FITFLEX

Project Documentation

1.INTRODUCTION~~:~~

Project Title: FITFLEX-Fitness partner

* Team member: ELAKKIYA S
* Team member: GEETHA R
* Team member: GOWRI R
* Team member: LAKSHMI M 
* Team member: LAVANYA R

# 2. Project Overview

* Name: FitFlex NM
* Purpose: To provide users with personalized fitness and nutrition management (“NM” likely stands for Nutrition & Management / or maybe “New Module” etc.)
* Target Users: People wanting to track workouts, lose or maintain weight, gain muscle, plan meals, etc.
* Main Features
  1. Daily/weekly workout plans tailored by user goals (weight loss, muscle gain, general fitness)
  2. Nutrition / diet recommendation system
  3. Progress tracking (workouts completed, calories consumed / burned, weight changes, etc.)
  4. User authentication, profiles, perhaps different subscription plans
  5. Possibly cross location or schedule flexibility for workouts (if that’s part of “FitFlex”)

# 3. Architecture & Technology Stack

* Frontend: React.js (or similar), or mobile app if applicable
* Backend: Node.js + Express, or Python / Flask / Django (if diet recommendation API, etc.)
* Database: MongoDB or relational DB depending on user / plan model
* APIs: For workout data, nutrition info, perhaps third party integration (food databases, etc.)
* Hosting / Deployment: e.g. cloud servers, Docker, maybe Netlify / Vercel / AWS etc.

# 4. System Design

* Components and Modules
  + User Module: Registration, authentication, profile management o Workout Module: Plan creation, schedule, tracking
  + Nutrition Module: Meal / diet plans, calorie tracking, food database o Progress Module: Charts, metrics
  + Admin / Subscription Module: Manage subscriptions, offers, pricing  Data Models / Database Schema: Users, Plans, Meals, Exercises, Logs etc.
* API endpoints: e.g.
* POST /signup
* POST /login
* GET /workouts
* POST /workouts/log
* GET /meals /recommendations
* GET /progress
* Security: JWT tokens, password hashing, validation, data privacy

5

.

U

X

/

U

I

D

e

s

i

g

n

* Wireframes or mockups of major screens/pages: Signup, Dashboard, Workout plan, Meal planner, Progress graphs  Navigation structure

# 6. Project Structure (Code Organization)

* Folder / file organization
* Key dependencies / modules
* How to run locally, prerequisites

7

.

C

h

a

l

l

e

n

g

e

s

&

S

o

l

u

t

i

o

n

s

* Any tricky parts (e.g. matching diet plans to user preferences / allergies, integrating third party APIs, handling offline data or state, etc.)  How they were / will be solved

# 8. Testing & Quality Assurance

* Unit tests, integration tests
* Edge cases (e.g. invalid input, network failures)
* Performance (e.g. for large food/exercise databases)

9

.

D

e

p

l

o

y

m

e

n

t

&

M

a

i

n

t

e

n

a

n

c

e

CI/CD setup

* Versioning
* Monitoring & logs
* Updates / maintenance plan

1

0

.

R

o

a

d

m

a

p

&

F

u

t

u

r

e

E

n

h

a

n

c

e

m

e

n

t

s

* Features planned in future (e.g. social sharing, AI based coaching, custom meal upload, wearable integration)
* Scalability improvements

1

1

.

D

o

c

u

m

e

n

t

a

t

i

o

n

&

R

e

s

o

u

r

c

e

s

* API documentation (endpoints, request/response formats)
* User manual / onboarding materials
* Developer guide (how to contribute, setup, coding standards)

# 1. Project Overview

* Name: FitFlex NM
* Purpose: To provide users with personalized fitness and nutrition management (“NM” likely stands for Nutrition & Management / or maybe “New Module” etc.)
* Target Users: People wanting to track workouts, lose or maintain weight, gain muscle, plan meals, etc.
* Main Features
  1. Daily/weekly workout plans tailored by user goals (weight loss, muscle gain, general fitness)
  2. Nutrition / diet recommendation system
  3. Progress tracking (workouts completed, calories consumed / burned, weight changes, etc.)
  4. User authentication, profiles, perhaps different subscription plans
  5. Possibly cross location or schedule flexibility for workouts (if that’s part of “FitFlex”)

# 2. Architecture & Technology Stack

* Frontend: React.js (or similar), or mobile app if applicable
* Backend: Node.js + Express, or Python / Flask / Django (if diet recommendation API, etc.)
* Database: MongoDB or relational DB depending on user / plan model
* APIs: For workout data, nutrition info, perhaps third party integration (food databases, etc.)

H

o

s

t

i

n

g

/

D

e

p

l

o

y

m

e

n

t

:

e

.

g

.

c

l

o

u

d

s

e

r

v

e

r

s

,

D

o

c

k

e

r

,

m

a

y

b

e

N

e

t

l

i

f

y

/

V

e

r

c

e

l

/

A

W

S

e

t

c

.

3

.

S

y

s

t

e

m

D

e

s

i

g

n

* Components and Modules
  + User Module: Registration, authentication, profile management o Workout Module: Plan creation, schedule, tracking
  + Nutrition Module: Meal / diet plans, calorie tracking, food database o Progress Module: Charts, metrics
  + Admin / Subscription Module: Manage subscriptions, offers, pricing  Data Models / Database Schema: Users, Plans, Meals, Exercises, Logs etc.
* API endpoints: e.g.
* POST /signup
* POST /login
* GET /workouts
* POST /workouts/log
* GET /meals /recommendations
* GET /progress
* Security: JWT tokens, password hashing, validation, data privacy

4

.

U

X

/

U

I

D

e

s

i

g

n

* Wireframes or mockups of major screens/pages: Signup, Dashboard, Workout plan, Meal planner, Progress graphs  Navigation structure

# 5. Project Structure (Code Organization)

* Folder / file organization
* Key dependencies / modules
* How to run locally, prerequisites

6

.

C

h

a

l

l

e

n

g

e

s

&

S

o

l

u

t

i

o

n

s

* Any tricky parts (e.g. matching diet plans to user preferences / allergies, integrating third party APIs, handling offline data or state, etc.)  How they were / will be solved

# 7. Testing & Quality Assurance

* Unit tests, integration tests
* Edge cases (e.g. invalid input, network failures)

Performance (e.g. for large food/exercise databases)

8

.

D

e

p

l

o

y

m

e

n

t

&

M

a

i

n

t

e

n

a

n

c

e

* CI/CD setup
* Versioning
* Monitoring & logs
* Updates / maintenance plan

9

.

R

o

a

d

m

a

p

&

F

u

t

u

r

e

E

n

h

a

n

c

e

m

e

n

t

s

* Features planned in future (e.g. social sharing, AI based coaching, custom meal upload, wearable integration)
* Scalability improvements

1

0

.

D

o

c

u

m

e

n

t

a

t

i

o

n

&

R

e

s

o

u

r

c

e

s

* API documentation (endpoints, request/response formats)
* User manual / onboarding materials
* Developer guide (how to contribute, setup, coding standards)