ST. XAVIER'S COLLEGE (AUTONOMOUS)

(Recognized as 'College with Potential for Excellence' by UGC)

(Accredited by NAAC with "A++" Grade with a CGPA of 3.66 out of 4 in IV Cycle)

PALAYAMKOTTAI – 627002

DEPARTMENT OF COMPUTER APPLICATIONS

FITNORA (A SMART WORKOUT AND FOOD TRACKING APP)

PROJECT REVIEW – I

Submitted By

SUBBU RAKESH S B

(Register No.: 23UCA503)

Under the Guidance of

Prof. E. ALEX PRABAHAR, M.Sc., M.Phil.,

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

Fitnora (A Smart Workout and Food Tracking App) is an Android-based fitness application designed to combine workout tracking and diet monitoring within a single, easy-to-use platform. The application addresses the inconvenience of using multiple fitness apps by providing a unified solution for managing daily workouts and nutrition.

Fitnora allows users to log workout activities such as exercises, sets, repetitions, weights, and cardio duration. Users can also track nutritional information including calorie intake, protein consumption, and body weight. After registration and login, users can record their daily progress through a simple and intuitive mobile interface, enabling consistent tracking without complex setup or configuration.

The application follows a local-first design, where workout, food tracking and progress data are stored securely on the user's device, ensuring offline access, fast performance, and full control over personal data. Fitnora also provides an optional cloud backup feature to allow users to back up and restore their data when needed.

The mobile application is developed using Flutter to deliver a smooth, responsive, and cross-platform experience. Local data storage is managed using an embedded database (SQFLITE), while cloud services are used only for optional backup. This approach results in a lightweight, privacy-focused, and practical solution for combined workout and diet tracking.

1. INTRODUCTION

In the modern world, fitness and health have become a primary priority for individuals across all age groups. While many people actively pursue fitness goals such as weight loss, muscle gain, and overall well-being, only a limited number successfully achieve consistent results. One of the major reasons for this gap is the lack of proper tracking of workout routines and nutritional intake. Even disciplined individuals often experience slow or inconsistent progress due to fragmented monitoring methods and the absence of structured guidance.

Existing fitness applications such as **HealthifyMe** and **Hevy** provide partial solutions but introduce usability and functionality challenges. **HealthifyMe** is not specifically tailored for users focused on muscle building, while **Hevy** lacks integrated food and nutrition tracking features. Additionally, users are often forced to rely on multiple applications to manage workouts, diet, and progress analysis, increasing complexity and reducing consistency.

To address these limitations, **Fitnora** is proposed as a unified fitness tracking platform that simplifies workout routine creation, nutrition tracking, and progress monitoring under a single system. The application aims to provide an intuitive and streamlined experience for both trainees and fitness coaches, enabling structured planning, real-time progress tracking, and data-driven fitness improvements through a single, integrated solution.

2. SYSTEM STUDY

2.1 EXISTING SYSTEM

In the existing system, users are required to switch between multiple applications to track their fitness progress effectively. For workout routine management, applications such as Hevy are used, while food and nutrition tracking require separate applications like HealthifyMe. This fragmentation makes it difficult for users to maintain consistency and accurately monitor their overall progress.

- **Multiple Application Dependency:** Users must rely on more than one application to track workouts, diet, and progress, leading to inconvenience and reduced user engagement.
- **Lack of Integrated Progress Analysis**: Since workout data and nutrition data are stored in separate platforms, generating a combined and detailed progress analysis becomes difficult.
- **Data Redundancy:** Measurements and body weight data are redundant, and users are required to update the same information in two or more applications.

2.2 PROPOSED SYSTEM

In the proposed system, **Fitnora** provides a single integrated application to track fitness progress efficiently. Workout routines, food and nutrition tracking, and progress monitoring are combined into one platform. This integration helps users maintain consistency and accurately monitor their overall fitness progress.

- **Single Application Platform:** Users can manage workouts, diet, and progress using one application, reducing dependency on multiple platforms.
- **Integrated Progress Analysis:** Workout and nutrition data are stored together, enabling detailed and accurate progress analysis.
- **Elimination of Data Redundancy:** Measurements and body weight data are stored once and updated within a single system, avoiding duplicate entries.

2.3 PROBLEM DEFINITION AND PROJECT DESCRIPTION

The project titled **"FITNORA – Online Workout and Nutrition Tracking System"** is a mobile-based application designed to manage and track fitness activities and nutritional intake efficiently. The Fitnora application allows users to create workout routines, log daily exercises, and track food consumption within a single platform. To access the system, users are required to register, and upon successful registration, they can log in to the application to manage their fitness activities and monitor progress.

The system stores user data, workout records, and nutrition details in a secure local database, ensuring fast access and offline availability. Additionally, the application supports data backup and synchronization features, enabling users to restore their fitness data when required and preventing data loss. The primary objective of Fitnora is to simplify fitness tracking and provide users with a reliable, user-friendly platform to support consistent progress toward their health and muscle-building goals.

USER MODULE:

- Manage Account
- Manage exercises
- Manage routines
- Manage routine composition
- Log workout session
- Record exercise performance
- View progress overview
- Manage meal
- Manage food
- Track body measurements
- Backup data

3. SYSTEM ANALYSIS

3.1 REQUIREMENTS SPECIFICATION

Hardware Requirements

Processor	:	Intel i3 or Higher
RAM	:	Minimum 16 GB RAM
Internal Storage	:	Minimum 100 GB internal storage
Internet Connectivity	:	Wi-Fi / Mobile Data
Keyboard	:	Standard 104-keys Keyboard
Mouse	:	Optical Mouse

Software Requirements

Operating System	:	Windows 10 / 11 (64 bit)
Local Server	:	Golang with Gin framework
Database	:	MySQL (Backend), SQFLITE (Mobile)
Frontend	:	Flutter (Android)
Backend	:	Golang
Database Connectivity	:	GORM (Backend), SQFLITE (Mobile)
Android SDK Version	:	API 30 (Android 11)
Other Tools	:	Visual Studio Code, Android Studio, Gemini AI Studio, MySQL (XAMPP)

3.2 FEASIBILITY STUDY

A feasibility study is conducted to evaluate the practicality and viability of the proposed Fitnora (A Smart Workout and Food Tracking App). The study analyzes resource availability, development cost, operational efficiency, and long-term benefits. Fitnora aims to integrate workout routine management and food tracking into a single, user-friendly platform, eliminating the need for multiple applications.

The feasibility study is examined under the following aspects:

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

TECHNICAL FEASIBILITY

Technical feasibility examines whether the required hardware, software, and technologies are available to develop Fitnora within the defined constraints.

- Fitnora is developed using modern and reliable technologies such as Flutter for mobile application development, Go (Golang) for backend services, and (MySQL + SQFLITE) for structured data storage, ensuring scalability and performance.
- The system ensures accuracy and reliability in tracking workout routines, nutrition data, and user progress.
- Basic security mechanisms are implemented to protect user data and ensure safe access to the application.

Hence, the proposed system is technically feasible.

OPERATIONAL FEASIBILITY

Operational feasibility evaluates whether Fitnora can be effectively used in real-time fitness tracking scenarios.

- The system provides a simple and user-friendly interface, reducing the learning curve for users.
- Fitnora integrates workout tracking and food logging into a single application, minimizing user effort and resistance.
- The application supports smooth operation and requires minimal maintenance.

Therefore, the system is operationally feasible.

ECONOMIC FEASIBILITY

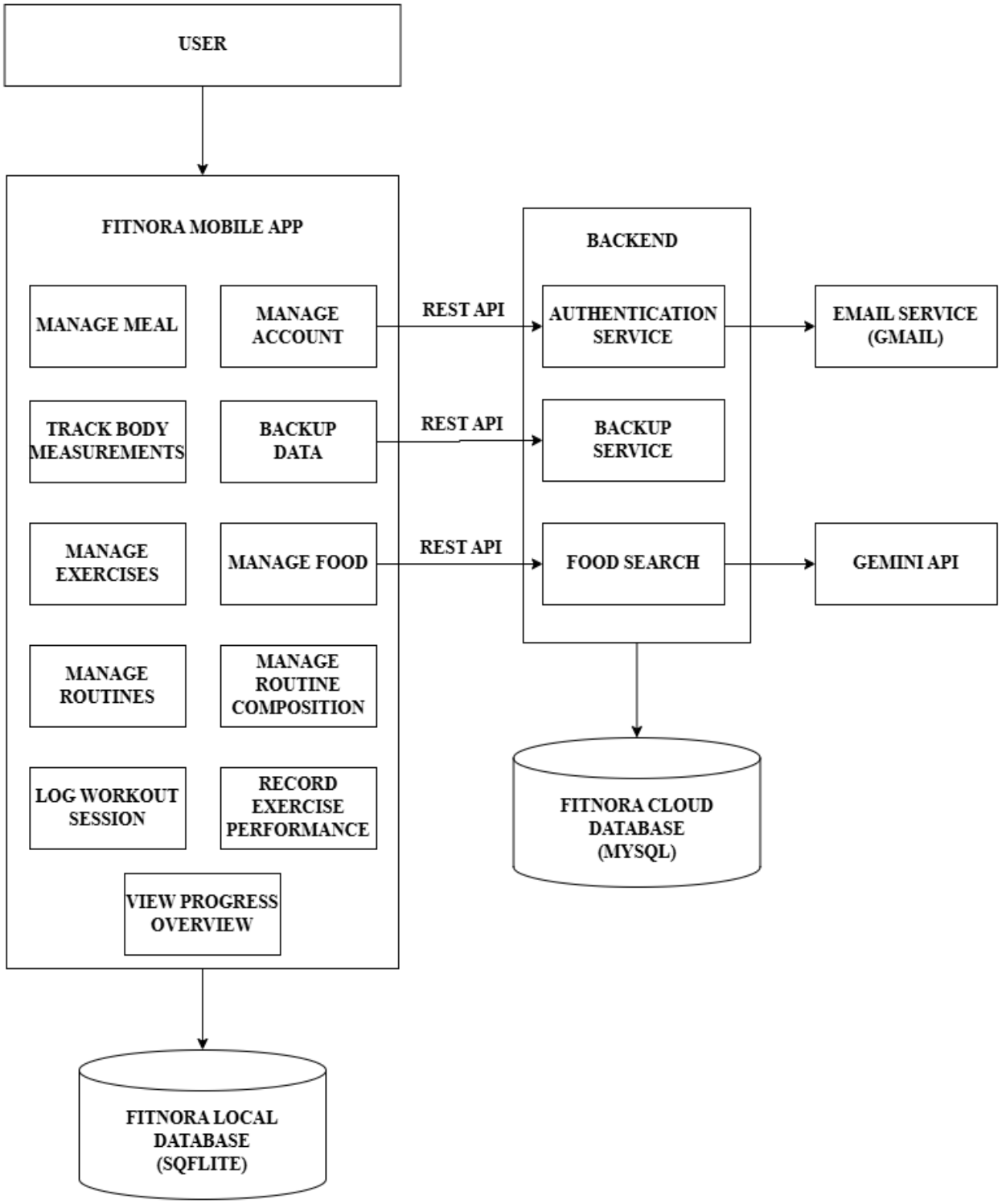
Economic feasibility determines whether the cost involved in developing Fitnora is justified by its benefits.

- The system uses open-source tools and existing mobile infrastructure, reducing development and operational costs.
- No additional hardware investment is required beyond standard Android devices and global backend server.
- The long-term benefits of efficient fitness tracking outweigh the initial development cost.
- Because of the use of a local database, very few requests are sent to the backend.

Thus, Fitnora is economically feasible.

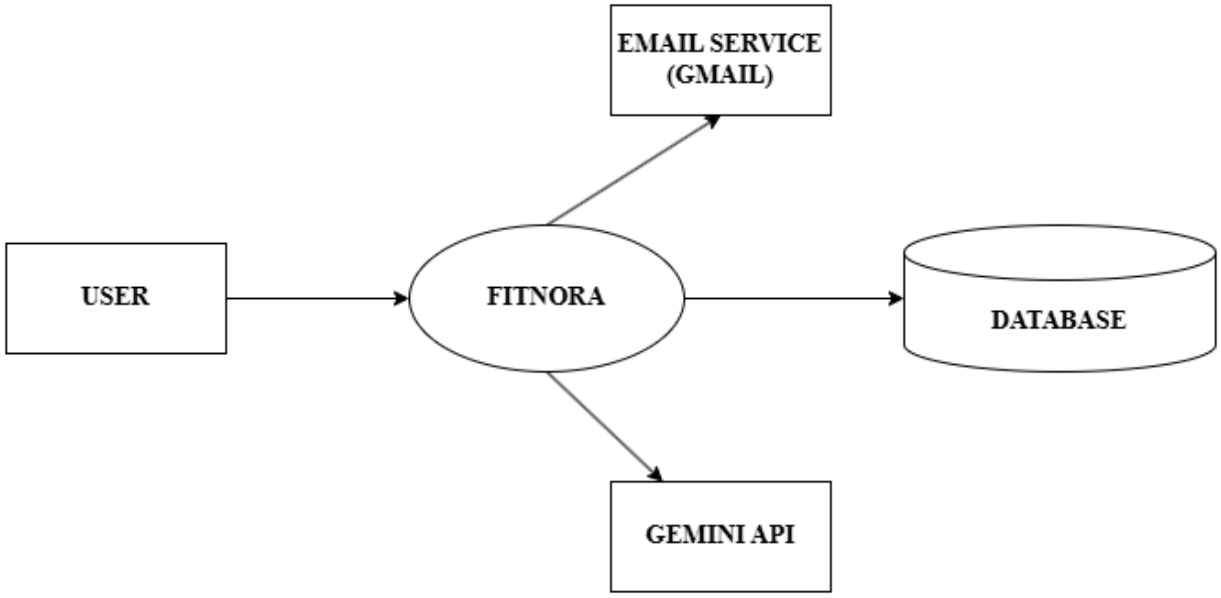
4. SYSTEM DESIGN

4.1 ARCHITECTURAL DESIGN



4.2 DATA FLOW DIAGRAM

DFD Level 0:



DFD Level 1:

EMAIL SERVICE (GMAIL)
MANAGE ACCOUNT
USERS
MANAGE EXERCISES
EXERCISES
MANAGE ROUTINES
ROUTINES
MANAGE ROUTINE COMPOSITION
ROUTINE_EXERCISE
LOG WORKOUT SESSION
SESSIONS
USER
RECORD EXERCISE PERFORMANCE
SESSION_EXERCISES
SESSION_SETS
VIEW PROGRESS OVERVIEW
MANAGE MEAL
MEALS
TRACK BODY MEASUREMENTS
BODY_MEASUREMENTS
BACKUP DATA
DATA_BACKUP
GEMINI API
MANAGE FOOD
FOODS

4.3 DATA DICTIONARY

Table name: users

Purpose: To store the details of the user

S. No.	Field Name	Data Type	Constraint	Description
1	user_id	Integer	Primary Key, Auto Increment	User unique id
2	user_email	Varchar (255)	Not Null, Unique	User email
3	user_fullname	Varchar (100)	Not Null	User full name
4	user_dob	Date	Not Null	User date of birth
5	password_hash	Text	Not Null	User hashed password
6	gender	Varchar (20)	Not Null	User gender
7	backup_enabled	Boolean	Not Null, Default	Is backup enabled
8	updated_at	Timestamp	Not Null, Default	Record updated timestamp

Table name: data_backup

Purpose: To store the backup files

S. No.	Field Name	Data Type	Constraint	Description
1	data_backup_id	Integer	Primary Key, Auto Increment	Data backup unique id
2	user_id	Integer	Foreign Key, Cascade	User id from ‘users’
3	user_dbfiles	Text	Not Null	Path for user’s db files
4	user_images	Text	Not Null	Path for user’s images
5	updated_at	Timestamp	Not Null, Default	Record updated timestamp

Table name: exercises

Purpose: To store the exercises of the user

S. No.	Field Name	Data Type	Constraint	Description
1	exercise_id	Integer	Primary Key, Auto Increment	Exercise unique id
2	exercise_name	Text	Not Null, Unique	Exercise name
3	exercise_image	Text	Not Null	Exercise image
4	exercise_equipment	Text	Not Null	Used exercise equipment
5	exercise_mode	Text	Not Null	Reps or Timer
6	is_deleted	Integer	Not Null, Default	To soft delete exercise
7	updated_at	Integer	Not Null	Record updated timestamp

Table name: routines

Purpose: To store the routines of the user

S. No.	Field Name	Data Type	Constraint	Description
1	routine_id	Integer	Primary Key, Auto Increment	Routine unique id
2	routine_name	Text	Not Null, Unique	Routine name
3	is_deleted	Integer	Not Null, Default	To soft delete routine
4	updated_at	Integer	Not Null	Record updated timestamp

Table name: routine_exercises

Purpose: Used to map the exercises to routines

S. No.	Field Name	Data Type	Constraint	Description
1	routine_exercise_id	Integer	Primary Key, Auto Increment	Exercise unique id
2	routine_id	Integer	Foreign Key, On Delete Cascade	Routine id from 'routines'
3	exercise_id	Integer	Foreign Key, On Delete Cascade	Exercise id from 'exercises'
4	exercise_order	Integer	Not Null	Used to store the order of the exercises

Table name: sessions

Purpose: To store the workout sessions of the user

S. No.	Field Name	Data Type	Constraint	Description
1	session_id	Integer	Primary Key, Auto Increment	Session unique id
2	routine_id	Integer	Foreign Key, On Delete Cascade	Routine id from 'routines'
3	executed_date	Date	Not Null	Routine executed date
4	started_at	Integer	Not Null	Starting Time
5	ended_at	Integer	Not Null	Ending Time

Table name: session_exercises

Purpose: To store the session's exercises of the user

S. No.	Field Name	Data Type	Constraint	Description
1	session_exercise_id	Integer	Primary Key, Auto Increment	Session exercise unique id
2	session_id	Integer	Foreign Key, On Delete Cascade	Session id from 'sessions'
3	exercise_id	Integer	Foreign Key, On Delete Cascade	Exercise id from 'exercises'
4	exercise_order	Integer	Not Null	Exercise executed order

Table name: sessions_sets

Purpose: To store the workout session's sets of the user

S. No.	Field Name	Data Type	Constraint	Description
1	set_id	Integer	Primary Key, Auto Increment	Set unique id
2	session_exercise_id	Integer	Foreign Key, On Delete Cascade	Session exercise id
3	reps	Integer	-	Reps executed
4	duration	Integer	-	Duration in seconds
5	weight	Real	-	Weights used

Table name: foods

Purpose: To store the food items of the user

S. No.	Field Name	Data Type	Constraint	Description
1	food_id	Integer	Primary Key, Auto Increment	Food unique id
2	food_name	Text	Not Null, Unique	Food name
3	food_calories	Real	Not Null	Food calories for 100g
4	food_protein	Real	Not Null	Food protein for 100g
5	food_carbs	Real	Not Null	Food carbs for 100g
6	food_fats	Real	Not Null	Food fats for 100g
7	food_fibers	Real	Not Null	Food fibers for 100g
8	food_source	Text	Not Null	User created or Gemini AI
9	updated_at	Integer	Not Null	Record updated timestamp

Table name: body_measurements

Purpose: To store the body measurement progress of the user

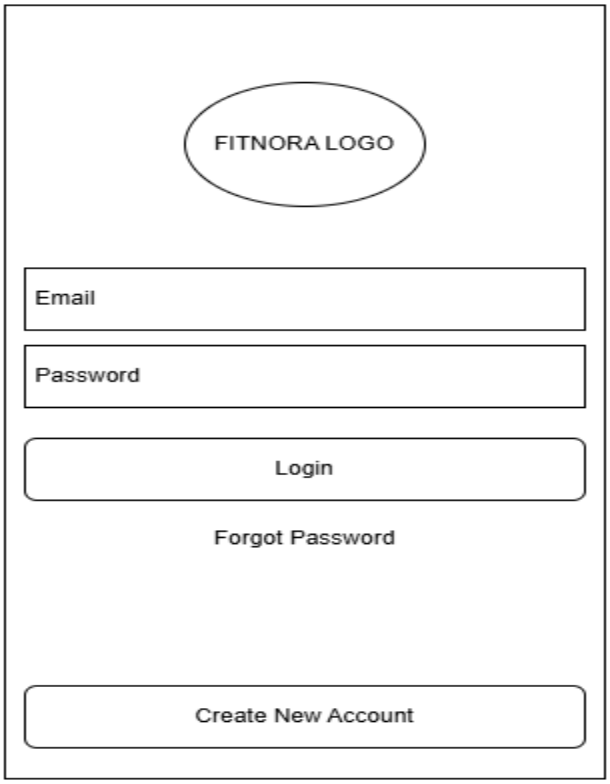
S. No.	Field Name	Data Type	Constraint	Description
1	measurement_id	Integer	Primary Key	Measurement unique id
2	user_image	Text	Not Null	Image of the user
3	user_weight	Real	Not Null	Weight of the user
4	user_height	Real	Not Null	Height of the user
3	recorded_at	Date	Not Null	Recorded date
4	updated_at	Integer	Not Null	Record updated timestamp

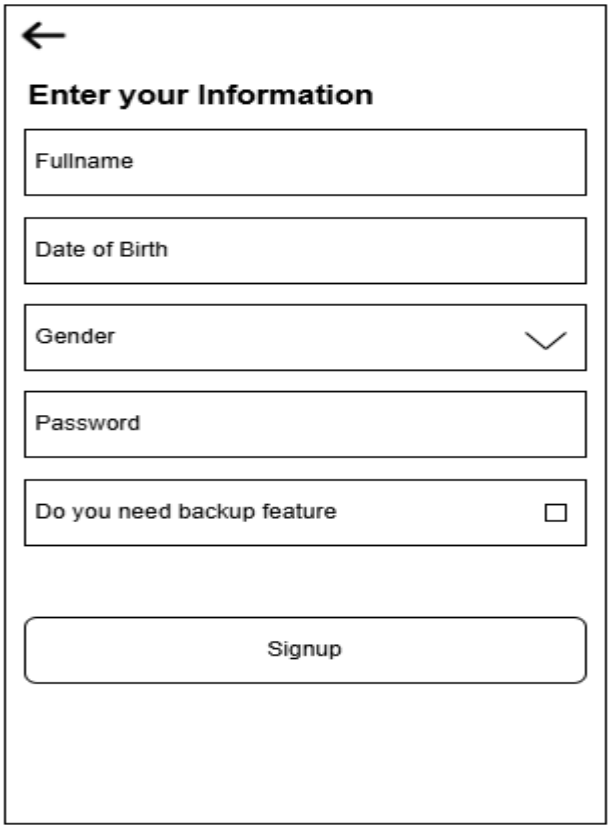
Table name: meals

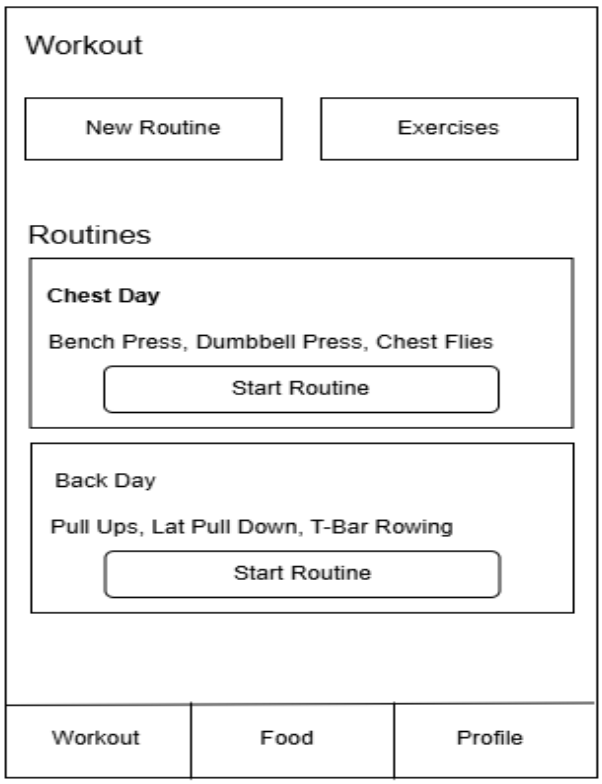
Purpose: To store the meals eaten by the user

S. No.	Field Name	Data Type	Constraint	Description
1	meal_id	Integer	Primary Key	Food unique id
2	food_id	Integer	Foreign Key, On Delete Cascade	Food name
3	meal_date	Date	Not Null	Meal eaten date
3	meal_type	Text	Not Null, Check	Breakfast, lunch, dinner or snacks
4	meal_quantity	Real	Not Null	Quantity of meal taken
5	meal_calories	Real	Not Null	Food calories
6	meal_protein	Real	Not Null	Food protein
7	meal_carbs	Real	Not Null	Food carbs
8	meal_fats	Real	Not Null	Food fats
9	meal_fibers	Real	Not Null	Food Fibers
10	updated_at	Integer	Not Null	Record updated timestamp

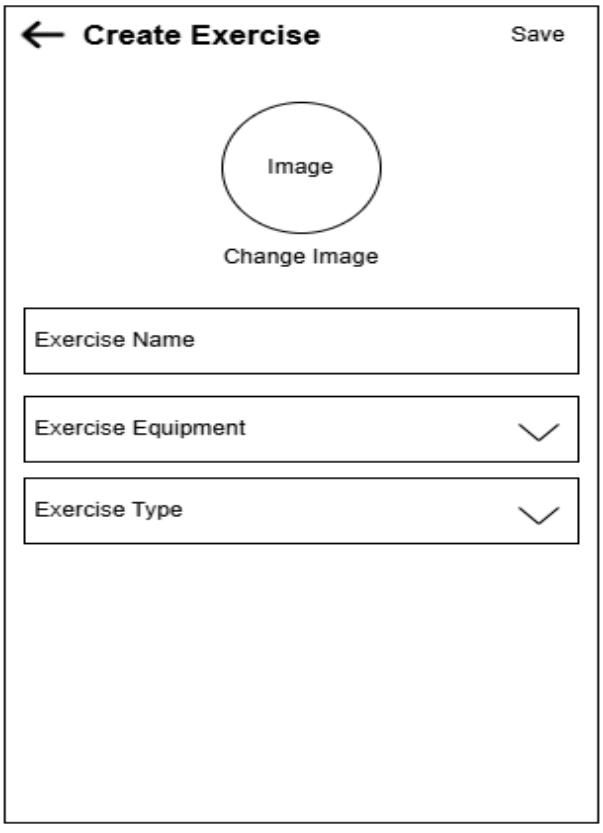
4.4 USER INTERFACE DESIGN

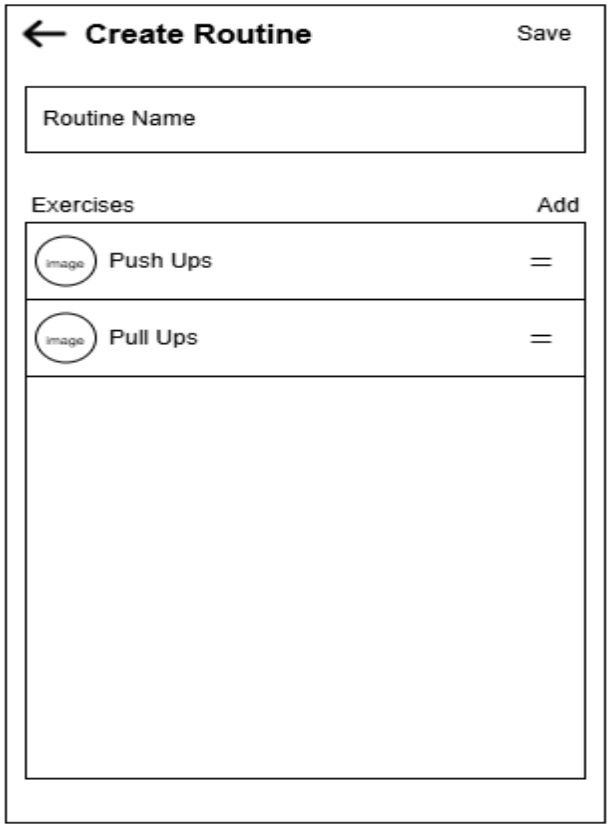


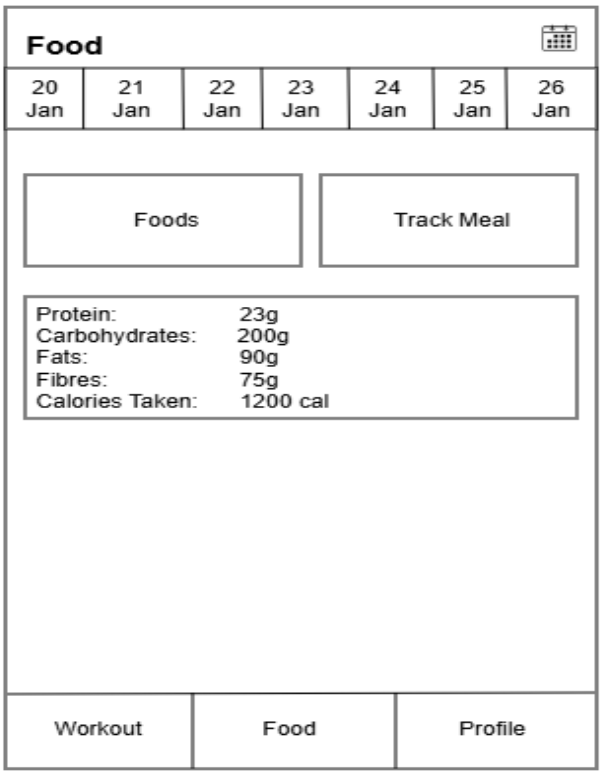



Workout
New Routine
Exercises
Routines
Chest Day
Bench Press, Dumbbell Press, Chest Flies
Start Routine
Back Day
Pull Ups, Lat Pull Down, T-Bar Rowing
Start Routine
Workout
Food
Profile

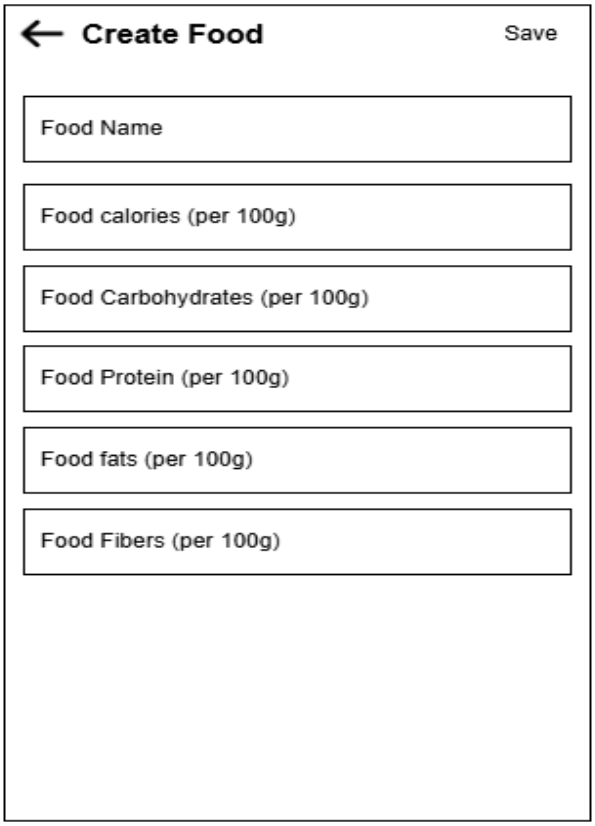
Exercises
Create
Search
Image
Push Ups
Image
Pull Ups

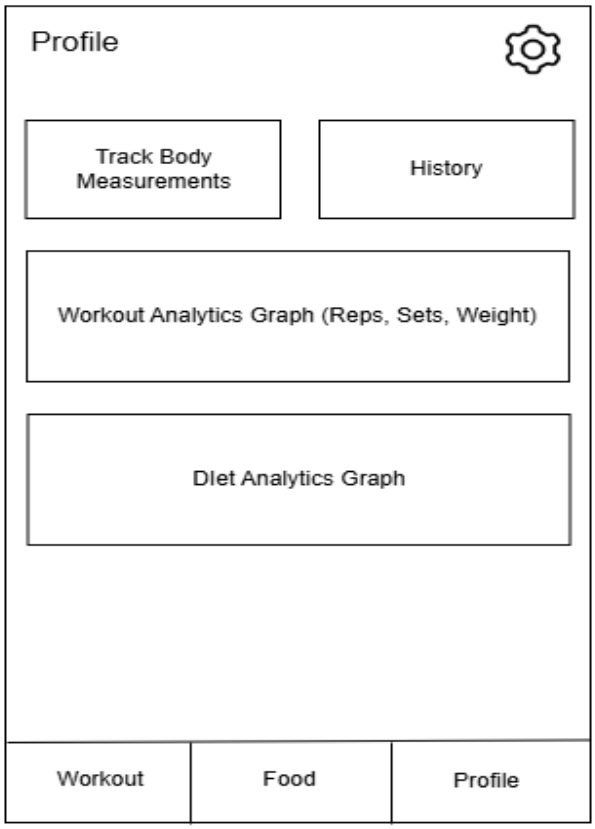

Create Exercise
Save
Image
Change Image
Exercise Name
Exercise Equipment
Exercise Type


Create Routine
Save
Routine Name
Exercises
Add
Push Ups
Pull Ups


Food
20 Jan
21 Jan
22 Jan
23 Jan
24 Jan
25 Jan
26 Jan
Foods
Track Meal
Protein: 23g
Carbohydrates: 200g
Fats: 90g
Fibres: 75g
Calories Taken: 1200 cal
Workout
Food
Profile

Foods
Create
Search
Idly
Pongal


Create Food
Save
Food Name
Food calories (per 100g)
Food Carbohydrates (per 100g)
Food Protein (per 100g)
Food fats (per 100g)
Food Fibers (per 100g)


Profile
Track Body Measurements
History
Workout Analytics Graph (Reps, Sets, Weight)
Diet Analytics Graph
Workout
Food
Profile

4.5 NORMALIZATION

Normalization is the process of organizing the data in the database. Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization divides the larger table into the smaller table and links them using relationship. The normal form is used to reduce redundancy from the database table.

First Normal Form (1NF)

For a table to be in the First Normal Form, it should follow the following 4 rules:

- It should only have single(atomic) valued attributes/columns.
- Values stored in a column should be of the same domain.
- All the columns in a table should have unique names.
- And the order in which data is stored, does not matter.

Second Normal Form (2NF)

For a table to be in the Second Normal Form,

- It should be in the First Normal form.
- And, it should not have Partial Dependency. Partial Dependency occurs when a non-prime attribute is functionally dependent on part of a candidate key.

Third Normal Form (3NF)

A table is said to be in the Third Normal Form when,

- It is in the Second Normal form.
- And, it doesn't have Transitive Dependency.

Boyce and Codd Normal Form (BCNF)

Boyce and Codd Normal Form is a higher version of the Third Normal form. This form deals with certain type of anomaly that is not handled by 3NF. A 3NF table which does not have multiple overlapping candidate keys is said to be in BCNF. For a table to be in BCNF, following conditions must be satisfied:

- R must be in 3rd Normal Form
- For each functional dependency (X → Y), X should be a super Key.

Fourth Normal Form (4NF)

A table is said to be in the Fourth Normal Form when,

- It is in the Boyce-Codd Normal Form.
- And, it doesn't have Multi-Valued Dependency.