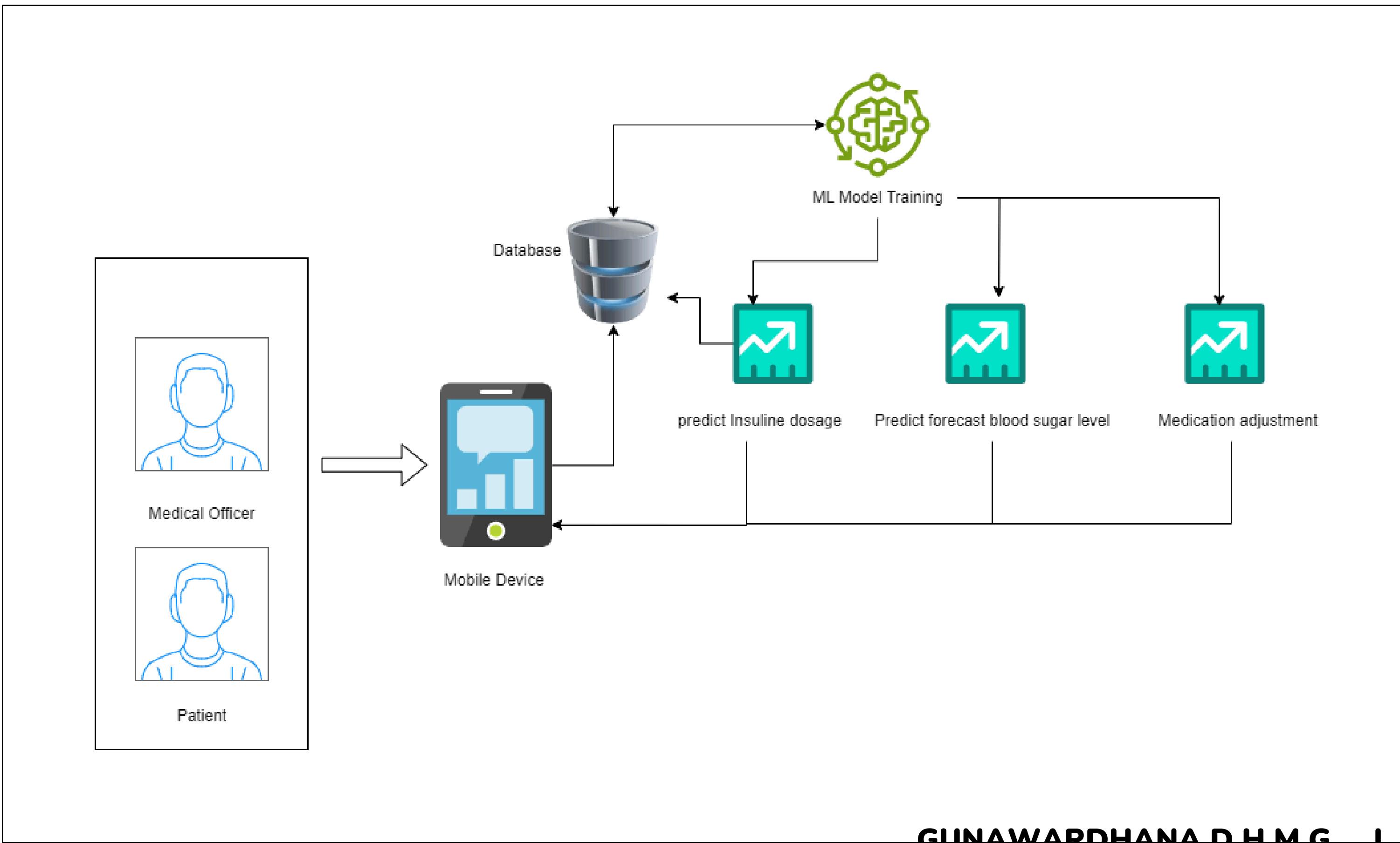
Underutilization of advanced AI, time-series analysis, and cloud computing for seamless and scalable diabetes care solutions.

# DATA SET

- **Size:** 269,768 rows and 104 columns
- **Key Features:**
- **Timestamps:** Track blood glucose and insulin changes over time.
- **Blood Glucose Levels:** Real-time glucose readings.
- **Insulin Doses:** Administered insulin amounts.
- **Carbohydrate Intake:** Records of carb consumption affecting glucose levels.
- **Lag Variables:** Previous values (e.g., glucose\_lag\_1) for predictive analysis.
- **Patient Info:** Details like weight and insulin type for personalized care.

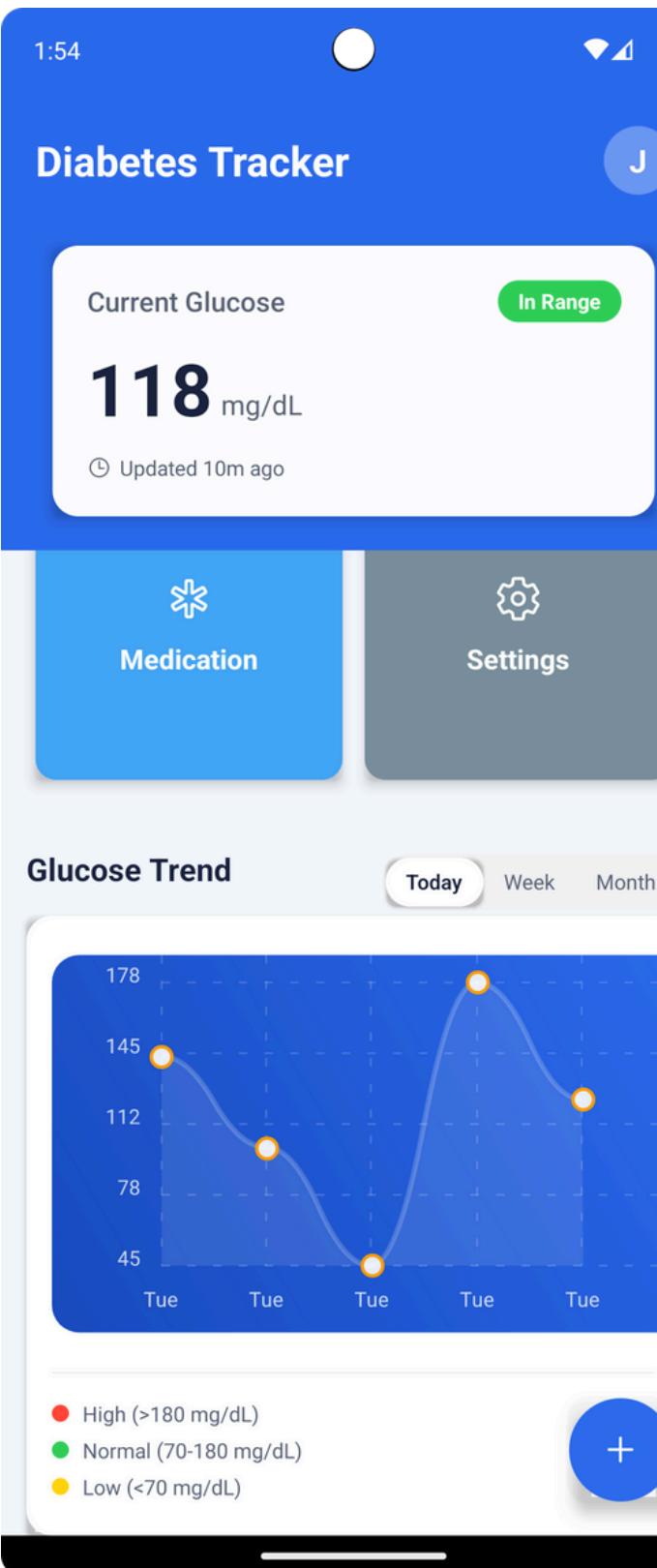
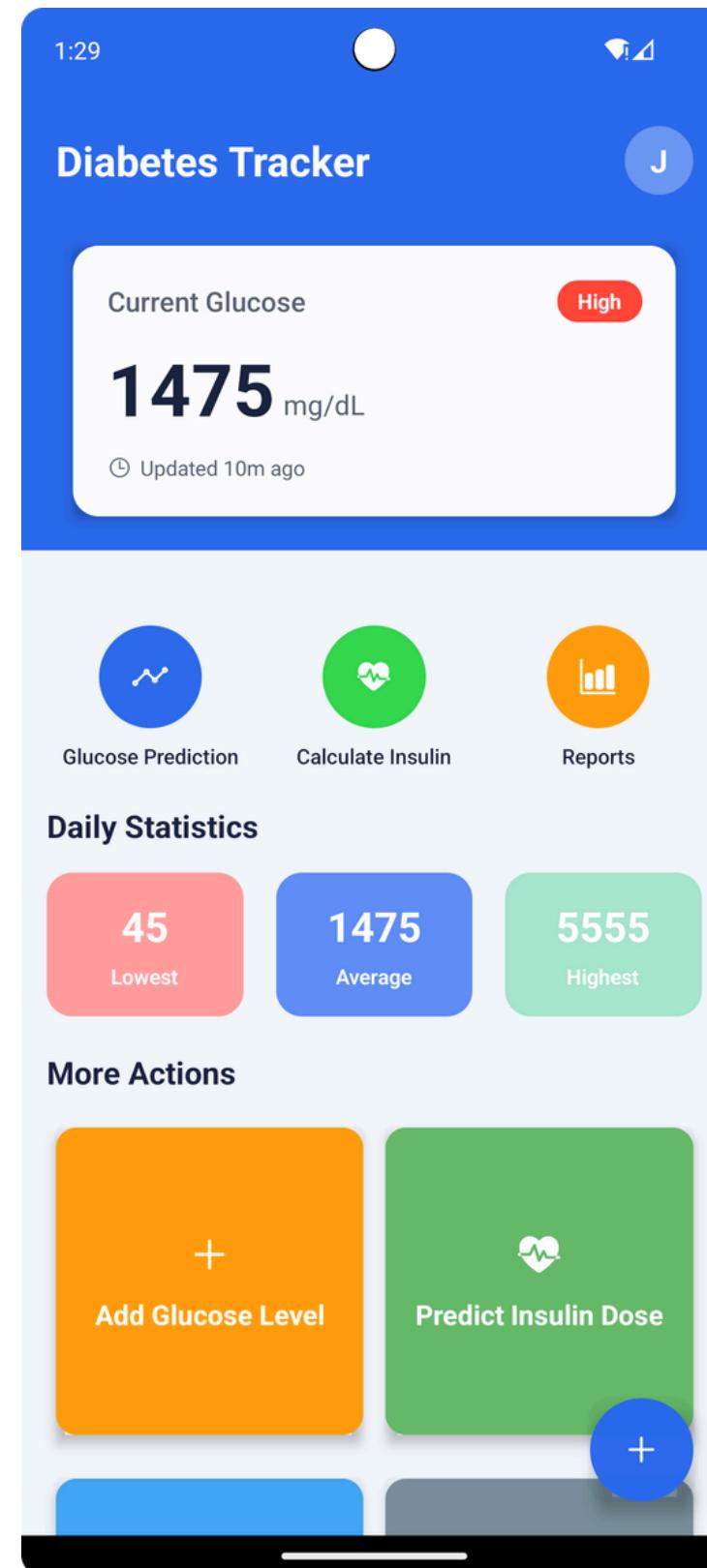


# SYSTEM DIAGRAM



# PROOF OF COMPLETION

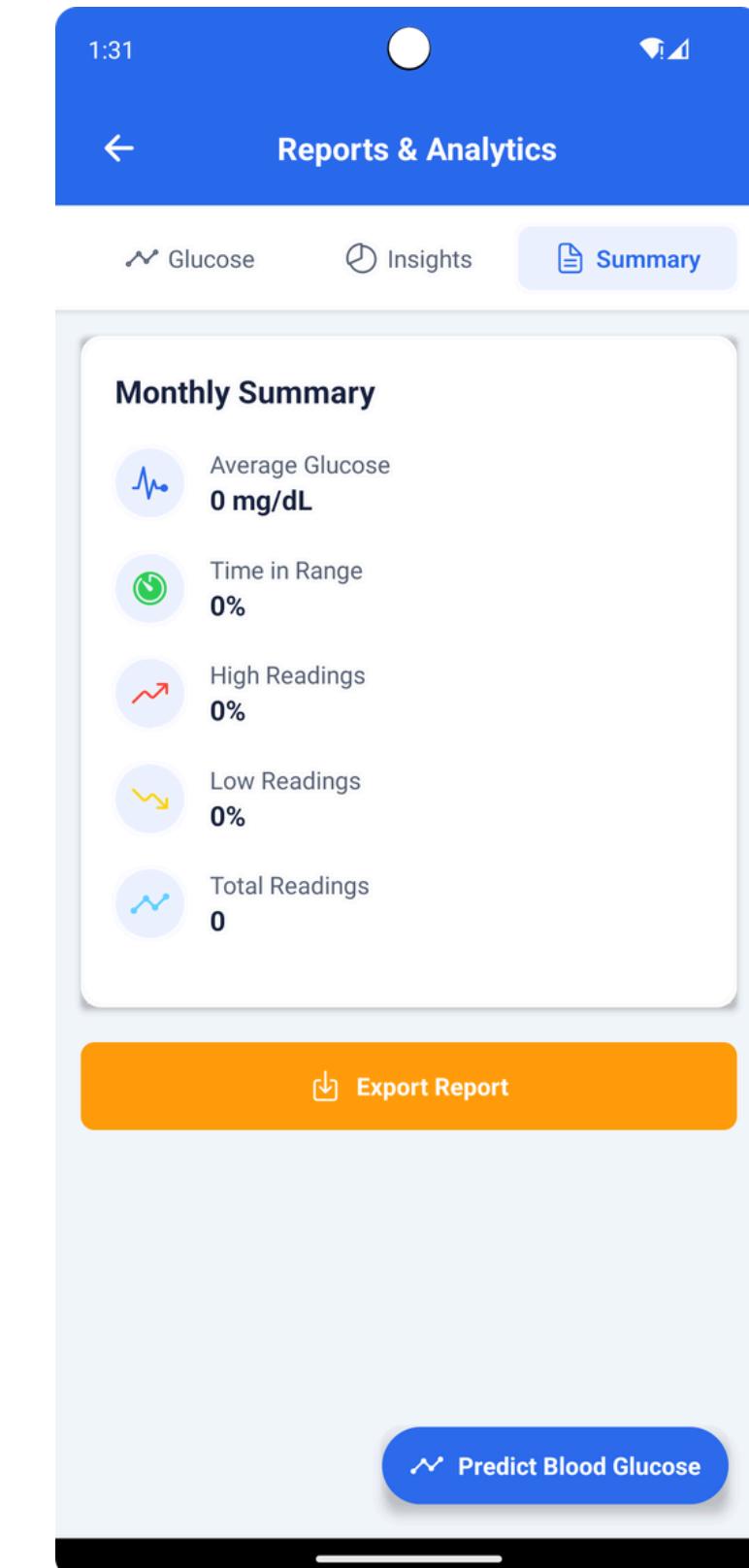
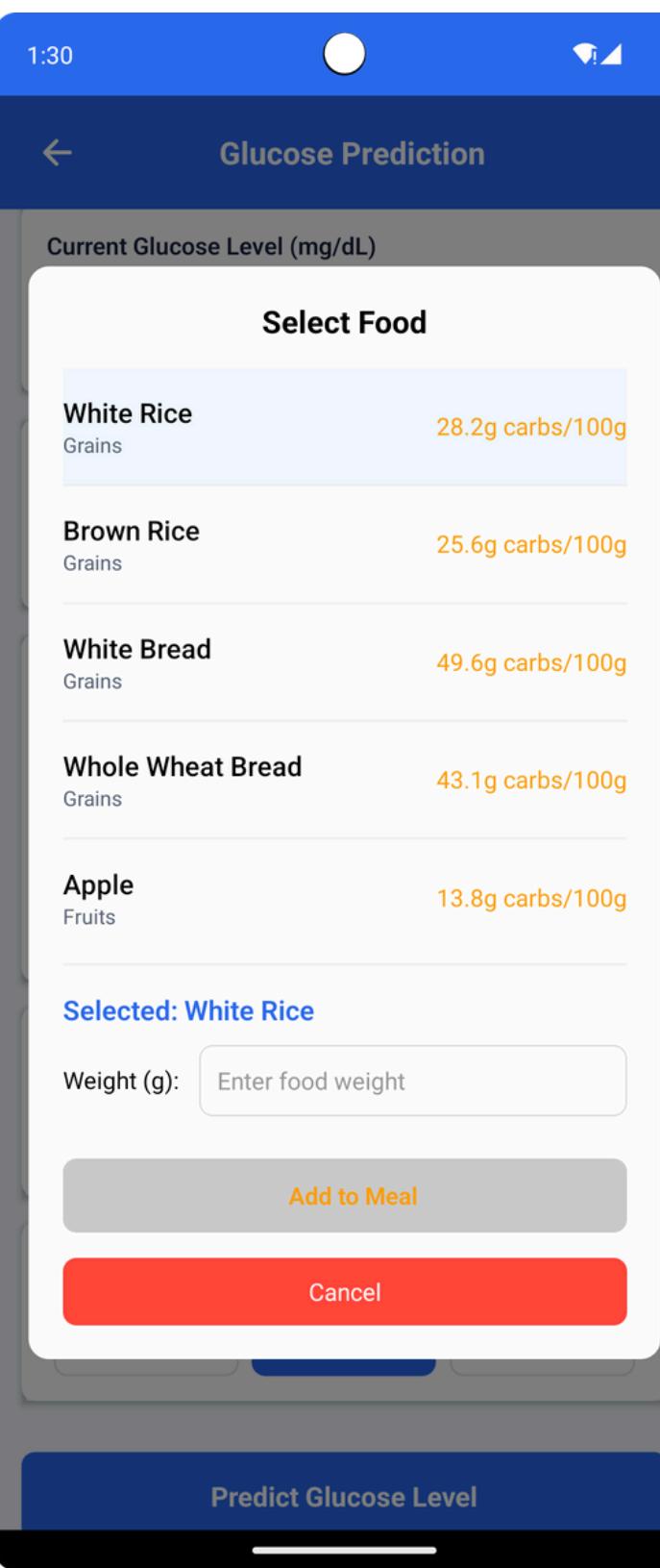
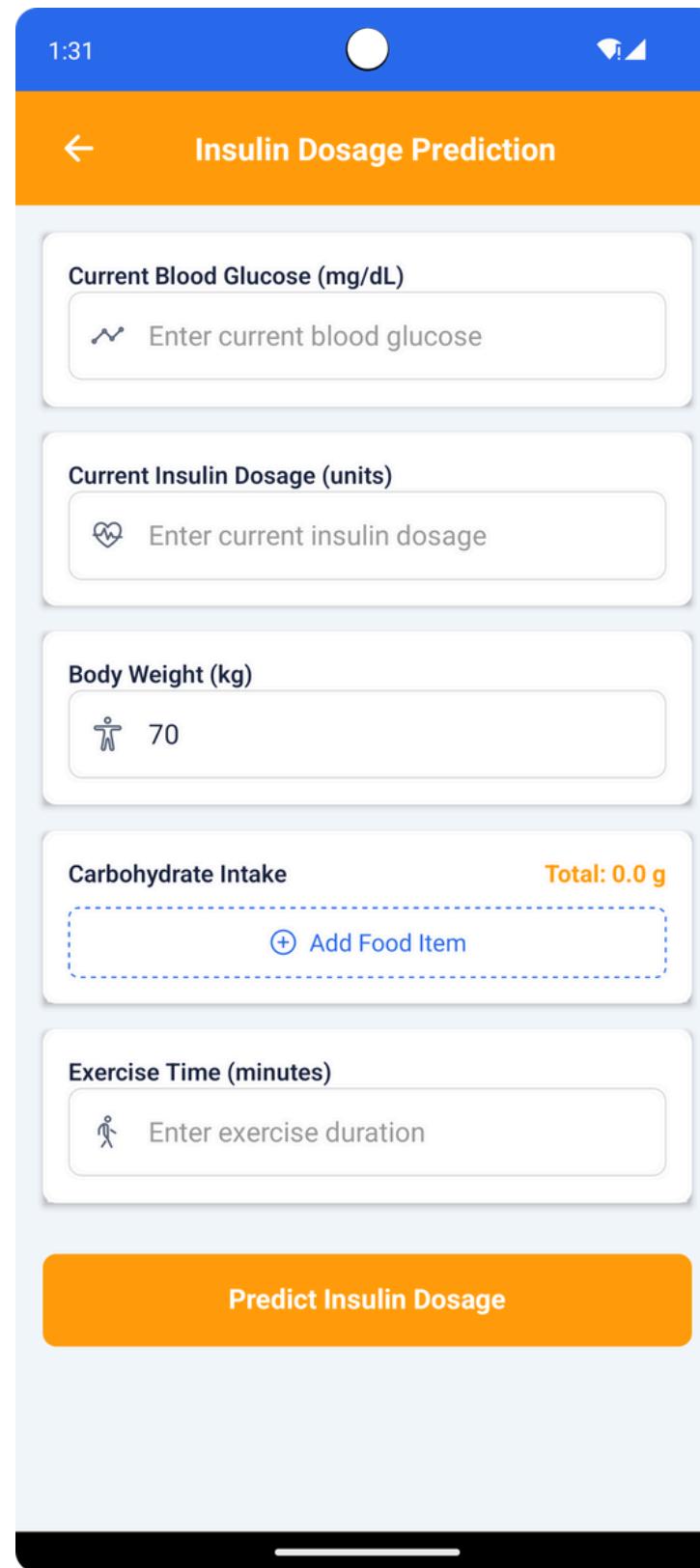
## MOBILE APPLICATION



The "Glucose Prediction" screen has a back arrow at the top left. It contains several input fields: "Current Glucose Level (mg/dL)" with a placeholder "Enter current glucose level", "Insulin Dose (units)" with a placeholder "Enter insulin dose", "Food Intake" with a "Add Food Item" button and a note "No food items added", "Exercise Duration (minutes)" with a placeholder "Enter exercise duration", and "Exercise Intensity" with buttons for "Low", "Medium" (highlighted in blue), and "High". At the bottom is a large blue "Predict Glucose Level" button.

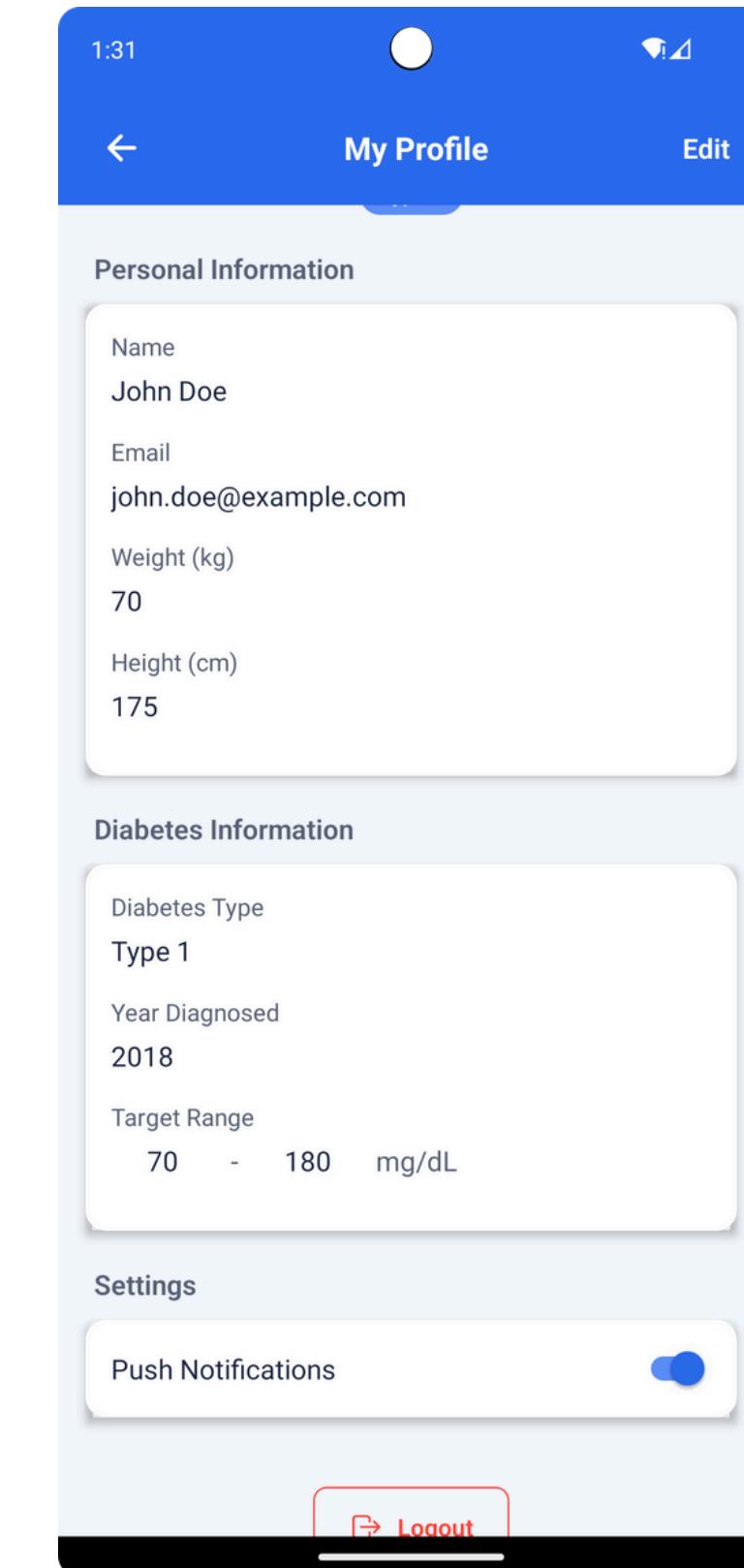
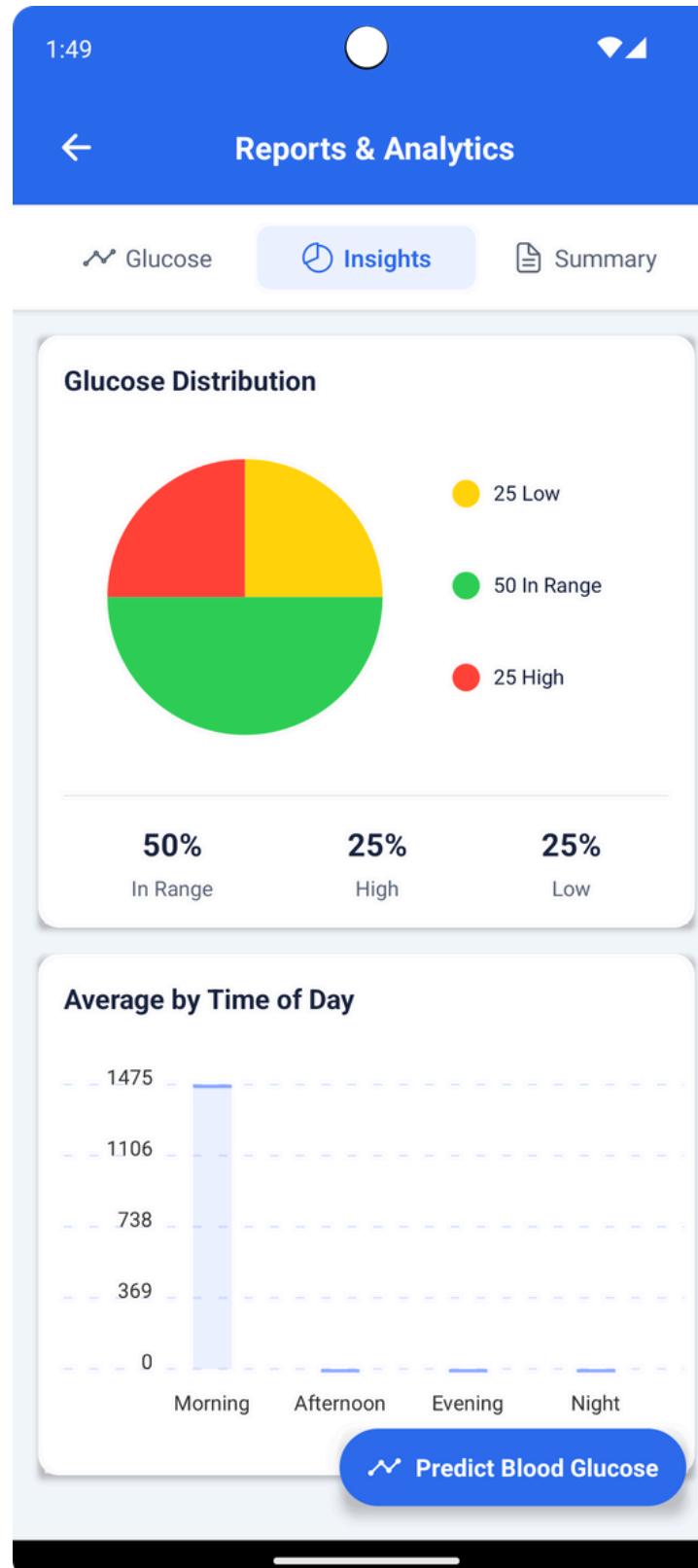
# PROOF OF COMPLETION

## MOBILE APPLICATION



# PROOF OF COMPLETION

## MOBILE APPLICATION



# **FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS**

## **FUNCTIONAL REQUIREMENTS**

- USER MANAGEMENT
- GLUCOSE TRACKING
- INSULIN MANAGEMENT
- CARBOHYDRATE TRACKING
- ANALYSIS & REPORTING

## **NON-FUNCTIONAL REQUIREMENTS**

- PERFORMANCE & SCALABILITY
- SECURITY & PRIVACY
- AVAILABILITY & RELIABILITY
- USABILITY & ACCESSIBILITY

# METHODOLOGY

## Frontend

- React Native
- Responsive Design
- Intuitive User Interface

## Backend

- Node.js with Express
- RESTful API Design

## Database

- MongoDB
- Scalable Data Storage

## Feature Engineering

- Time-series analysis
- Contextual feature extraction

## Machine Learning Algorithms

- Random Forest
- Gradient Boosting



# FUTURE WORKS

**Fix Some bugs In frontend**

**Integrate Mobile App**

# REFERENCES

- [1] C. R. Marling and R. C. Bunescu, "The OhioT1DM Dataset for Blood Glucose Level Prediction," Proceedings of the 3rd International Workshop on Knowledge Discovery in Healthcare Data, Stockholm, Sweden, 2018.
- [2] D. R. Zisser, J. E. Jovanovic, and M. D. Laffel, "Clinical Evaluation of a Continuous Glucose Monitoring System for Type 1 Diabetes," Journal of Diabetes Science and Technology, vol. 3, no. 4, pp. 1-8, 2009.
- [3] K. C. Santarcangelo, P. J. Johnson, and C. D. Garcia, "A Machine Learning Approach for Predicting Blood Glucose in Diabetes Management," Journal of Healthcare Engineering, vol. 2019, Article ID 2909471, 2019.
- [4] R. D. P. Haug, M. F. Gubbi, and S. P. George, "Optimizing Insulin Dosing: A Data-Driven Approach," IEEE Transactions on Biomedical Engineering, vol. 68, no. 3, pp. 1-10, Mar. 2021.
- [5] S. R. Basak, S. Ghosh, and A. K. Bhattacharya, "Predictive Models for Diabetes Management: A Review," Journal of Medical Systems, vol. 42, no. 6, pp. 23-45, 2018.
- [6] A. G. Ganaie and M. A. Ziarati, "Real-Time Blood Glucose Monitoring and Prediction Using Machine Learning Algorithms," IEEE Access, vol. 8, pp. 150345-150353, 2020.
- [7] P. M. Santos and R. M. Lima, "Cloud-Based Healthcare System for Diabetes Management with AI Integration," Journal of Health Informatics, vol. 28, no. 2, pp. 58-70, 2022.
- [8] L. M. Greenfield, "Integrating Time-Series Data for Predictive Analytics in Healthcare," International Journal of Data Science and Analytics, vol. 5, no. 3, pp. 211-219, 2021.



**IT21286032 | KAJEEVAN J**

**SPECIALIZATION - DATA SCIENCE**

**REAL-TIME GLYCEMIC EVENT PREDICTION AND  
MANAGEMENT TOOL.**

# Introduction

Diabetes is a growing health challenge in Sri Lanka, with prevalence rising to 8.7% and affecting over 1.2 million individuals. Effective management requires continuous monitoring of blood glucose levels to prevent dangerous glycemic events.

- **Problem:** Existing healthcare solutions lack real-time predictive capabilities and personalized management options, leading to preventable complications.
- **Our solution:** Transforming diabetes care through AI-powered prediction of glycemic events and personalized management tools.
- **Implementation progress:** Our initial prototype has evolved into a functional mobile application with 90% of planned features completed, demonstrating the practical application of machine learning for healthcare.



# Research Question

**How can CGM data and machine learning predict and manage glycemic events in real-time for diabetes patients in Sri Lanka?**

**What technologies best support real-time glucose monitoring, predictive analytics, and personalized recommendations?**

# Objectives



## Specific Objectives

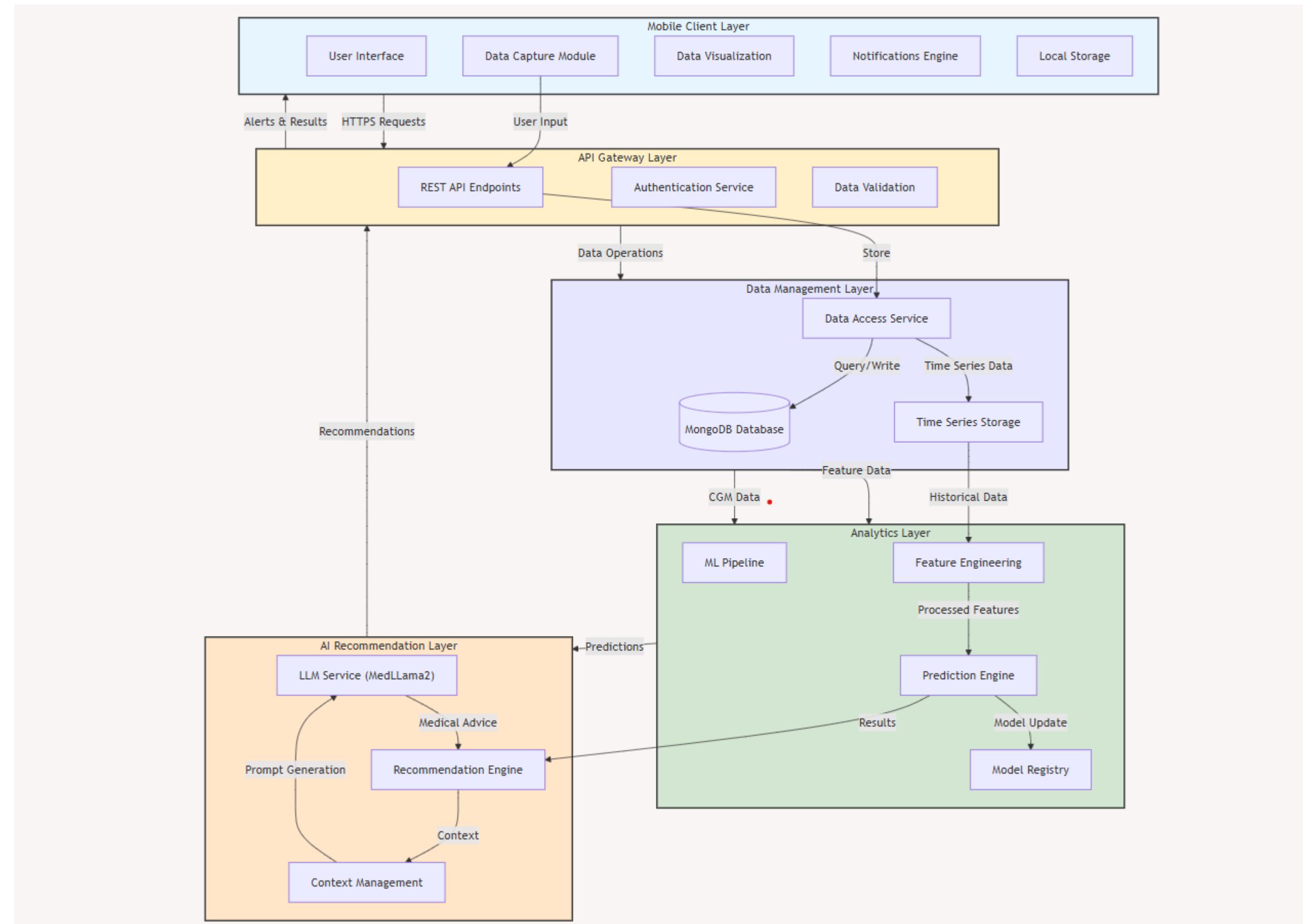
- Develop Predictive Algorithms & Management System for Glycemic Events



## Sub Objectives

- Integrate CGM data with insulin, activity, and diet.
- Develop predictive models for accurate event forecasting.
- Create a recommendation system for real-time management.

# Component Diagram



# What I Have Finished:

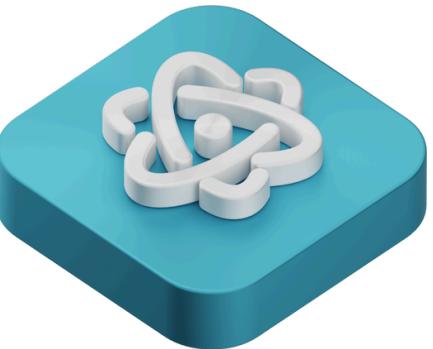
- Time Series Machine Learning Model
- UI Implementation
- Backend API Services
- MongoDB Integration
- Prediction Engine
- Medlama based Recommendation System

# What I Need to Finish:

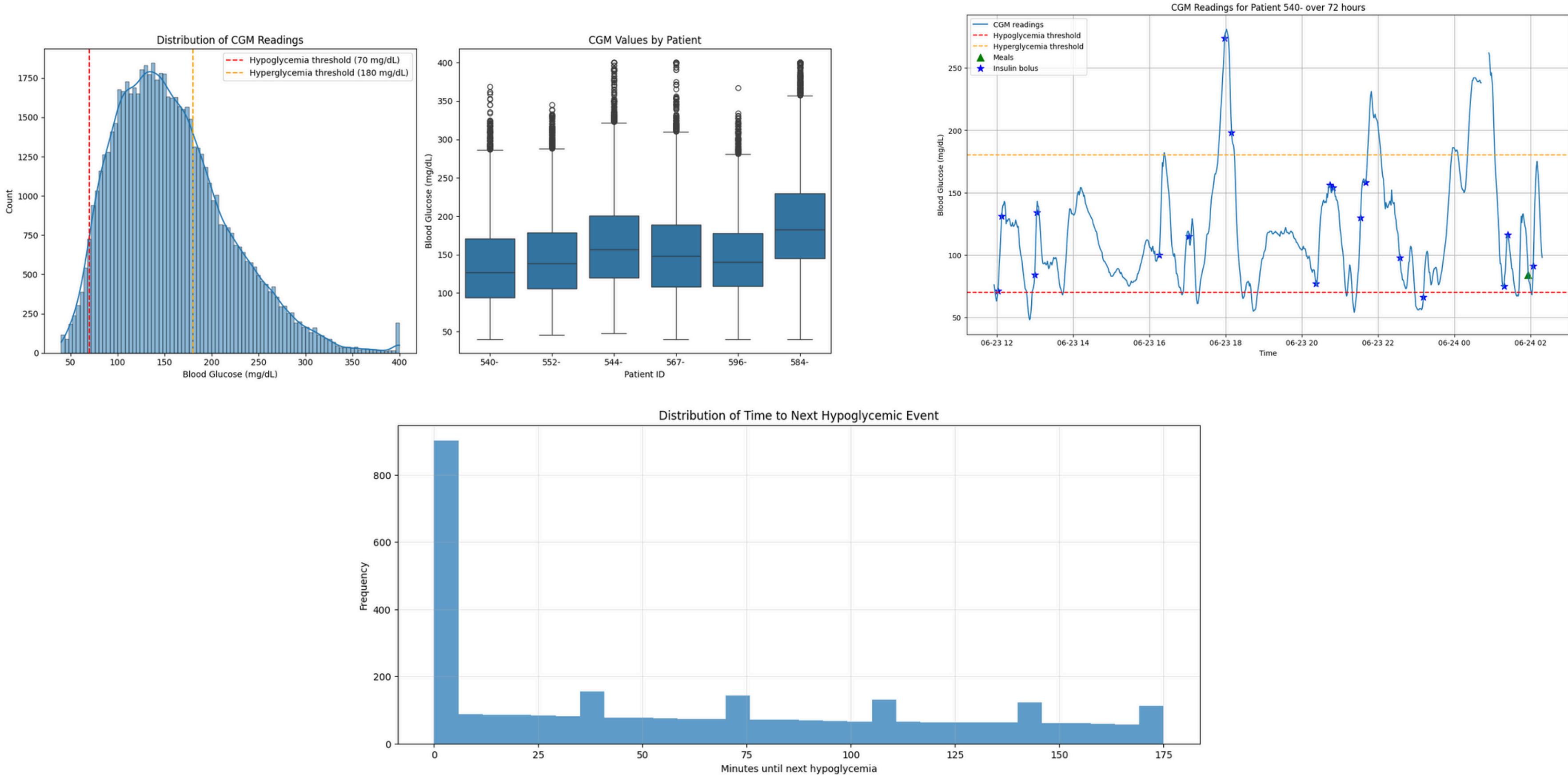
- Real-Time CGM Device Integration
- Kubernetes Deployment
- HIPAA-Compliant Security Implementation

# Technologies

- **Backend:** Python with Flask APIs, MongoDB for data storage
- **Frontend:** React Native with Expo Router
- **Machine Learning:** Keras with TensorFlow (GRU for time series prediction)
- **AI Recommendation:** MedLLama2 on Ollama for medical advice
- **Data Integration:** CGM devices, fitness bands (via APIs)
- **Deployment:** Kubernetes for scalability and reliability
- **Security:** OAuth2, data encryption, HIPAA compliance measures
- **Analytics:** Time-series data processing pipeline



# Proof Of Completion- EDA



# Proof Of Completion

## Model Training

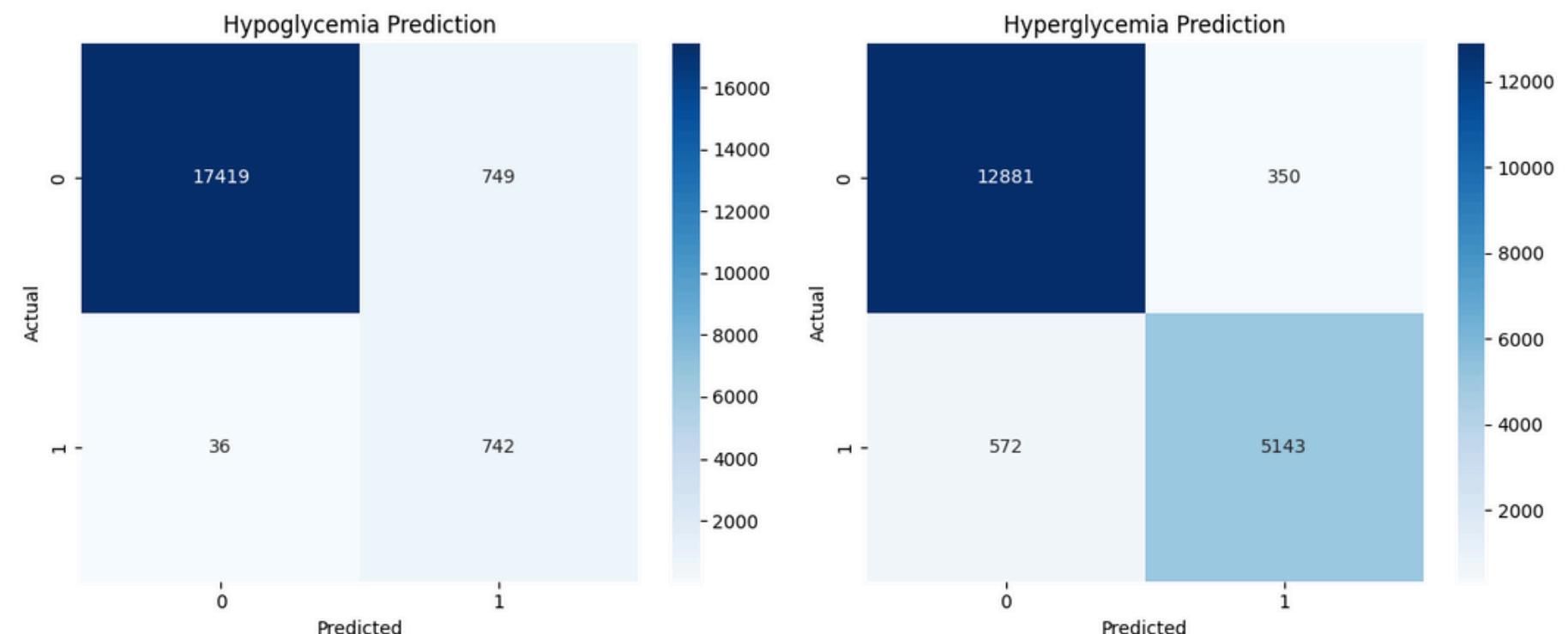
```
3685/3685 82s 16ms/step - loss: 0.0506 - mae: 0.0663 - val_loss: 0.2106 - val_mae: 0.1233
Epoch 10/50
3685/3685 80s 15ms/step - loss: 0.0487 - mae: 0.0649 - val_loss: 0.2096 - val_mae: 0.1224
Epoch 11/50
3685/3685 59s 16ms/step - loss: 0.0464 - mae: 0.0635 - val_loss: 0.2229 - val_mae: 0.1281
Epoch 12/50
3685/3685 80s 15ms/step - loss: 0.0445 - mae: 0.0624 - val_loss: 0.2303 - val_mae: 0.1300
Epoch 13/50
3685/3685 83s 16ms/step - loss: 0.0427 - mae: 0.0614 - val_loss: 0.2303 - val_mae: 0.1267
Evaluating model...
593/593 3s 5ms/step
```

Hypoglycemia Prediction Performance:

	precision	recall	f1-score	support
0	1.00	0.96	0.98	18168
1	0.50	0.95	0.65	778
accuracy			0.96	18946
macro avg	0.75	0.96	0.82	18946
weighted avg	0.98	0.96	0.96	18946

Hyperglycemia Prediction Performance:

	precision	recall	f1-score	support
0	0.96	0.97	0.97	13231
1	0.94	0.90	0.92	5715
accuracy			0.95	18946
macro avg	0.95	0.94	0.94	18946
weighted avg	0.95	0.95	0.95	18946



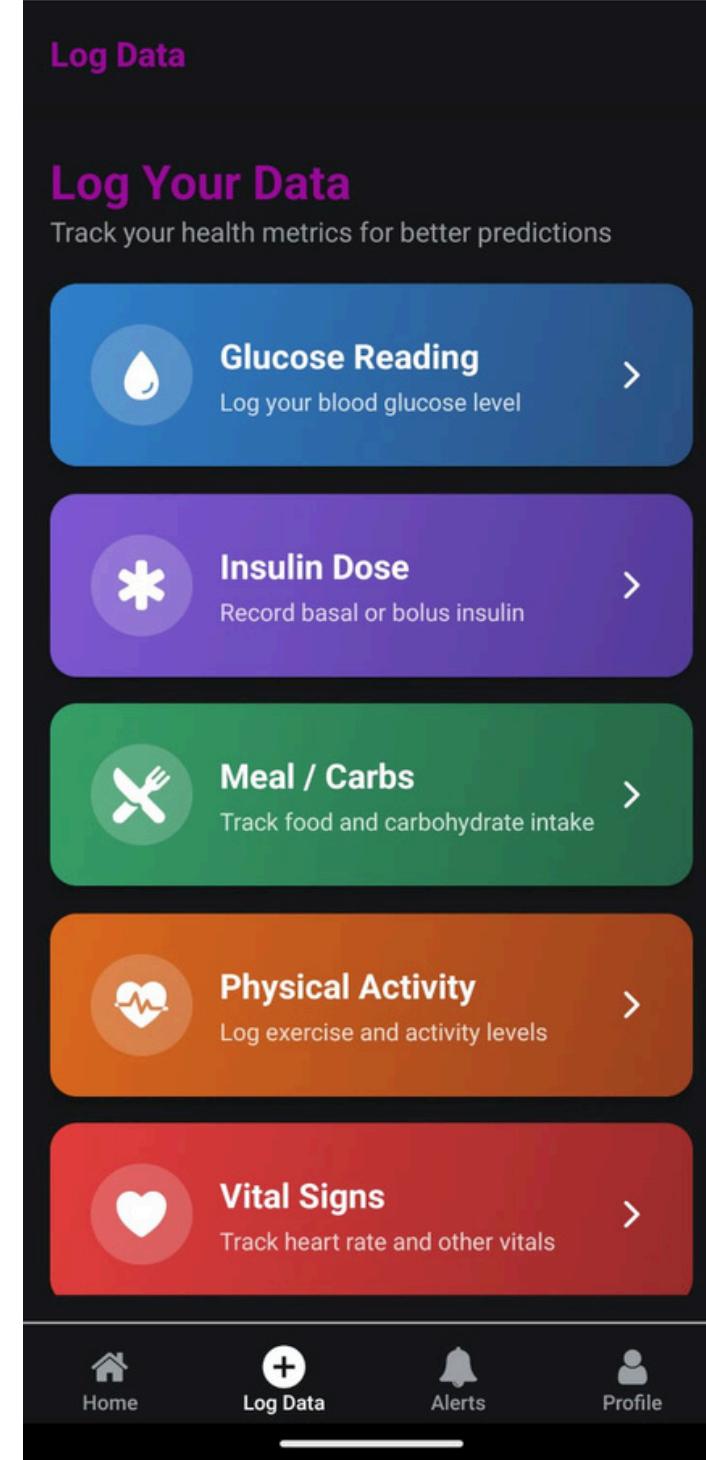
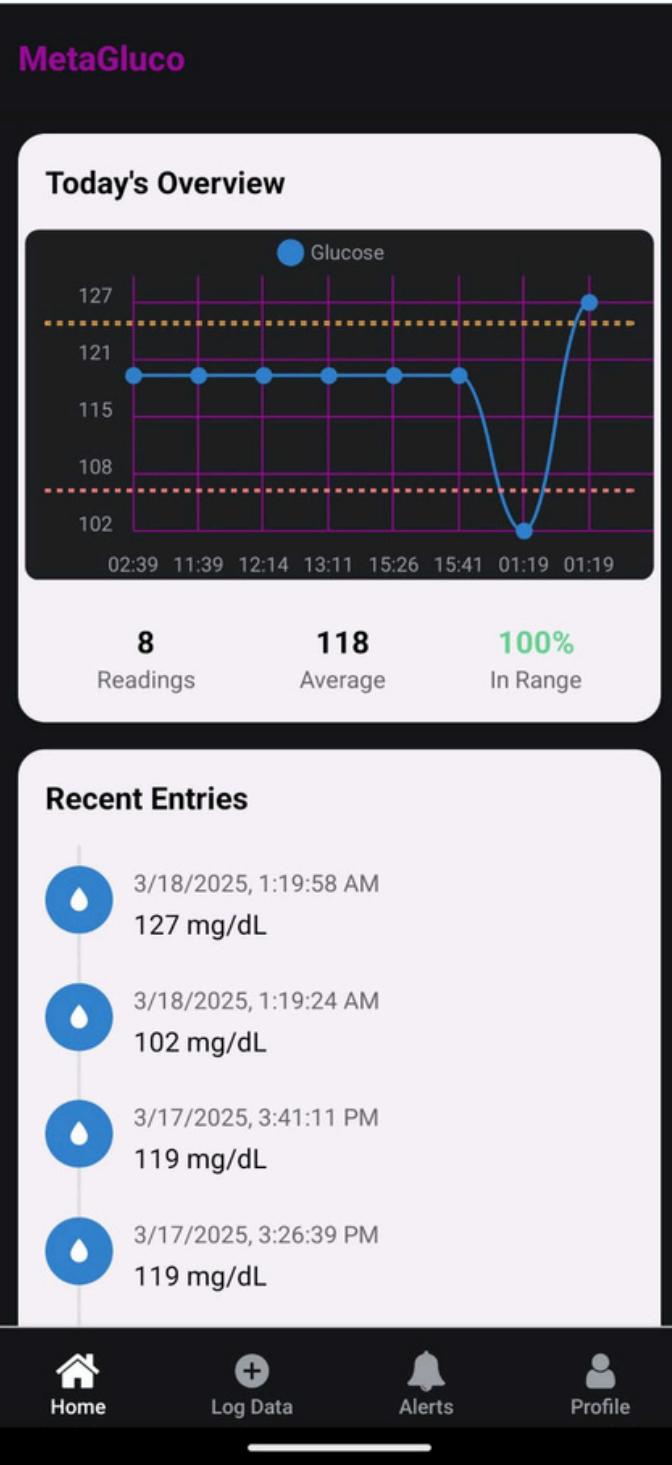
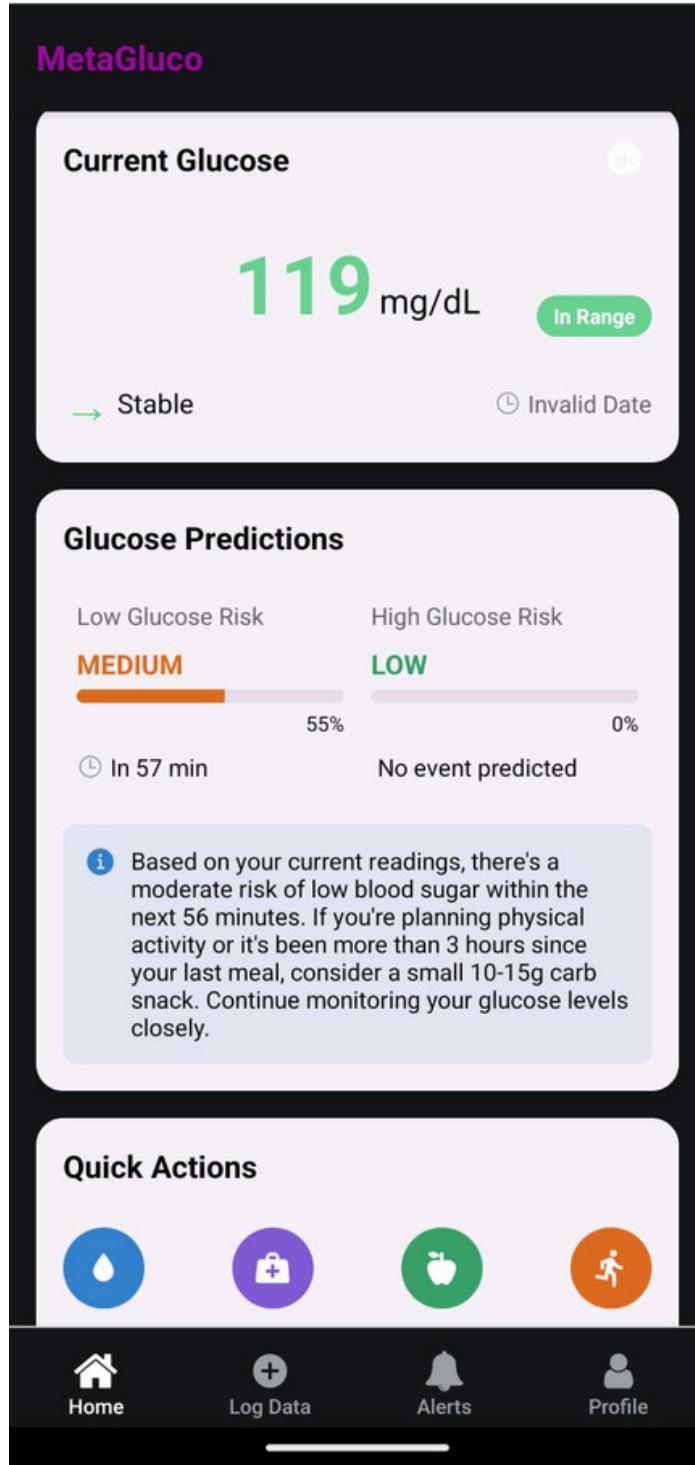
# Medllama2 Implementation

```
Command Prompt
llama_model_loader: - kv 17:     llama.attention.value_length u32      = 128
llama_model_loader: - kv 18:         general.file_type u32      = 15
llama_model_loader: - kv 19:             llama.vocab_size u32      = 128256
llama_model_loader: - kv 20:                 llama.rope.dimension_count u32      = 128
llama_model_loader: - kv 21:                     tokenizer.ggml.model str      = gpt2
llama_model_loader: - kv 22:                         tokenizer.ggml.pre str      = [llama-bpe
llama_model_loader: - kv 23:                             tokenizer.ggml.tokens arr[str,128256]      = ["!", "\", "#", "$", "%", "&", "'", ...
llama_model_loader: - kv 24:                               tokenizer.ggml.token_type arr[i32,128256]      = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
llama_model_loader: - kv 25:                                 tokenizer.ggml.merges arr[str,280147]      = ["-á -á", "-á -á-á", "-á -á -á-á", ...
llama_model_loader: - kv 26:                                     tokenizer.ggml.bos_token_id u32      = 128000
llama_model_loader: - kv 27:                                         tokenizer.ggml.eos_token_id u32      = 128009
llama_model_loader: - kv 28:                                             tokenizer.chat_template str      = {{- bos_token }}\n{%- if custom_tools ...
llama_model_loader: - kv 29:                                                 general.quantization_version u32      = 2
llama_model_loader: - type f32:  58 tensors
llama_model_loader: - type q4_K: 168 tensors
llama_model_loader: - type q6_K: 29 tensors
llm_load_vocab: special tokens cache size = 256
llm_load_vocab: token to piece cache size = 0.7999 MB
llm_load_print_meta: format      = GGUF V3 (latest)
llm_load_print_meta: arch       = llama
llm_load_print_meta: vocab type = BPE
llm_load_print_meta: n_vocab    = 128256
llm_load_print_meta: n_merges   = 280147
llm_load_print_meta: vocab_only = 1
llm_load_print_meta: model type = ?B
llm_load_print_meta: model ftype = all F32
llm_load_print_meta: model params = 3.21 B
llm_load_print_meta: model size  = 1.87 GiB (5.01 BPW)
llm_load_print_meta: general.name = Llama 3.2 3B Instruct
llm_load_print_meta: BOS token  = 128000 '<|begin_of_text|>'
llm_load_print_meta: EOS token  = 128009 '<|eot_id|>'
llm_load_print_meta: LF token   = 128 '|ä'
llm_load_print_meta: EOT token  = 128009 '<|eot_id|>'
llm_load_print_meta: EOM token  = 128008 '<|eom_id|>'
llm_load_print_meta: EOG token  = 128008 '<|eom_id|>'
llm_load_print_meta: max token length = 256
llama_model_load: vocab only - skipping tensors
[GIN] 2025/03/17 - 15:41:23 | 200 | 6.3882555s | 127.0.0.1 | POST "/api/generate"
```

The screenshot shows a developer's workspace with several open windows:

- EXPLORER**: Shows the project structure under the folder "METAGLUCO".
- RECOMMENDATION SERVICE CODE**: A Python file named "recommendation\_service.py" containing code for a recommendation service using OLLAMA. It includes routes for health checks and generating recommendations based on user data.
- TERMINAL**: A terminal window showing the output of a command prompt, likely related to the model loading process.
- PROBLEMS**: A list of log messages from the application, including prediction results and connection tests.
- INFO**: A detailed log of the prediction process, including token types, merges, and specific tokens like BOS and EOS.

# Frontend



# Frontend

**Log Glucose**

Glucose Reading (mg/dL)  
127

Low ————— High

Low <70    Normal 70-180    High >180

When was this reading taken?  
3/18/2025, 6:49:58 AM

Meal Context  
Before... After... Fasting Bedtime

Notes  
Add any additional notes

Save Reading

**Log Activity**

Activity Type  
Walking Running Cycling  
Swimming Gym Yoga  
Other

Duration (minutes)  
30

5m 30m 60m 120m 180m

15 min ✓ 30 min 45 min  
60 min 90 min

Intensity  
 Light - Easy, can talk or sing  
 Moderate - Breathing harder but can talk  
 Vigorous - Breathing hard, difficult to talk

**Log Meal/Carbs**

Carbohydrates (grams)  
0

Carb Amount  
0g 50g 100g 150g 200g

Quick Select:  
15g 30g 45g 60g 75g  
90g

Meal Type  
Breakfast Lunch Dinner  
Snack

When did you eat?  
3/18/2025, 6:50:39 AM

Meal Description  
Describe what you ate (optional)

**Alerts**

**Recommendations**

**Your Personalized Advice**

Based on your current readings, there's a moderate risk of low blood sugar within the next 56 minutes. If you're planning physical activity or it's been more than 3 hours since your last meal, consider a small 10-15g carb snack. Continue monitoring your glucose levels closely.

Home Log Data Alerts Profile

# Testing and Validation

The image shows a desktop screen with two main application windows open:

- Postman:** A tool for making API requests. The URL is `http://localhost:5050/api/predict`. The request method is POST. The body contains the following JSON payload:

```
1 {  
2   "glucose_readings": [120, 130, 140, 150, 160, 170, 180, 190, 200, 210, 220, 230],  
3   "insulin": {  
4     "basal": [0, 0, 0],  
5     "bolus": [0, 0, 0]  
6   }  
7 }
```

- MongoDB Compass:** A graphical interface for managing MongoDB databases. The left sidebar shows connections to a local MongoDB instance at `localhost:27017` and various databases: glycemic\_data, glucose\_readings, insulin\_doses, recommendations, activity\_entries, config, admin, meal\_entries, users, vitals\_entries, and local. The right panel displays the contents of the `recommendations` collection. One document is shown in detail:

```
_id: ObjectId('67d7c450ada924da70aa05c1')  
prediction_id: "006d614a-0002-45c7-99b1-a12f34010bc9"  
initiated_at: 2025-03-17T06:42:24.242+00:00  
prediction_data: Object  
  current_glucose: 0  
  hypo_probability: 0.4769461750984192  
  hyper_probability: 0  
  time_to_hypo: 53.06391143798828  
  time_to_hyper: null  
request_data: Object  
  current_glucose: 0  
  trend: "stable"  
recent_insulin: Object  
  basal: 0  
  bolus: 0  
  time_since_last_bolus: 0  
recent_carbs: 30  
recent_activity: 45  
latest_reading_time: "2025-03-17T02:39:45"  
status: "completed"  
completed_at: 2025-03-17T06:42:30.770+00:00
```

# Standards and Best Practices

- Mobile: React Native component architecture with reusable UI elements
- Backend: RESTful API design with proper endpoint naming and HTTP methods
- ML: Time-series data processing for continuous CGM readings
- AI: MedLLama2 prompt engineering for accurate medical recommendations
- Data: Real-time data streaming architecture for CGM integration
- Security: Encrypted medical data storage following healthcare standards
- Testing: Model validation with historical glucose datasets
- Prediction: Confidence intervals for glycemic event predictions
- Ethics: Responsible AI implementation with human oversight

# Commercialization Potential

- Market Size: 1.2 million diabetes patients in Sri Lanka (8.7% prevalence rate)
- Growth Rate: 4.7% annual increase in diabetes diagnoses
- Target Users: 35% of patients use smartphones for health management
- Cost Savings: Potential 30% reduction in hospitalization costs
- Revenue Model: Freemium subscription (\$5/month premium features)
- Expansion Potential: Adaptable to neighboring countries (India, Bangladesh)
- Competitive Edge: 15-20 minute earlier prediction than existing solutions
- ROI: Good return on development and maintenance costs within few months

# References

- M. Gadaleta, A. Facchinetti, E. Grisan, and M. Rossi, "Prediction of Adverse Glycemic Events From Continuous Glucose Monitoring Signal," IEEE Journal of Biomedical and Health Informatics, vol. PP, no. 99, pp. 1-1, April 2018, doi: 10.1109/JBHI.2018.2823763. [1]
- N. Nuwarapaksha, H.M.R.K.G. Nandasena, and N. Nanayakkara, "Prevalence of Long-term Complications among Type 2 Diabetes Mellitus Patients at National Hospital - Kandy, Sri Lanka," Diabetes Research and Clinical Practice, vol. 186S, p. 109535, January 2022. [2]
- Deldjoo, Y., He, Z., & McAuley, J., "A Review of Modern Recommender Systems Using Generative Models (Gen-RecSys)," Polytechnic University of Bari, Bari, Italy; University of California, La Jolla, USA.[3]
- Lin, Y., Liu, Y., Lin, F., Miao, C., "A Survey on Reinforcement Learning for Recommender Systems," September 2021.[4]
- Syafrudin, M., Alfian, G., Fitriyani, N. L., Anshari, M., "Future Glycemic Events Prediction Model Based On Artificial Neural Network," 2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), November 2022, [5]
- Crilly, P., Uddin, N., "A mixed method study investigating the use of the Sugar Smart app to help community pharmacists reduce public sugar intake," Royal Pharmaceutical Society (RPS) Winter Summit 2017, London, December 2017.[6]
- Zhu, Y., Yi, J., Xie, J., Chen, Z., "Deep Causal Reasoning for Recommendations," arXiv preprint arXiv:2201.01593, 6 Jan 2022 (v1), last revised 21 Nov 2022 (v2).[7]



**IT21287718 | WIMANSA P.P.H.S.D.**

**SPECIALIZATION – DATA SCIENCE**

**SIDE EFFECTS PREDICTION AND ALERT  
MECHANISM FOR DIABETES TREATMENT**

# INTRODUCTION

- Diabetes affects 537M adults worldwide (IDF 2021)
- Medication side effects reduce treatment adherence
- Existing solutions focus on glucose monitoring, not side effect prediction
- Research goal: Predict and alert side effects in real time

# RESEARCH PROBLEM

- Side effects of diabetes treatment affect patient adherence
- Current systems reactively manage side effects, not predict them
- Existing machine learning models focus on diagnosis, not treatment effects

# RESEARCH OBJECTIVES

- Develop an ML-based system for side effect prediction

- Train a robust model for personalized prediction

- Design an intuitive mobile interface for logging symptoms and receiving alerts

# DATA SET

## CDTOD Diabetes Side Effects Prediction Dataset

- Patient demographics (ID, Date, Age, Gender, BMI)
- Health metrics (Glucose, Insulin levels)
- Reported side effects (Headache, Nausea, Dizziness, Fatigue, etc.)
- Lifestyle factors (Sleep Hours, Diet Quality)

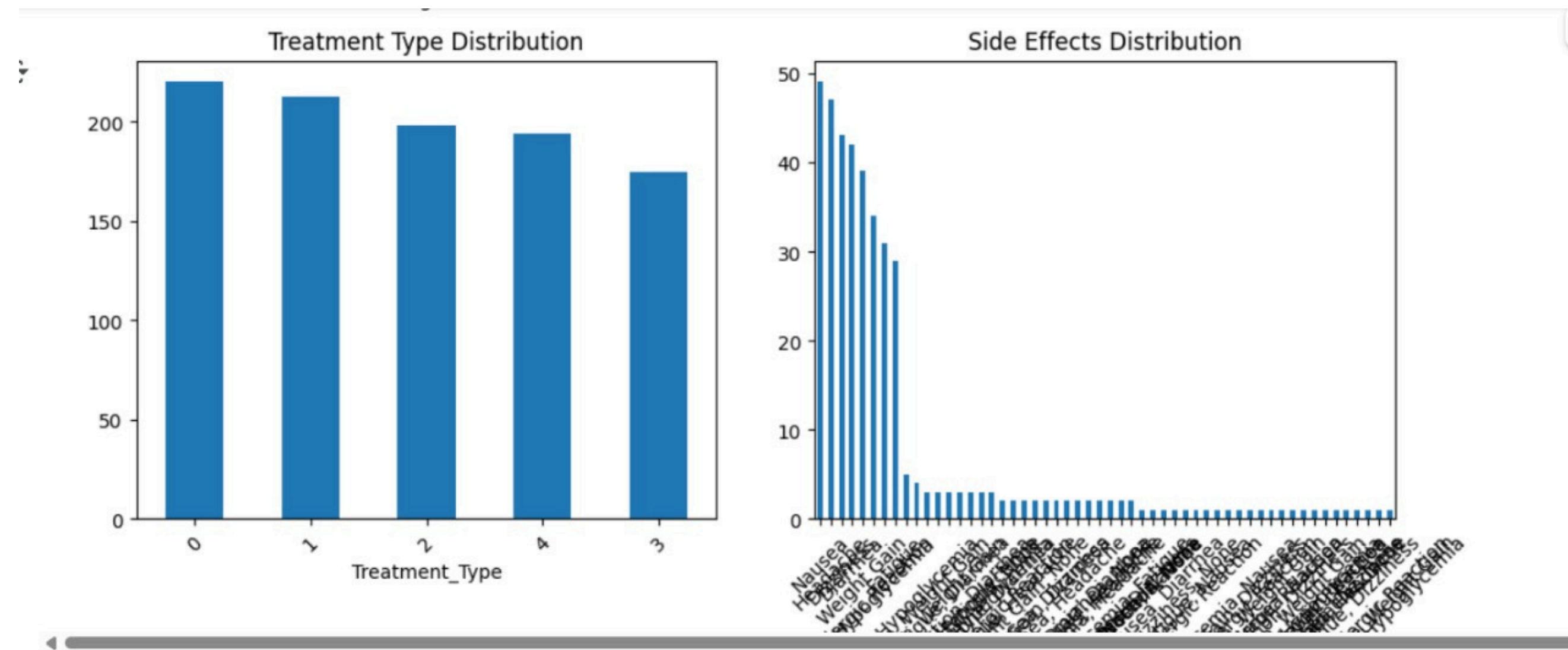
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
1	PatientID	Day	Glucose	Insulin	BloodPres	HeartRate	Headache	Nausea	Vomiting	BlurryVisio	Sweating	Fatigue	Dizziness	Irritability	Cough	StomachU	JointPain	Age	Gender	Weight	Hypertensi	KidneyDise	Neuropath	Med
2	1	Day1	91	14	97	83	0	0	0	0	0	0	0	1	0	0	0	0	68	Male	118	0	0	0
3	1	Day2	159	7	134	62	0	0	0	0	0	0	0	0	0	0	0	1	68	Male	118	0	0	0
4	1	Day3	98	36	98	71	0	0	1	0	0	0	0	0	1	1	0	0	68	Male	118	0	0	0
5	1	Day4	148	29	92	92	0	0	0	0	1	0	0	0	1	0	0	1	68	Male	118	0	0	0
6	1	Day5	187	48	98	80	0	0	0	0	0	0	0	0	0	0	0	0	68	Male	118	0	0	0
7	1	Day6	161	15	112	107	0	0	0	0	0	0	0	0	0	0	0	0	68	Male	118	0	0	0
8	1	Day7	227	5	97	114	0	0	0	0	0	0	0	0	0	0	0	1	68	Male	118	0	0	0
9	1	Day8	118	13	124	119	0	0	0	0	0	0	0	0	0	0	0	0	68	Male	118	0	0	0
10	1	Day9	109	18	109	108	0	0	0	0	0	0	0	1	0	0	0	1	68	Male	118	0	0	0
11	1	Day10	142	13	108	106	0	0	0	0	0	0	0	0	0	0	0	0	68	Male	118	0	0	0
12	1	Day11	157	34	130	63	0	0	0	0	0	0	1	0	0	0	0	0	68	Male	118	0	0	0
13	1	Day12	90	25	110	64	0	0	0	0	0	0	0	0	0	1	0	0	68	Male	118	0	0	0
14	1	Day13	247	16	101	105	0	0	0	0	0	0	0	0	0	0	0	0	68	Male	118	0	0	0
15	1	Day14	229	16	134	62	0	1	0	0	0	0	0	0	0	0	0	0	68	Male	118	0	0	0
16	1	Day15	215	23	131	60	0	0	0	0	0	0	0	0	0	0	0	1	68	Male	118	0	0	0



# DATA SET

A	B	C	D	E	F	G	H	I	J	K
Patient_ID	Age	Gender	Diabetes_Type	Treatment	Medication	Dosage	Treatment	Side_Effect	Severity_Score	
1	56	Male	Gestational	Insulin	Glargine	68.92	36	Allergic Re	9	
2	69	Male	Type 1	Metformin	Sitagliptin	12.71	20	None	0	
3	46	Female	Gestational	Insulin	Glipizide	95.39	18	None	0	
4	32	Male	Type 1	GLP-1 rece	Liraglutide	84.66	38	None	0	
5	60	Male	Gestational	Insulin	Metformin	81.48	67	Hypoglyce	8	
6	25	Female	Type 2	DPP-4 inhi	Metformin	83.18	23	Headache	3	
7	78	Male	Gestational	Insulin	Sitagliptin	93.61	96	Nausea	4	
8	38	Male	Gestational	GLP-1 rece	Lispro	56.7	109	None	0	
9	56	Male	Type 1	DPP-4 inhi	Metformin	24.03	16	Headache	3	
10	75	Male	Type 2	Sulfonylure	Liraglutide	63.59	59	None	0	
11	36	Male	Gestational	Sulfonylure	Sitagliptin	75.57	69	None	0	
12	40	Female	Type 2	Insulin	Glipizide	75.1	68	None	0	
13	28	Male	Gestational	Insulin	Liraglutide	54.54	21	Allergic Re	9	
14	28	Male	Gestational	GLP-1 rece	Sitagliptin	11.5	104	Headache	3	
15	41	Male	Type 2	GLP-1 rece	Liraglutide	40.26	115	None	0	
16	70	Male	Gestational	Metformin	Metformin	92.47	78	None	0	
17	53	Male	Type 2	GLP-1 rece	Glipizide	60.52	103	Nausea	4	
18	57	Female	Gestational	Metformin	Sitagliptin	56.14	95	None	0	
19	41	Male	Type 1	DPP-4 inhi	Glargine	30.54	78	Dizziness,	4	
20	20	Female	Type 2	Insulin	Glipizide	40	70	None	0	
21	39	Female	Gestational	DPP-4 inhi	Glargine	90.06	12	None	0	
22	70	Male	Gestational	GLP-1 rece	Liraglutide	68.31	16	None	0	
23	19	Male	Type 1	DPP-4 inhi	Lispro	79.8	81	None	0	
24	41	Male	Type 2	DPP-4 inhi	Glipizide	48.16	109	Headache	3	
25	61	Male	Gestational	DPP-4 inhi	Liraglutide	64.87	87	None	0	
26	47	Female	Type 1	GLP-1 rece	Glargine	28.6	14	Headache	3	
27	55	Male	Gestational	DPP-4 inhi	Metformin	72.02	61	None	0	

# DATA SET



# TRAINING MODELS

**Model Used - LSTM (Long Short-Term Memory).**

**Why LSTM?**

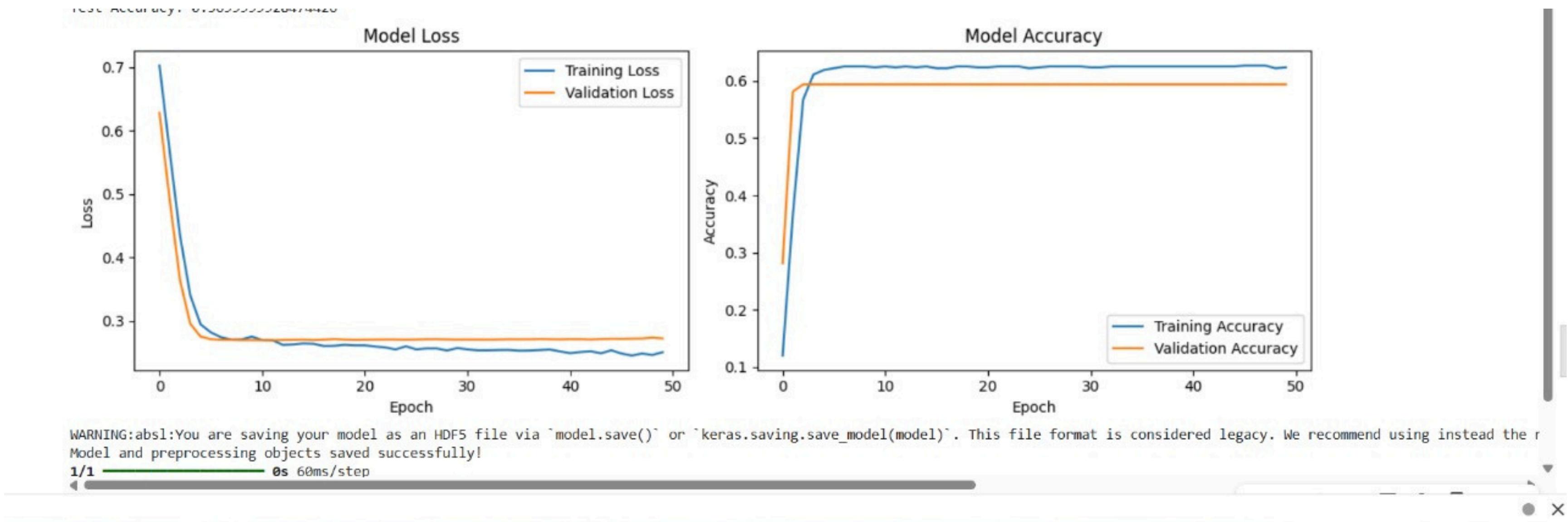
- **Captures sequential trends in daily symptom logs.**
- **This model learns from 30-day patient history to predict tomorrow's side effects.**
- **Handles sequential dependencies, capturing patterns in glucose, insulin, and symptoms.**

```
✓ 0s [9] # Build LSTM model
    input_layer = Input(shape=(x_train_scaled.shape[1], 1))
    bi_lstm = Bidirectional(LSTM(64, return_sequences=True))(input_layer)
    attention = MultiHeadAttention(num_heads=4, key_dim=64)(bi_lstm, bi_lstm)
    norm = LayerNormalization()(attention)
    flatten = Flatten()(norm)
    dropout = Dropout(0.3)(flatten)
    output_layer = Dense(11, activation='sigmoid')(dropout)

    model = Model(inputs=input_layer, outputs=output_layer)
    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

✓ 5m # Train model
    history = model.fit(x_train_scaled, y_train, epochs=20, batch_size=32, validation_data=(x_test_scaled, y_test))
```

# TRAINING MODELS



# PROOF OF COMPLETION

The screenshot shows the Postman application interface. On the left, the sidebar displays 'My Workspace' with sections for Collections, Environments, Flows, and History. Under Environments, 'Employee Management- MERN' is selected, and under it, 'GET New Request' is highlighted. The main workspace shows a POST request to 'http://localhost:5070/predict'. The 'Body' tab is selected, showing the following JSON input:

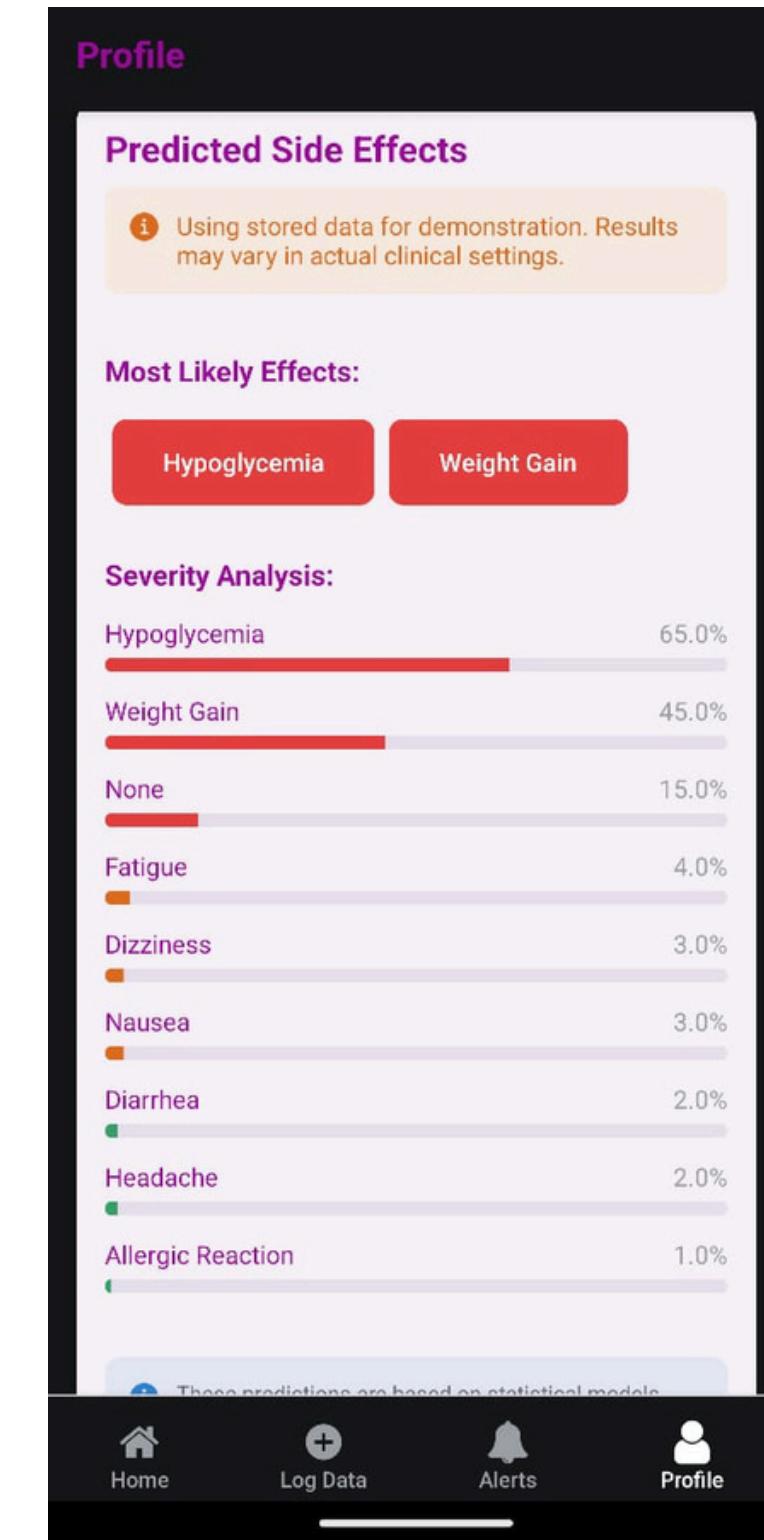
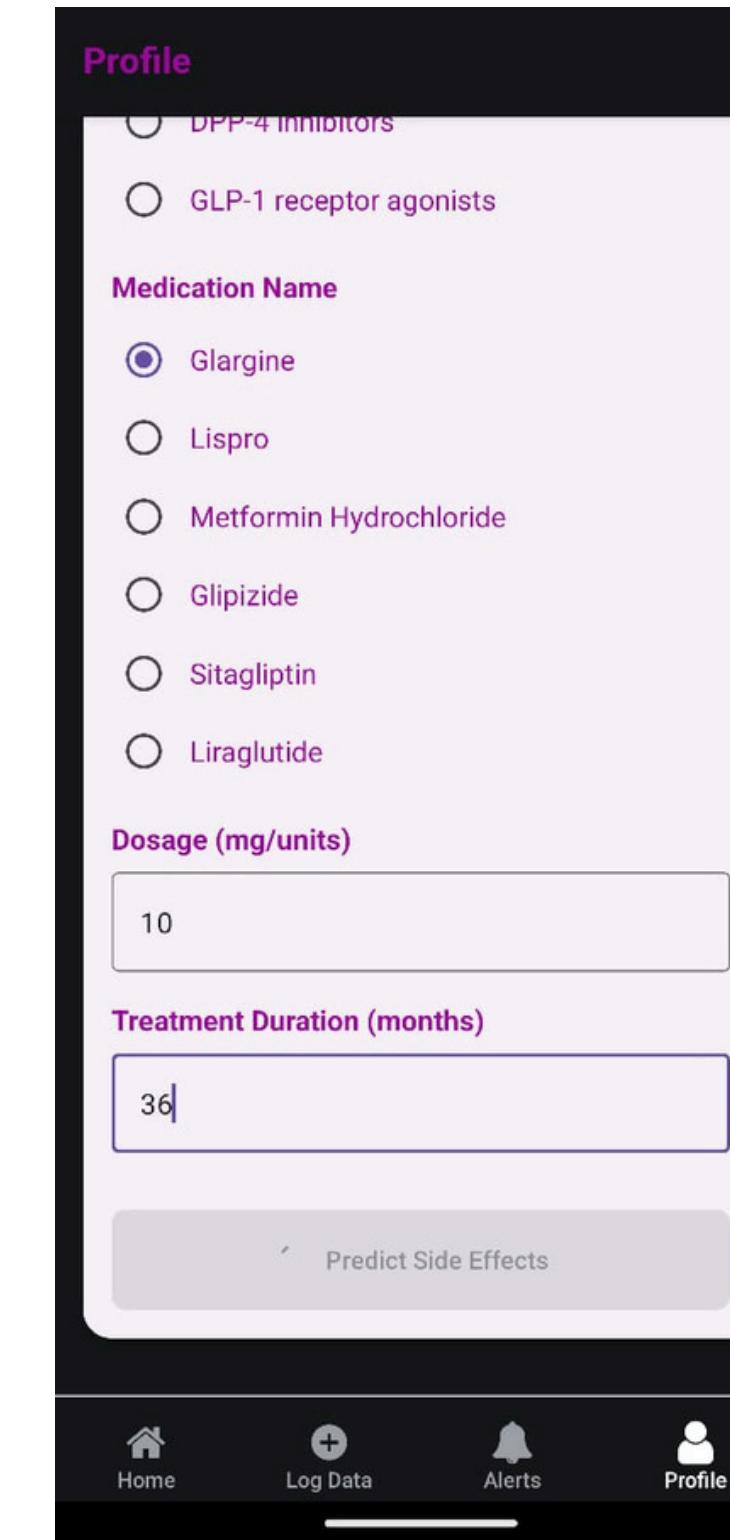
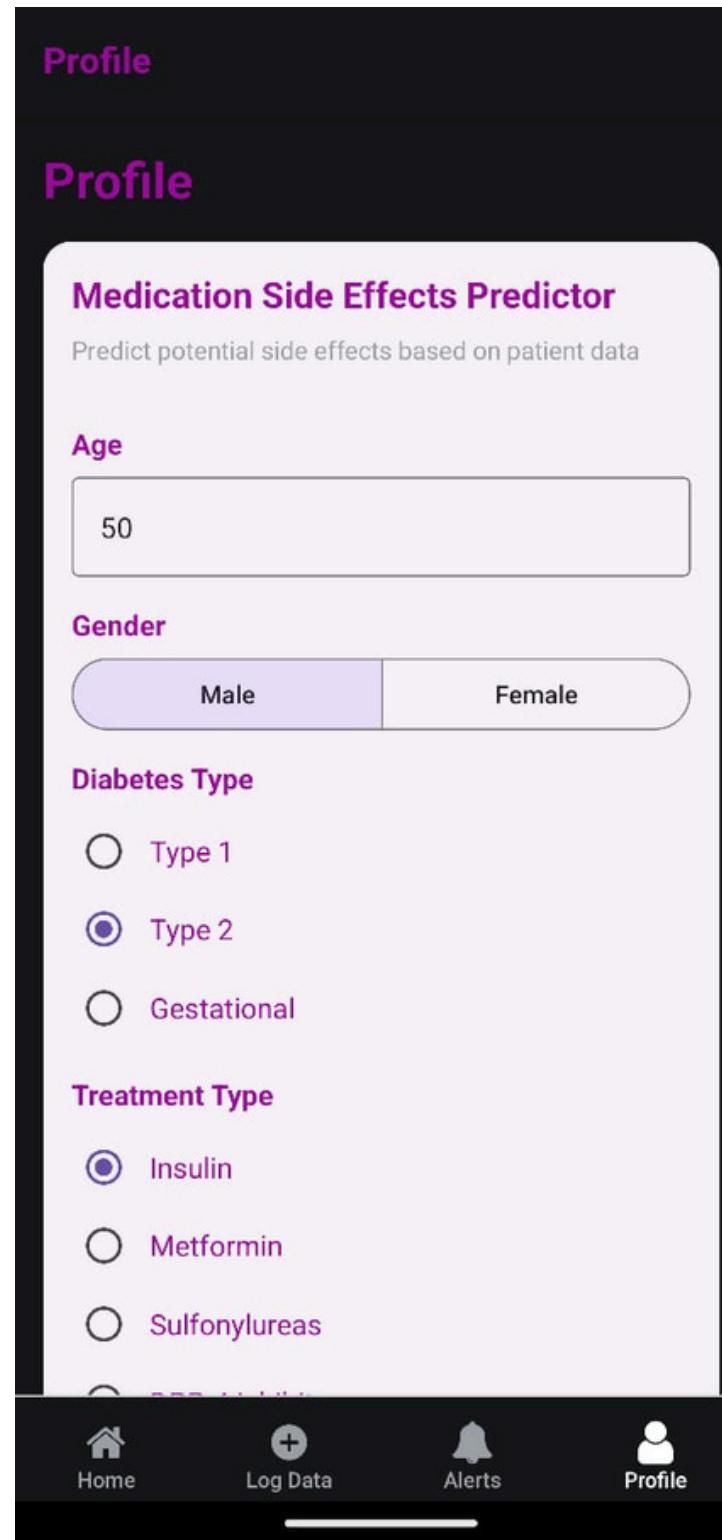
```
1 {  
2   "Age": 75,  
3   "Gender": "Male",  
4   "Diabetes_Type": "Type 2",  
5   "Treatment_Type": "Insulin",  
6   "Medication_Name": "Glargine",  
7   "Dosage": 110.0,  
8   "Treatment_Duration": 6  
9 }
```

Below the request, the response is displayed. The status bar indicates '200 OK' with a response time of '58 ms' and a size of '777 B'. The response body is shown in JSON format:

```
10   "None",  
11   "Weight Gain"  
12 ],  
13   "predicted_side_effects": [  
14     "Diarrhea",  
15     "None"  
16   ],  
17   "severity_scores": {  
18     "Allergic Reaction": 0.029948722571134567,  
19     "Diarrhea": 0.08479944616556168,  
20     "Dizziness": 0.015525388531386852,  
21     "Fatigue": 0.017201270908117294,  
22     "Headache": 0.021724001201202774
```

At the bottom, there are various status indicators and navigation links.

# PROOF OF COMPLETION



# REFERENCES

**M. S. Abdul Rahman, S. M. Rahimifard, and M. M. Al-Shami, "Prediction of DiabetesRelated Complications Using Machine Learning Algorithms: A Comprehensive Review," in 2021 IEEE International Conference on Artificial Intelligence and Machine Learning (AIML), 2021, pp. 1-6.**

**N. Jothi and A. A. N. Islam, "A Framework for Predicting Adverse Drug Reactions in Diabetic Patients Using Machine Learning Techniques," in 2021 IEEE International Conference on Biomedical Engineering and Sciences (IECBES), 2021, pp. 212-217**

**R. Al-Mahdi, M. Elshamli, and A. S. Mokhtar, "A Machine Learning Approach for RealTime Prediction of Side Effects in Diabetes Treatment," in 2022 IEEE International Conference on Healthcare Informatics (ICHI), 2022, pp. 45-50.**

**J. Wu, J. Zhang, and Z. Li, "Mobile Health Application for Diabetes Management: RealTime Prediction of Blood Glucose Levels and Side Effects," in 2021 IEEE International Conference on Health Informatics (ICHI), 2021, pp. 74-80.**



**IT21283062 | DE SILVA L.K.N.**

**SPECIALIZATION - DATA SCIENCE**

**NUTRITIONAL GUIDANCE SYSTEM WITH**

**PERSONALIZED MEAL PLANS**

# INTRODUCTION

Managing diabetes requires a careful balance of nutrition and lifestyle adjustments. This research proposes a machine learning-based personalized meal recommendation system designed for diabetic patients. The system uses patient profiles, including health metrics, dietary preferences, and allergies, to generate meal plans tailored to their diabetes type and other variables.

# RESEARCH PROBLEMS

Lack of real-time data utilization  
for dynamically adjusting meal  
plans

Generic recommendations with  
insufficient deep personalization  
for individual users.

Inaccurate or non-personalized  
predictive models for glucose  
response to meals.

# SPECIFIC OBJECTIVE

*To create a personalized nutritional guidance system that uses dietary, medical, and activity data to provide optimized meal plans for better diabetes management.*

# SUB OBJECTIVES

- ***Comprehensive Data Integration***
- ***Personalized Nutritional Guidance***
- ***Optimization of Meal Plans***
- ***Improvement of Diabetes Management Outcomes***

# DATA COLLECTION

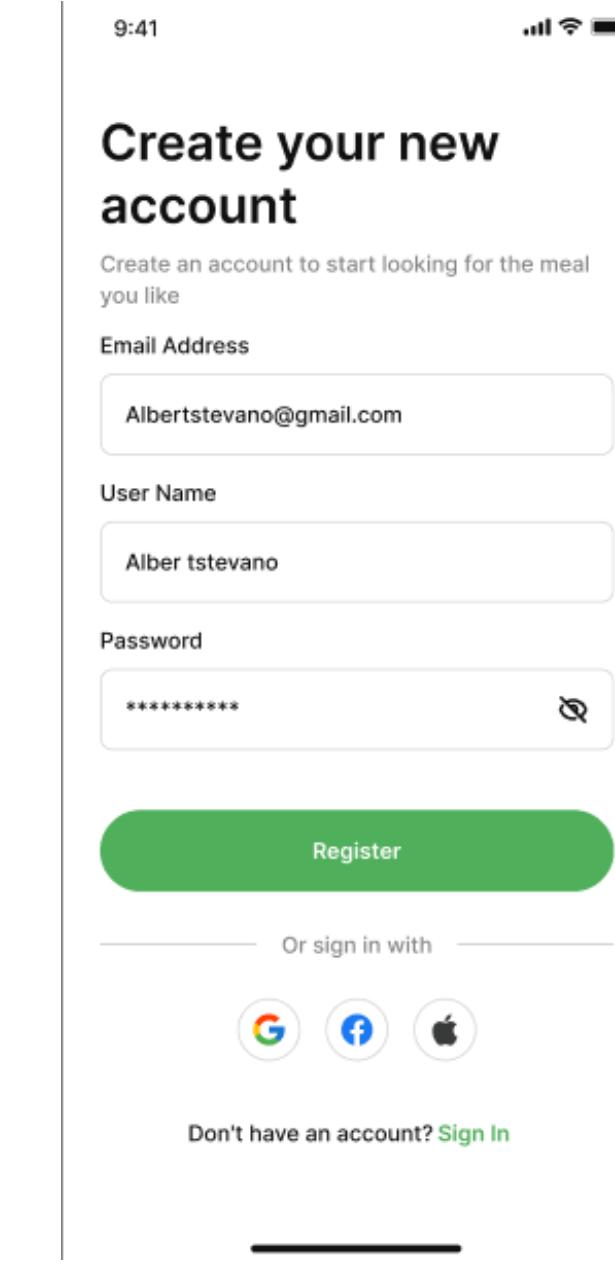
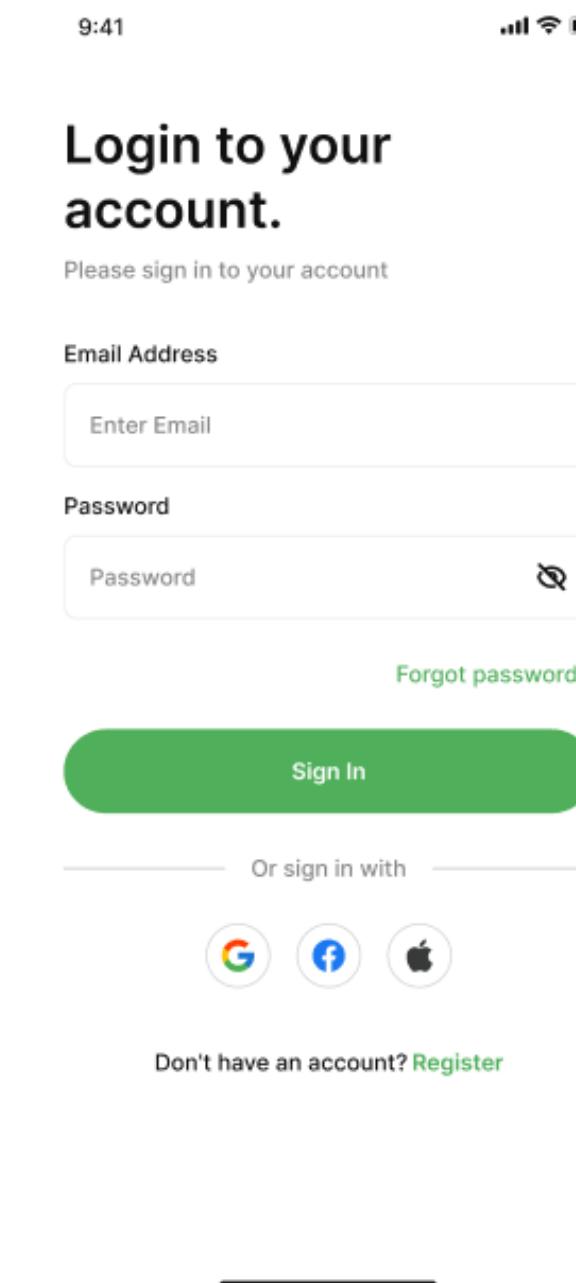
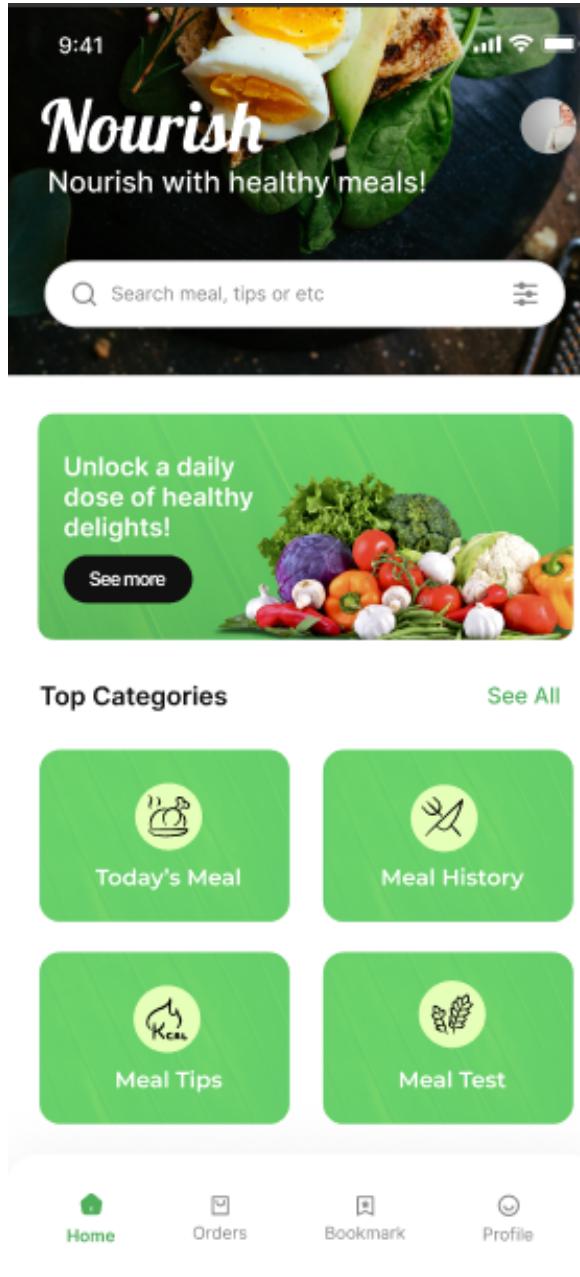
Python

	RecordID	Name	Age	Gender	Height	Weight	Location	Occupation	DiabetesType	DiagnosedYearsAgo	...	DietFollowed	TriggerFoods	Allergies	Int
0	1	Nimal Fernando	64	Male	168	80	Jaffna	Homemaker	Gestational	2	...	Vegetarian	NaN	Dairy	
1	2	Kumari Rathnayake	55	Female	178	96	Kandy	Office worker	Type 1	14	...	Low-carb	Sugary snacks	Nuts	
2	3	Anjali Perera	57	Female	180	55	Jaffna	Homemaker	Type 1	10	...	NaN	Sugary snacks	Dairy	
3	4	Amara Wijesinghe	48	Female	156	56	Colombo	Office worker	Type 1	4	...	Vegetarian	Sugary snacks	Dairy	
4	5	Kumari Rathnayake	70	Female	175	50	Colombo	Homemaker	Type 2	5	...	Vegetarian	White bread	NaN	

MealID	MealName	MealDetails	CalorieCount	AllergyStatus	Preferences	Type
1	Traditional Sri Lankan Breakfast	String hoppers, pol sambol, dhal curry	350	Contains coconut	Vegetarian	Breakfast
1	Rice and Curry Lunch	White rice, fish curry, mallung, papadam	600	Contains fish, gluten	Non-Vegetarian	Lunch
1	Light Dinner	Vegetable soup, brown bread	250	Contains gluten	Vegetarian	Dinner
2	Coconut Milk Rice Breakfast	Kiribath, lunu miris	300	Contains coconut	Vegetarian	Breakfast
2	Seafood Curry Lunch	White rice, prawn curry, gotukola salad	650	Contains shellfish	Non-Vegetarian	Lunch
2	String Hopper Dinner	String hoppers, chicken curry	450	Contains gluten	Non-Vegetarian	Dinner
3	Healthy Breakfast	Oats porridge, banana, nuts	320	Contains nuts	Vegetarian	Breakfast
3	Chicken Fried Rice Lunch	Brown rice, chicken, vegetables	700	Contains egg, soy	Non-Vegetarian	Lunch
3	Vegetable Roti Dinner	Whole wheat roti, mixed vegetables	400	Contains gluten	Vegetarian	Dinner
4	Spicy Breakfast	Hoppers, lunu miris, dhal curry	300	Contains coconut	Vegetarian	Breakfast
4	Fish Curry Lunch	Red rice, fish curry, beetroot salad	550	Contains fish	Non-Vegetarian	Lunch

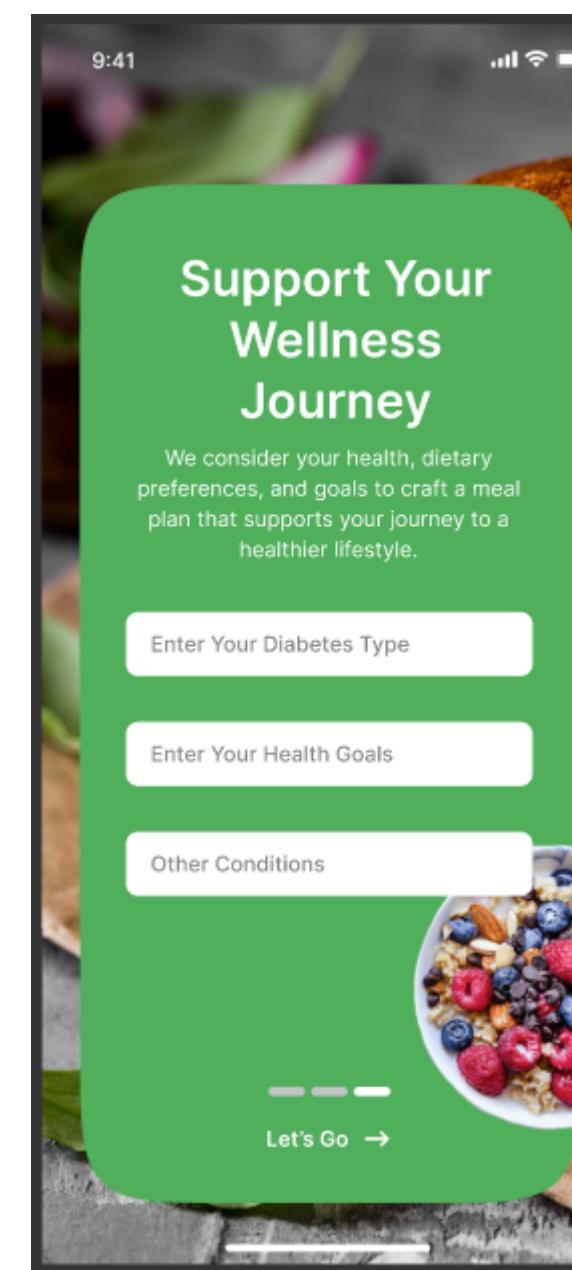
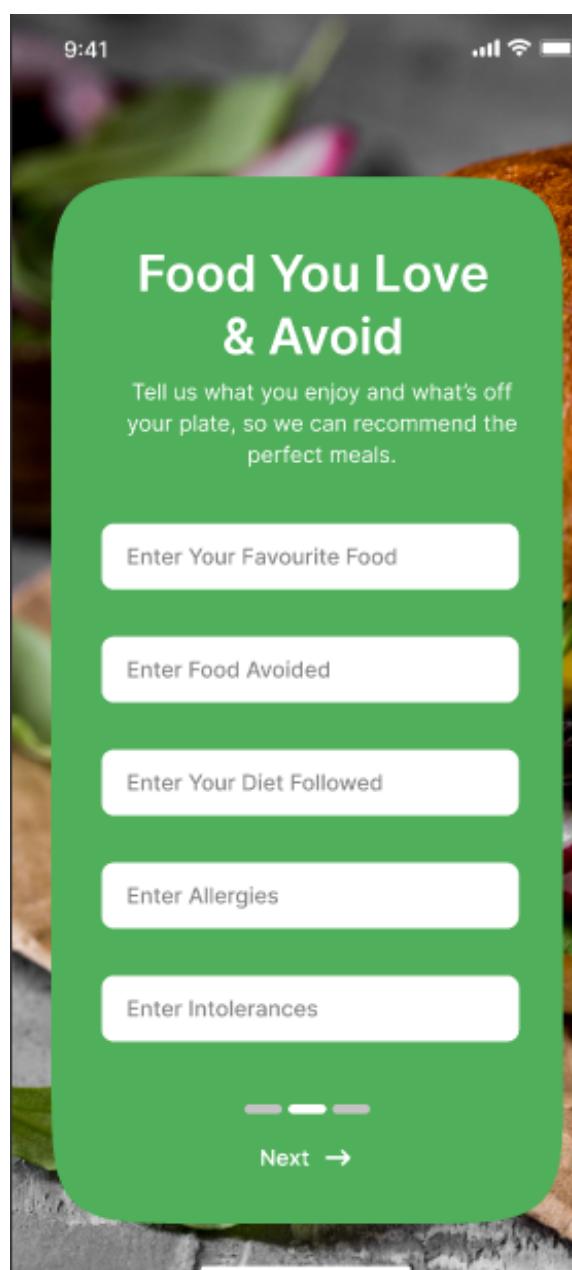
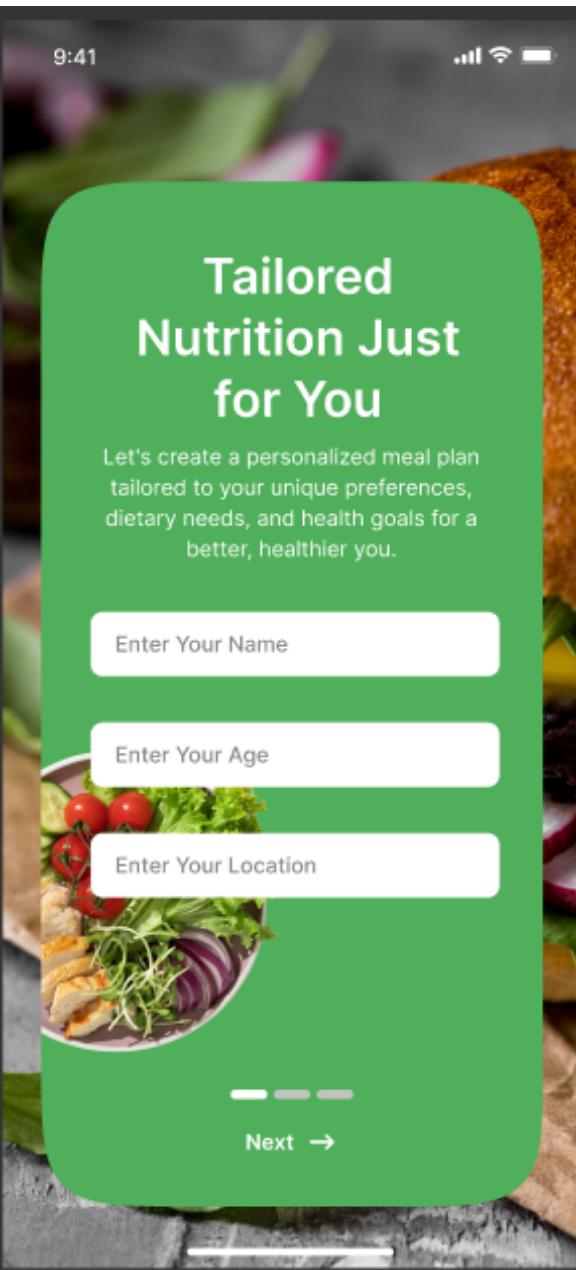
# PROOF OF COMPLETION

## MOBILE APPLICATION



# PROOF OF COMPLETION

## MOBILE APPLICATION



# PROOF OF COMPLETION

## MOBILE APPLICATION

The screenshots demonstrate the mobile application's interface for creating and managing meal plans.

**Screenshot 1: Today's Meal Plan**  
Shows a date picker for November 2022. The 25th is selected. Below it, a meal plan entry for "Avocado Toast with Poached Egg" is shown, including details, preferences, and allergies.

**Screenshot 2: Today's Meal Plan**  
Shows another meal plan entry for "Avocado Toast with Poached Egg".

**Screenshot 3: Meal Plan - 001**  
Shows a detailed view of the meal plan. It includes sections for Breakfast, Lunch, and Dinner, each listing a meal item with its details, preferences, allergies, and calorie count.

**Screenshot 4: Meal History**  
Shows a history of five meal plans from February 2025, each with a title, date, and time (12:00 PM). The meal plans are numbered 001 through 005.

# PROOF OF COMPLETION

## MOBILE APPLICATION



The image displays two side-by-side screenshots of a mobile application interface titled "Meal Profile". Both screenshots show a user profile with the initials "AS" in a circular icon. The left screenshot shows a list of fields with placeholder text, while the right screenshot shows the same fields filled with specific data. The right screenshot includes decorative blue sparkles.

Field	Left Screenshot (Placeholder)	Right Screenshot (Filled)
Full Name	Albert Stevano Bajefski	Albert Stevano Bajefski
Age	Albert Stevano Bajefski	Albert Stevano Bajefski
Location	Albert Stevano Bajefski	Albert Stevano Bajefski
Favourite Food	19/06/1999	19/06/1999
Food Avoided	19/06/1999	19/06/1999
Diet Followed	+1 325-433-7856	+1 325-433-7856
Allergies	albertstevano@gmail.com	albertstevano@gmail.com
Intolerances	albertstevano@gmail.com	albertstevano@gmail.com
Diabetes Type	albertstevano@gmail.com	albertstevano@gmail.com
Health Goal	albertstevano@gmail.com	albertstevano@gmail.com
Other Conditions	albertstevano@gmail.com	albertstevano@gmail.com

**Left Screenshot Fields (Placeholder):**

- Full Name
- Age
- Location
- Favourite Food
- Food Avoided
- Diet Followed
- Allergies
- Intolerances
- Diabetes Type
- Health Goal
- Other Conditions

**Right Screenshot Fields (Filled):**

- Full Name: Albert Stevano Bajefski
- Age: Albert Stevano Bajefski
- Location: Albert Stevano Bajefski
- Favourite Food: 19/06/1999
- Food Avoided: 19/06/1999
- Diet Followed: +1 325-433-7856
- Allergies: albertstevano@gmail.com
- Intolerances: albertstevano@gmail.com
- Diabetes Type: albertstevano@gmail.com
- Health Goal: albertstevano@gmail.com
- Other Conditions: albertstevano@gmail.com

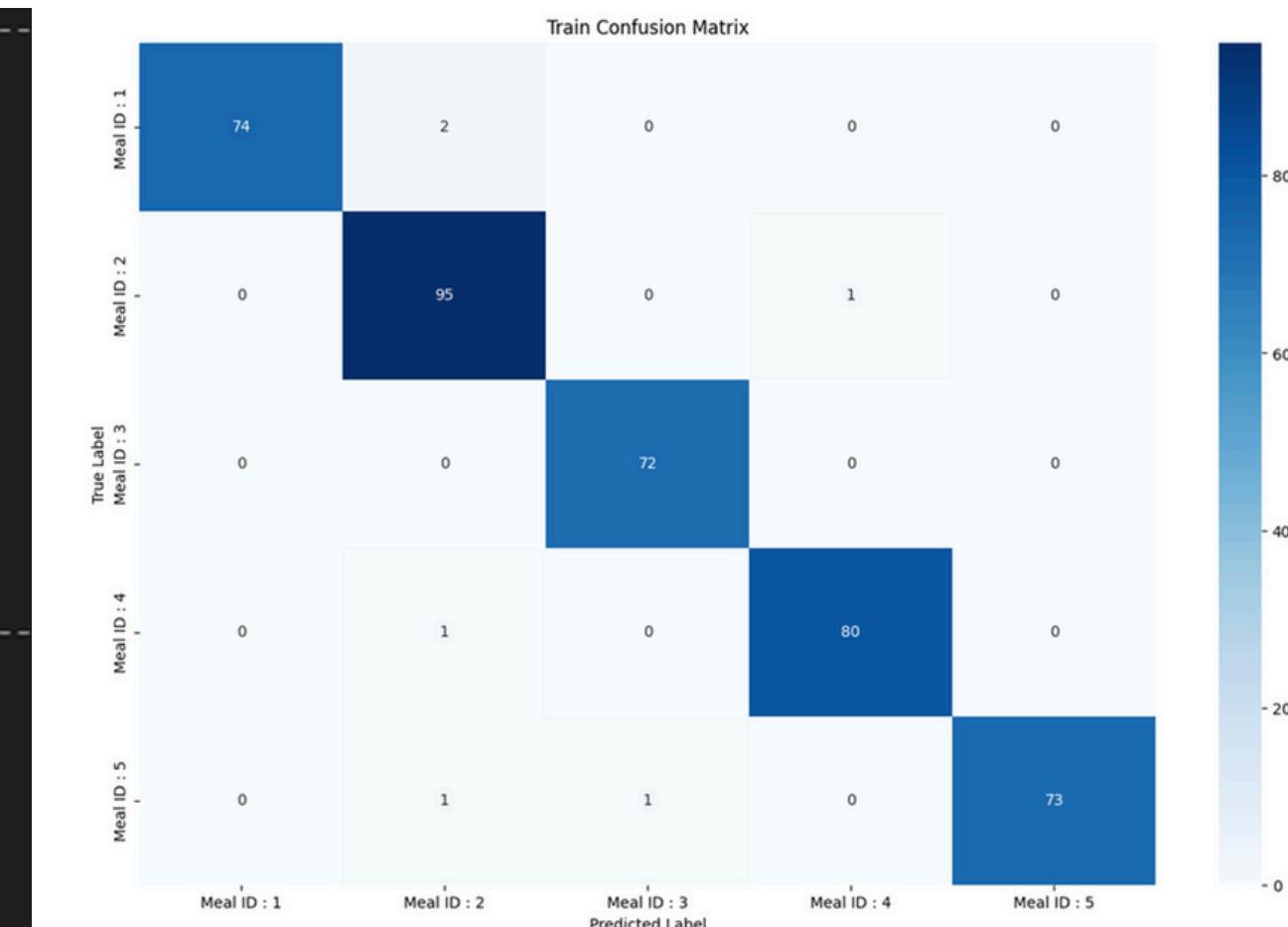
# PROOF OF COMPLETION

## IMPLEMENTATION - CATBOOSTCLASSIFIER MODEL

- ENSEMBLE LEARNING - BOOSTING FRAMEWORK
- CLASSIFICATION TASKS
- CATEGORICAL DATA

```
> cat = CatBoostClassifier(  
    iterations=200,  
    learning_rate=0.1,  
    loss_function='MultiClass',  
    depth=6  
)  
  
cat.fit(  
    X, Y,  
    eval_set=(  
        X_test,  
        Y_test  
    ),  
    verbose=100  
)  
54] ✓ 0.5s
```

```
..----- Train CLS REPORT -----  
precision recall f1-score support  
  
Meal ID : 1 1.00 0.97 0.99 76  
Meal ID : 2 0.96 0.99 0.97 96  
Meal ID : 3 0.99 1.00 0.99 72  
Meal ID : 4 0.99 0.99 0.99 81  
Meal ID : 5 1.00 0.97 0.99 75  
  
accuracy 0.98 400  
macro avg 0.99 0.98 0.99 400  
weighted avg 0.99 0.98 0.99 400  
  
..----- Test CLS REPORT -----  
precision recall f1-score support  
  
Meal ID : 1 1.00 1.00 1.00 22  
Meal ID : 2 0.95 1.00 0.98 20  
Meal ID : 3 1.00 1.00 1.00 26  
Meal ID : 4 1.00 1.00 1.00 16  
Meal ID : 5 1.00 0.94 0.97 16  
  
accuracy 0.99 100  
macro avg 0.99 0.99 0.99 100  
weighted avg 0.99 0.99 0.99 100
```

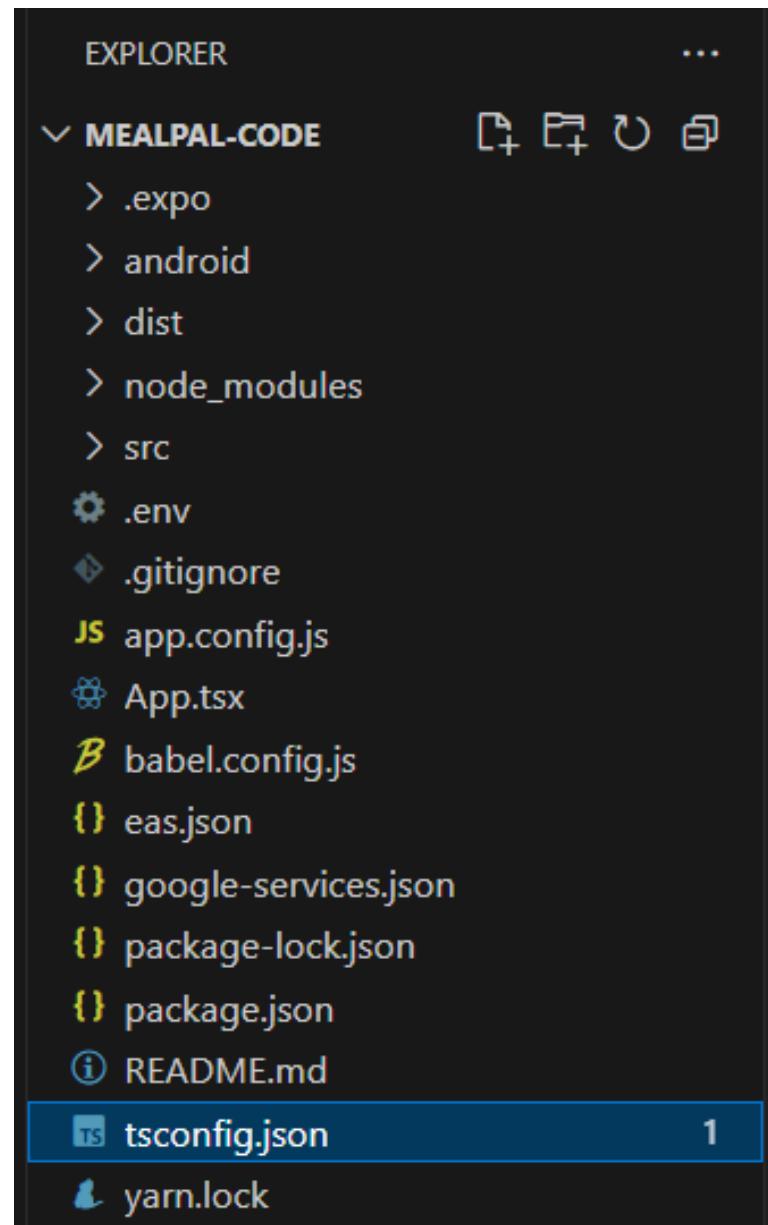


## EVALUATING THE MODEL

- CLASSIFICATION REPORT
- CONFUSION MATRIX

# PROOF OF COMPLETION

## IMPLEMENTATION - CODE SNIPPETS



```
JS app.config.js X
JS app.config.js > default > splash
1
2  export default ({ config }) => {
3    return {
4      ...config,
5      name: "MySugarApp",
6      slug: "MySugarApp",
7      owner: "mysugar-app",
8      version: "1.0.0",
9      orientation: "portrait",
10     icon: "./src/assets/images/adaptive-icon.png", // Icons & UI Preferences
11     scheme: "myapp",
12     userInterfaceStyle: "automatic",
13     splash: { // Splash Screen Configuration
14       image: "./src/assets/images/splash.png",
15       resizeMode: "contain",
16       backgroundColor: "#ffffff",
17     },
18     ios: {
19       supportsTablet: true,
20     },
21     bundleIdentifier: "com.mysugarapp",
22   },
23   android: {
24     ...
25   }
26 }

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
C/C++: 2 warnings generated.

Deprecated Gradle features were used in this build, making it incompatible with Gradle 9.0.

You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugins.

For more on this, please refer to https://docs.gradle.org/8.8/userguide/command_line_interface.html#sec:command_line_warnings in the Gradle documentation.

BUILD SUCCESSFUL in 6m 17s
679 actionable tasks: 194 executed, 97 from cache, 388 up-to-date
Starting Metro Bundler
```

App.config.js

# PROOF OF COMPLETION

## IMPLEMENTATION - CODE SNIPPETS

```
src > screens > auth > DetailsScreenOne.tsx > ...
1 import React from "react";
2 import { Image, ImageBackground, StyleSheet, Text, View } from "react-native";
3 import { Button } from "react-native-paper";
4 import { Formik } from "formik";
5 import * as Yup from "yup";
6 import ScrollViewWrapper from "@/components/wrappers/ScrollViewWrapper";
7 import { LightPalette } from "@/theme";
8 import { CustomTextInput } from "@/components/customComponents/VhTextInput";
9 import StepperIndicator from "@/components/customComponents/StepperIndicator";
10 import { routeKeys } from "@/navigation/config";
11 import { useCustomNavigation } from "@/hooks";
12 import CustomDropdown from "@/components/customComponents/customDropdown";
13 import { useRoute } from "@react-navigation/native";
14
15 //validation rules
16 const validationSchema = Yup.object().shape({
17   Name: Yup.string().required("Name is required"),
18   Age: Yup.number().required("Age is required"),
19   Location: Yup.string().required("Location is required"),
20   Weight: Yup.string().required("Weight is required"),
21 });
22
23 const DetailsScreenOne = () => {
24   const navigation = useCustomNavigation();
25   const route = useRoute();
26
27   return (
28     <ImageBackground
29       source={require("../assets/images/details-bg.png")}
30       style={styles.background}
31     >
```

DetailsScreenOne.tsx

```
src > services > getMeals.ts > ...
1 // services/postService.ts
2
3 import { storage } from "@/localStorage";
4 import { postData, postData } from "@/query/config";
5 import { ToastAndroid } from "react-native";
6
7 // API function to fetch posts
8 export const getMeals = async (data) => {
9   try {
10     const response = await postData("predict_meal", data);
11     // console.log("Users", response);
12     return response; // Return response data
13   } catch (error) {
14     ToastAndroid.show("An error occurred getting data", 1000);
15
16     throw error; // Rethrow error for React Query to handle
17   }
18 };
19
```

getMeals.js

# PROOF OF COMPLETION

## IMPLEMENTATION - CODE SNIPPETS

```
{ google-services.json }
```

```
{ google-services.json > ...}
```

```
1 { "project_info": {
```

```
2   "project_number": "799795395727",
```

```
3   "project_id": "mealplan-6e07d",
```

```
4   "storage_bucket": "mealplan-6e07d.firebaseio.storage.app"
```

```
5 },
```

```
6 "client": [
```

```
7   {
```

```
8     "client_info": {
```

```
9       "mobilesdk_app_id": "1:799795395727:android:a5c7a1f72661d5011604fa",
```

```
10      "android_client_info": {
```

```
11        "package_name": "com.mysugarapp"
```

```
12      }
```

```
13    },
```

```
14    "oauth_client": [],
```

```
15    "api_key": [
```

```
16      {
```

```
17        "current_key": "AIzaSyB6cnlUomTi6kAVZYk5nZvkh7rlfN0P8aY"
```

```
18      }
```

```
19    ],
```

```
20  ],
```

```
21  "services": {
```

```
22    "appinvite_service": {
```

```
23      "other_platform_oauth_client": []
```

```
24    }
```

```
25  },
```

```
26 },
```

```
27 ],
```

```
28 "configuration_version": "1"
```

```
29 }
```

google-service.json

```
{ TodaysMealScreen.tsx }
```

```
src > screens > home > TodaysMealScreen.tsx > ...
```

```
1 import { StyleSheet, ToastAndroid, View } from "react-native";
```

```
2 import React, { useEffect, useState } from "react";
```

```
3 import ScrollViewWrapper from "@/components/wrappers/ScrollViewWrapper";
```

```
4 import CustomHeader from "@/components/customComponents/CustomHeader";
```

```
5 import { CustomButton } from "@/components/customComponents/CustomButton";
```

```
6 import MealAccordion from "@/components/customComponents/MealAccordion";
```

```
7 import { getMeals } from "@/services/getMeals";
```

```
8 import { storage, storageKeys } from "@/localStorage";
```

```
9 import { getFormattedDate } from "@/utils/getFormattedDate";
```

```
10 import { useMMKVObject } from "react-native-mmkv";
```

```
11
```

```
12 const TodaysMealScreen = () => {
```

```
13   const [todaysMeal, setTodaysMeal] = useState([]);
```

```
14   const [loading, setLoading] = useState(true);
```

```
15   const [userDetails, setUserDetails] = useMMKVObject(storageKeys.USER_DETAILS);
```

```
16
```

```
17   const getTodayMeal = async () => {
```

```
18     try {
```

```
19       setTodaysMeal([]);
```

```
20       setLoading(true);
```

```
21     }
```

```
22     const todayMeal = await getMeals(userDetails);
```

```
23
```

```
24     setTodaysMeal(todayMeal.Meals);
```

```
25     setLoading(false);
```

```
26   } catch (error) {
```

```
27     console.log("EEE", error);
```

```
28     setLoading(false);
```

```
29   }
```

TodayMealScreen.tsx

# COMPLETED TASKS

- Data Cleaning and Preprocessing & Data Analysis
- Created a model to generate and recommend meal plan
- Improved the model accuracy for better performance.
- Designed Wireframes & UI/UX Designs.
- Developed a mobile app for meal recommendation.



# FUTURE WORKS

- develop feedback system for future improvement.
- Mobile app Intergration

# METHODOLOGY

## FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

### FUNCTIONAL REQUIREMENTS

- USER MANAGEMENT
- DATA COLLECTION & INTEGRATION
- MEAL PLAN RECOMMENDATION
- MOBILE APP FEATURES
- BLOOD GLUCOSE PREDICTION & ANALYSIS

### NON-FUNCTIONAL REQUIREMENTS

- PERFORMANCE & SCALABILITY
- SECURITY & PRIVACY
- AVAILABILITY & RELIABILITY
- USABILITY & ACCESSIBILITY

# REFERENCES

- "Mobile health applications for managing diabetes: A systematic review" by El-Gayar, O., Timsina, P., Nawar, N., & Eid, W. (2013). Journal of Diabetes Science and Technology.
- "Data preprocessing techniques for classification without noise in real-world" by García, S., Luengo J., & Herrera, F. (2015). Knowledge and Information Systems.
- "Collaborative filtering for dietary recommendations" by Harvey, M., Ludwig, B., & Elsweiler, D. (2013). Proceedings of the 7th ACM conference on Recommender systems
- "Adaptive learning algorithms for personalized diet recommendations" by Ueta, T., Koizumi, Y., & Shirota, S. (2019) IEEE Access

# THANK YOU!