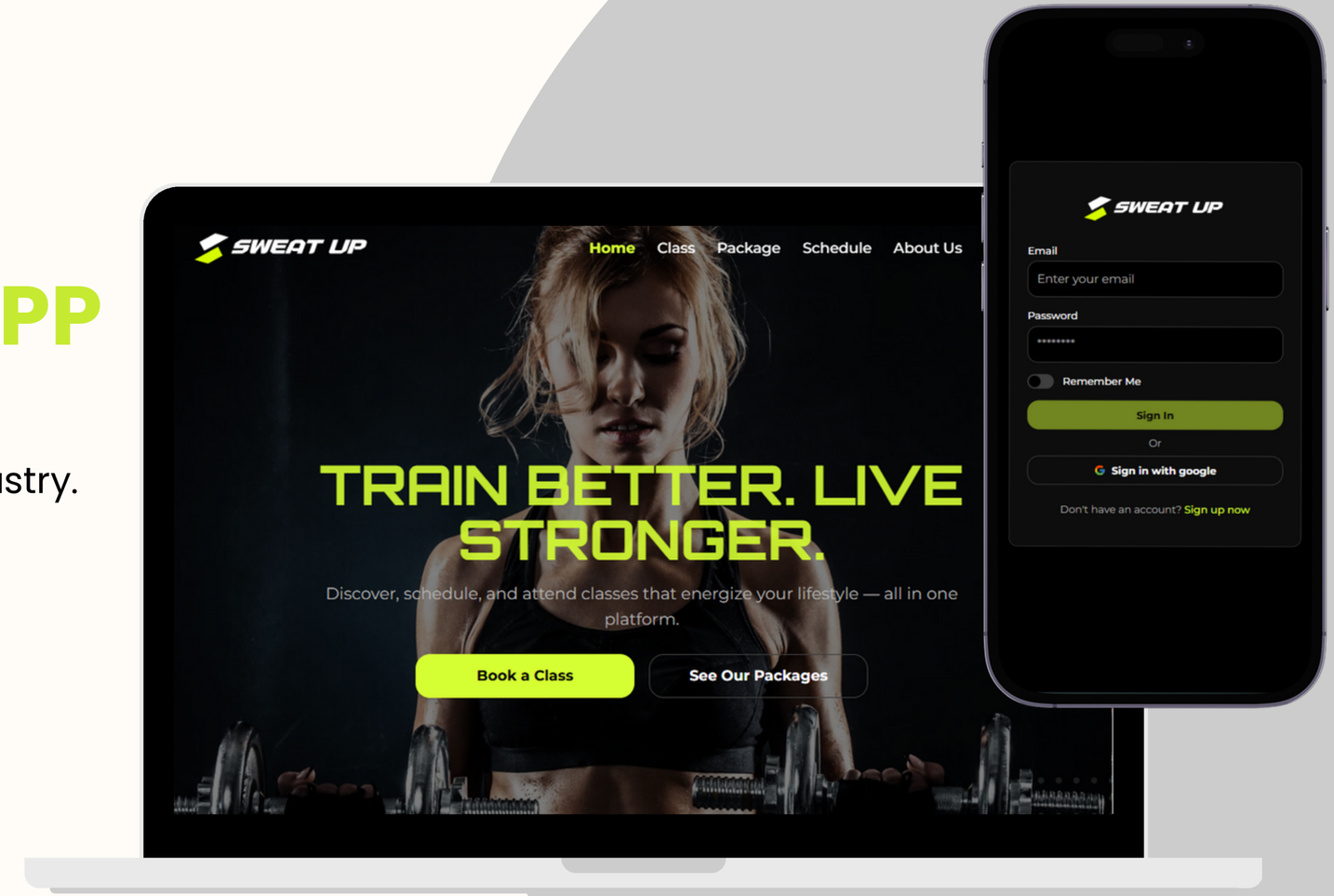
SWEAT UP

FITNESS BOOKING APP

A capstone project built with precision to streamline user experience in the fitness industry.



AHMAD FIQRI OEMRY

DIBIMBING CAPSTONE

About Me

Freelance Web Developer with 5+ years of experience in quality control and inspection in the engineering industry. I bring strong analytical thinking, attention to detail, and structured problem-solving into the field of software development. I specialize in building scalable fullstack web applications with clean architecture, modular codebase, and production-ready infrastructure.

Education

2025
Dibimbing.id
Backend Developer - Golang

2024
Purwadhika
Data Science & Machine Learning

2011 - 2016
UNIVERSITAS SUMATERA UTARA
Bachelor Degree - Mechnical Eng.
GPA 3.09

Experience

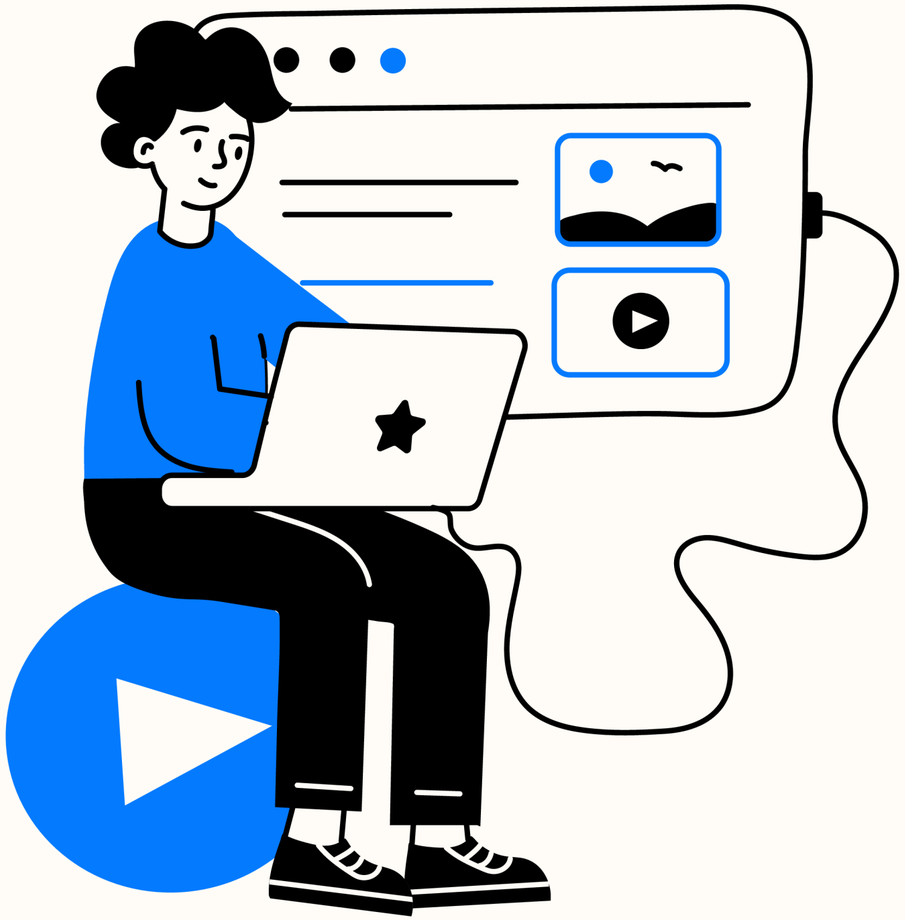
2019 - 2022
PT. SUCOFINDO PERSERO
Technical Inspector

2022 - 2023
PT. REZEKI ALAM INDONESIA
Site Controller

2022 - 2023
PT. RADIANT UTAMA INTERINSCO
Quality Inspector

2024 - NOW
FREELANCE
Web Developer

Table Of Content



01. OVERVIEW

02. PROBLEM STATEMENT

03. PROPOSED SOLUTION

04. CORE FEATURE

05. TECHNOLOGY STACK

06. TIMELINE

07. DESIGN

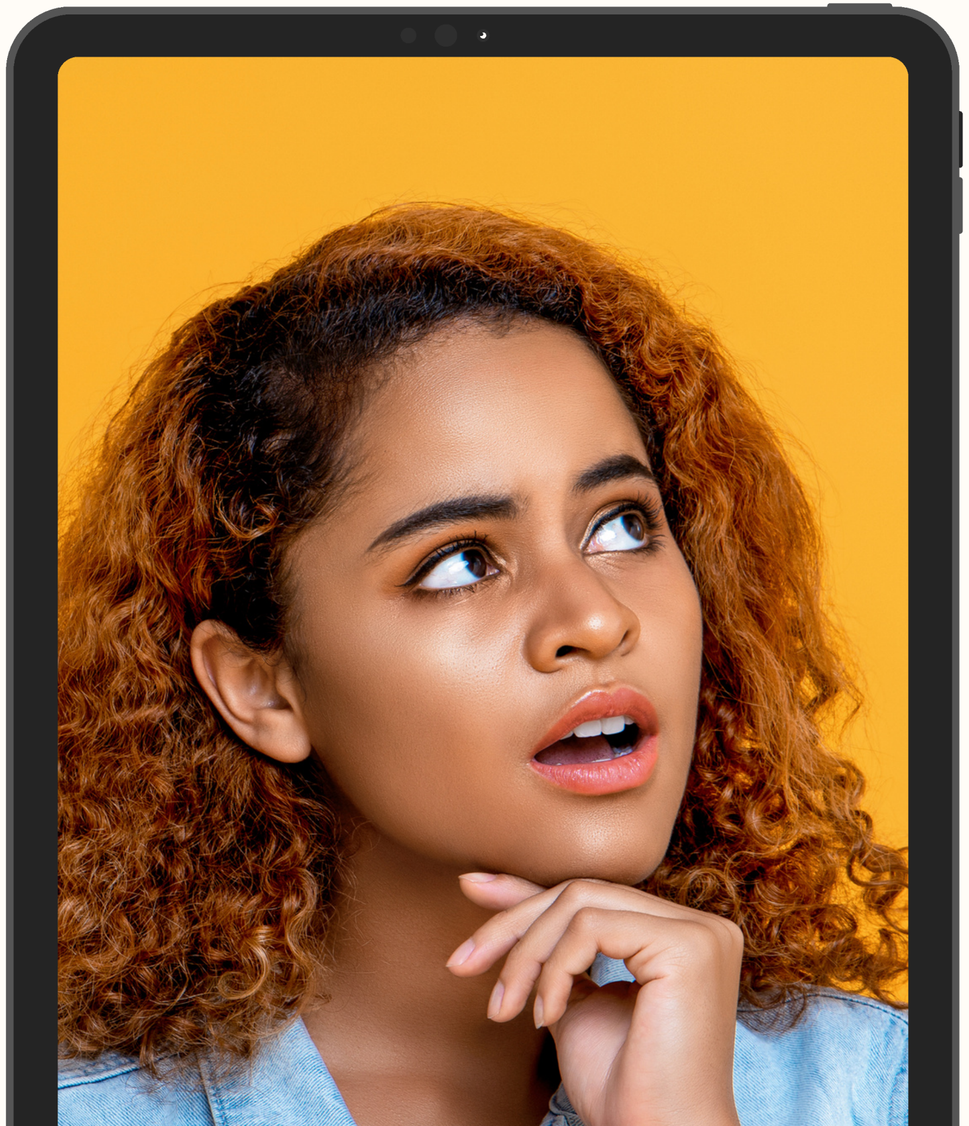
08. API ENDPOINTS

09. DEVELOPMENT CHALLANGES

10. CONCLUSION

01. OVERVIEW

What is Sweat up ?

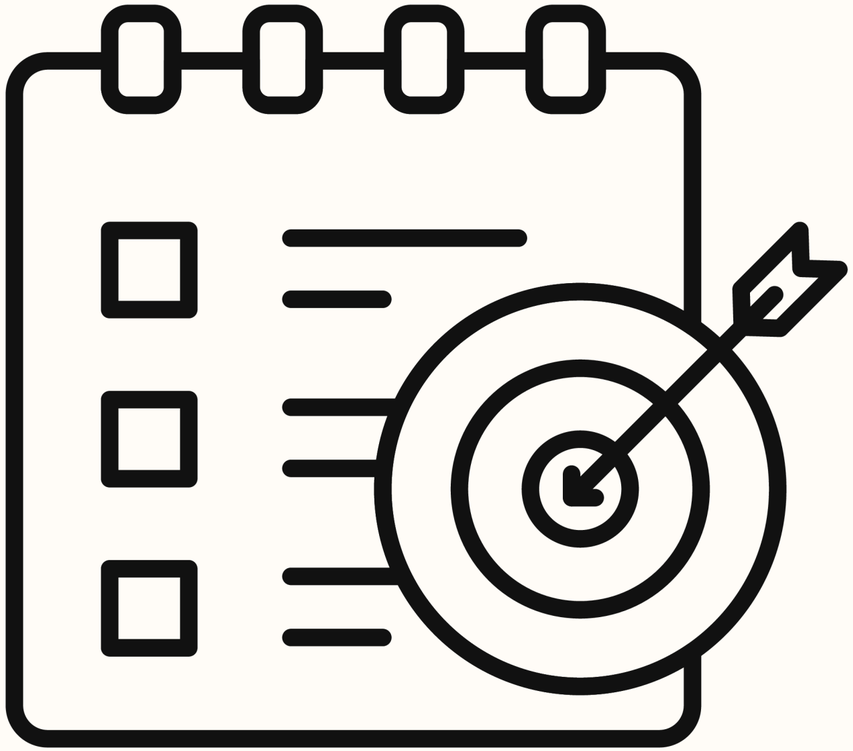


SweatUp adalah sebuah aplikasi booking kebugaran berbasis web yang dirancang untuk mempermudah pengguna dalam mencari, memesan, dan menghadiri kelas kebugaran secara terjadwal.

Platform ini menghubungkan pengguna, instruktur, dan manajemen studio dalam satu sistem terintegrasi yang mencakup fitur otentikasi, pemesanan, pembayaran, hingga absensi.

01. OVERVIEW

Goals & Objective



✅ Simplify the booking process for users with real-time availability and class filters.

✅ Automate subscription and payment management securely.

✅ Enable personalized user experience through notifications, attendance history, and feedback integration.

✅ Provide operational efficiency for fitness studios and instructors.

02. PROBLEM STATEMENT

Challenges in Fitness Management

- ❌ Manual booking is time-consuming and error-prone
- ❌ Limited visibility on class schedules and availability
- ❌ Lack of real-time confirmation & reminders
- ❌ Difficulty in tracking user attendance & subscriptions
- ❌ Fragmented systems between payment, scheduling, and communication

02. PROBLEM STATEMENT

Why this App Matters

🔄 Digitizing and automating the entire booking lifecycle.

📅 Offering real-time, user-friendly scheduling tools

💳 Integrating secure and seamless payment workflows.

📲 Enabling instant notifications and smart attendance tracking.

📊 Helping fitness businesses scale without operational bottlenecks.

Keys Benefit

⚙️ Centralized system for user, class, and instructor management

🔒 Secure and scalable architecture built for high-traffic environments

⏱️ Real-time operations including booking confirmation and live class status

🧠 Intelligent automation for reminders, attendance, and subscription renewals

🔄 API-first design to support mobile apps and frontend flexibility

04. CORE FEATURES

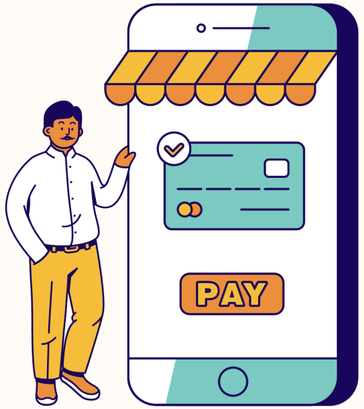
User Authentication & Role management

- Secure login via Email, OTP, and Google OAuth
- Role-based access: Customer, Instructor, Admin
- Token-based session (JWT + Refresh Token)

Class Scheduling & Package Management

- One-time & recurring class scheduling
- Auto-generated schedule templates
- Class filtering by category, level, type, and location
- Package system with quota and class validation

04. CORE FEATURES



Package Purchasement & Class Booking

- Seamless package purchase with integrated payment (Stripe)
- Auto-check class eligibility & package validity
- Booking with check-in/out and user history
- Auto-cancellation & real-time capacity check

Notifications & Review System

- Class reminders, promotions, and updates via notification
- User-controlled notification settings
- Automated review request post-attendance
- Reviews affect instructor rating and class visibility

05. TECHLONOGY STACK

Backend Development

CLOUDINARY

MYSQL

REDIS.IO

STRIPE PAYMENT

GO (GIN)

GOOGLE OATUH

NODEMAILER

05. TECHLONOGY STACK

Frontend Development

TANSTACK-QUERY

GOOGLE OAUTH

REACT.JS

MOTIONS

ZUSTAND

VITE

TAILWIND CSS

TYPESCRIPT

05. TECHLONOGY STACK

Deployment & Supporting Tools

HOSTINGER

GIT ACTIONS

NOTIONS

NGINX

NGROK

VS CODE

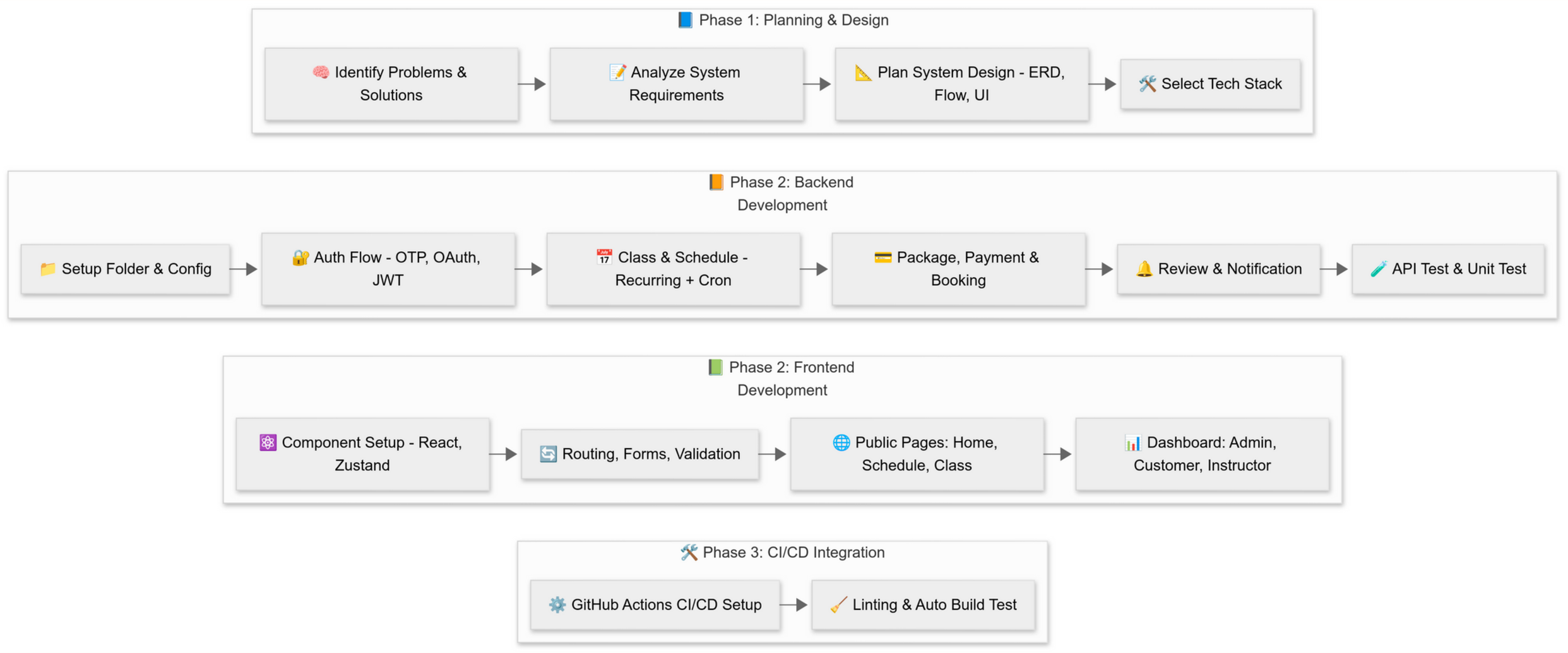
POSTMAN

CHAT GPT

EXCALIDRAW

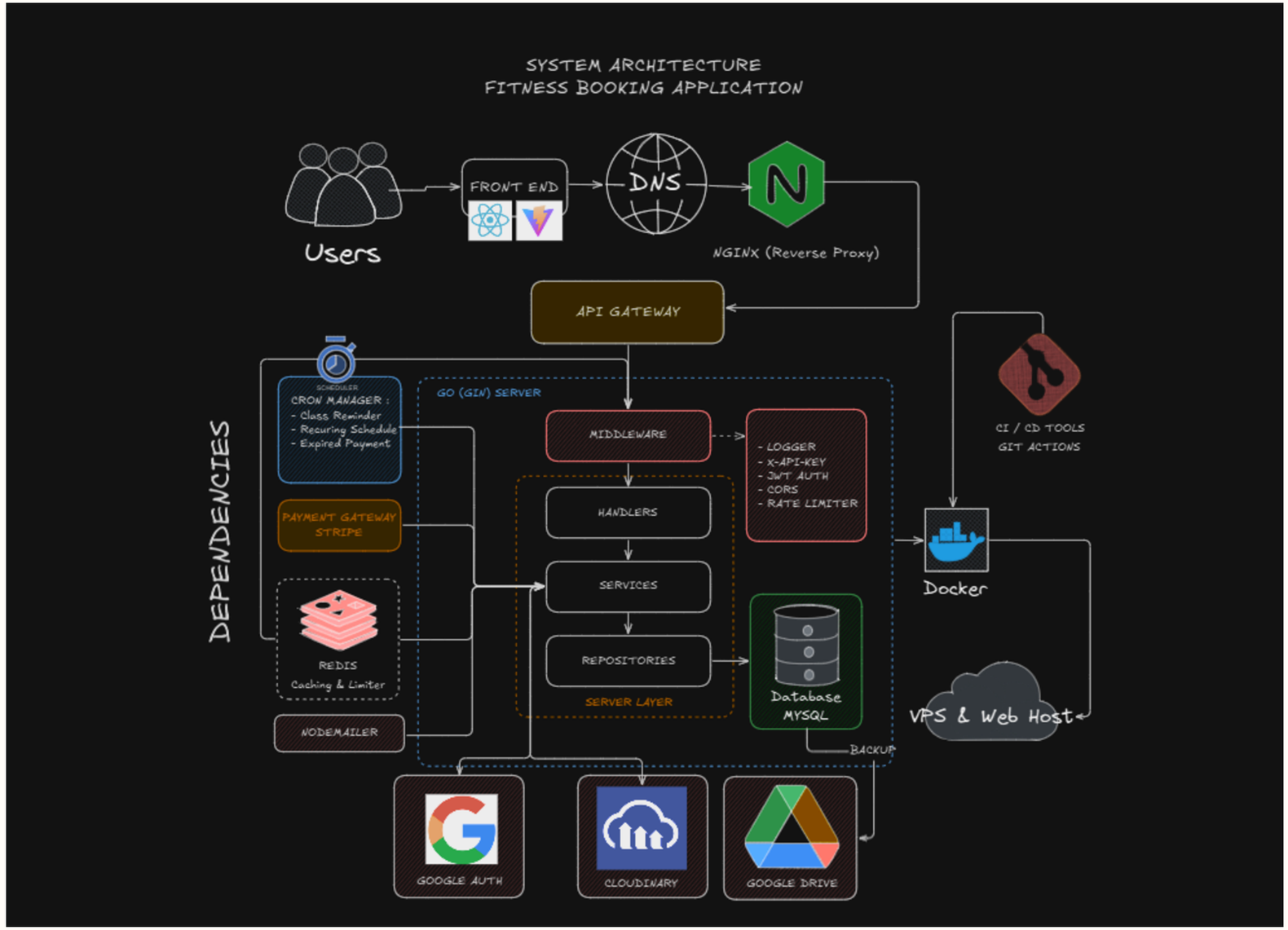
DBDIAGRAM.IO

06. PROJECT TIMELINE



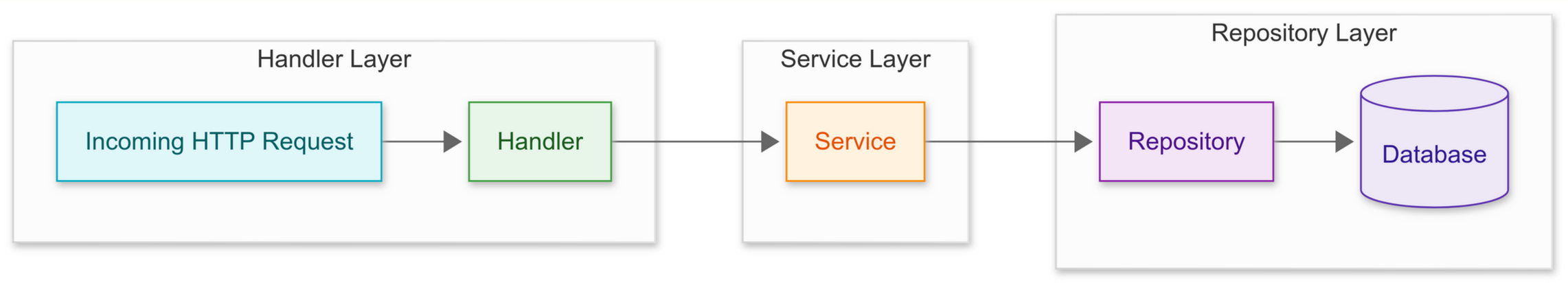
07. DESIGN

System Architecture



07. DESIGN

Service & Layer Breakdown



Handler Layer

- Handles HTTP Request and Response
- Delegates Logic to services
- Perform Input Validation & Error Handling

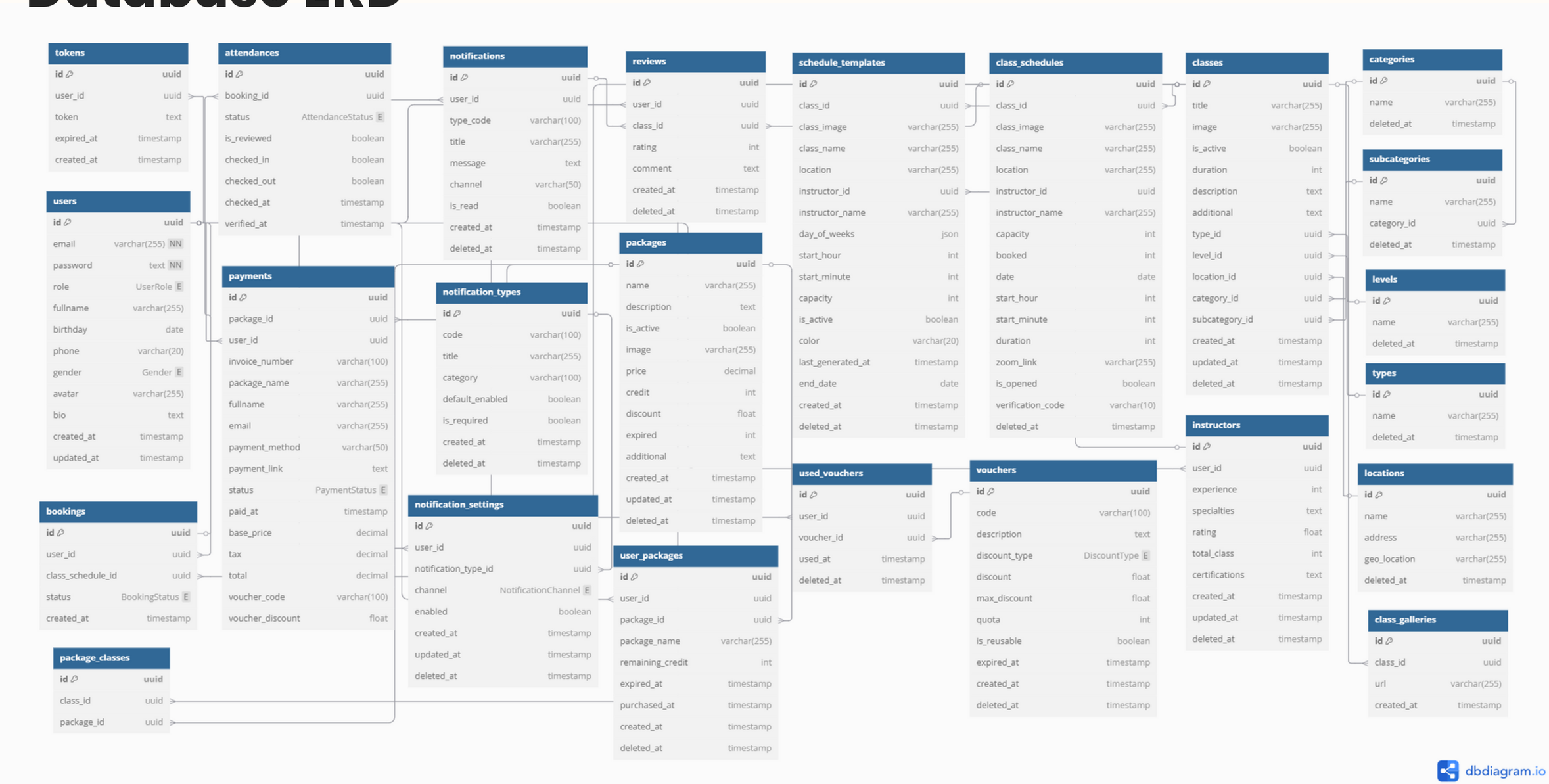
Service Layer

- Encapsulate Business Logic
- Acts As an Intermediatery
- Orchestration Multiple Operation

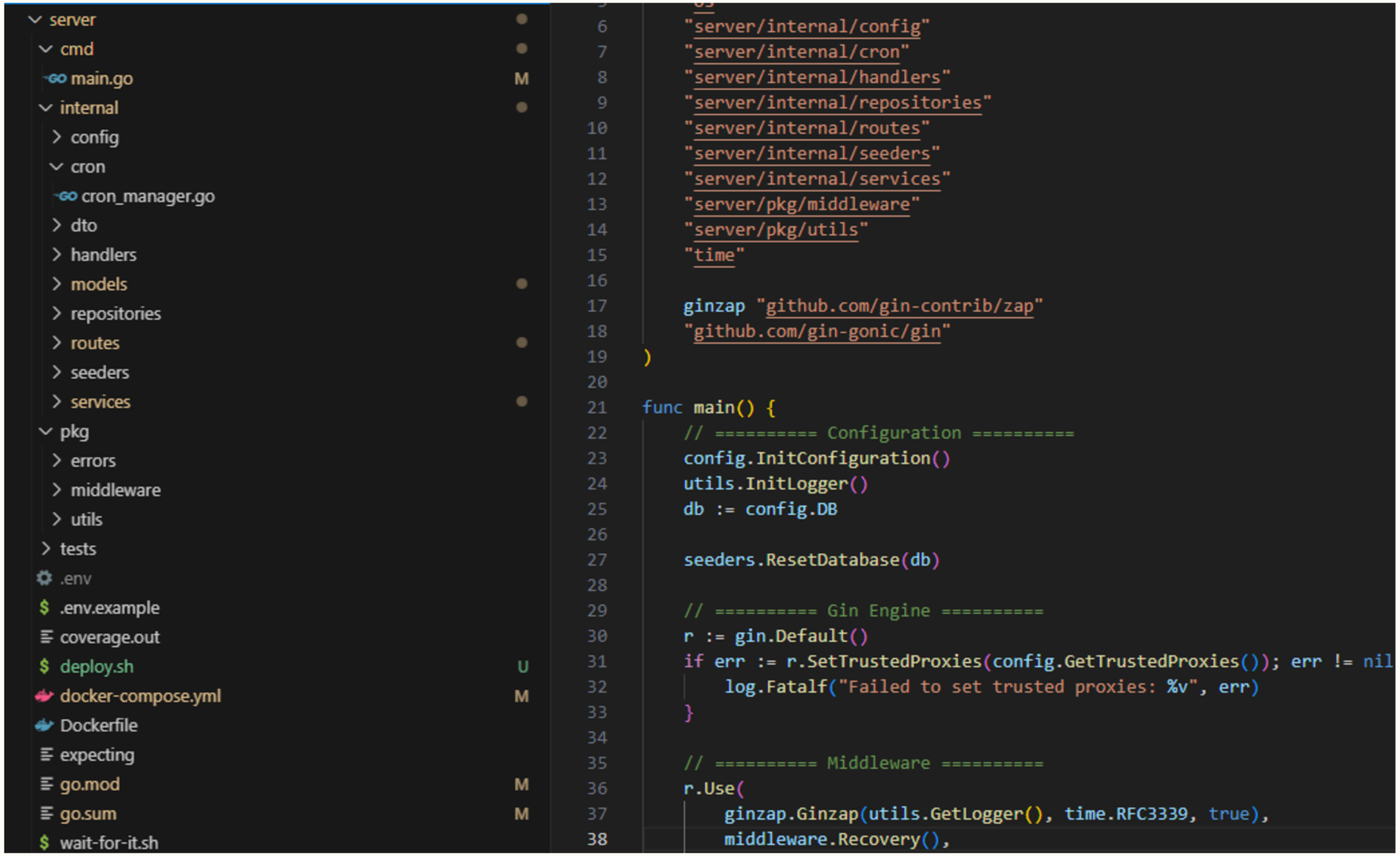
Repository Layer

- Manages Data Access to DB
- CRUD Operations

Database ERD



Backend Folder Structure



08. API ENDPOINT

User & Autentication

Method	Endpoint	Description
POST	/auth/login	Login with email & password
POST	/auth/logout	Logout and clear session
POST	/auth/register	Register new account (email/OTP)
POST	/auth/send-otp	Send OTP to email for verification
POST	/verify-otp	Verify OTP and complete register
POST	/auth/refresh-token	Refresh JWT token
GET	/auth/google	Redirect to Google OAuth login
GET	/auth/google/callback	Handle Google OAuth response
GET	/auth/me	Get current user profile

Method	Endpoint	Description
GET	/users/me	Get profile
PUT	/users/me	Update profile
PUT	/users/me/avatar	Update avatar
GET	/admin/users	Get all users (admin)
GET	/admin/users/:id	Get user detail (admin)
GET	/admin/users/stats	Get user statistics (admin)

08. API ENDPOINT

Class Schedule & Package Management

Method	Endpoint	Description
GET	/packages	Get all packages
GET	/packages/:id	Get package detail
POST	/admin/packages	Create new package
PUT	/admin/packages/:id	Update package
DELETE	/admin/packages/:id	Delete package

Method	Endpoint	Description
GET	/classes	Get all classes
GET	/classes/:id	Get class detail
POST	/admin/classes	Create new class
PUT	/admin/classes/:id	Update class
DELETE	/admin/classes/:id	Delete class
POST	/admin/classes/:id/gallery	Upload class gallery

Method	Endpoint	Description
GET	/schedules	Get all class schedules
GET	/schedules/:id	Get schedule detail
GET	/schedules/status	Get user booking status
GET	/instructor/schedules	Get instructor schedules
GET	/instructor/schedules/:id/attend	Get class attendances
PATCH	/instructor/schedules/:id/open	Open class for check-in
POST	/admin/schedules	Create class schedule (admin)
POST	/admin/schedules/recurring	Create recurring schedule (admin)
PUT	/admin/schedules/:id	Update schedule (admin)
DELETE	/admin/schedules/:id	Delete schedule (admin)

08. API ENDPOINT

Package Purchasement & Booking System

Method	Endpoint	Description
POST	/bookings	Create new booking
GET	bookings	Get list of user's bookings
GET	/bookings/:id	Get detail of a specific booking
POST	/bookings/:id/check-in	Check-in to a booked class
POST	/bookings/:id/check-out	Check-out from class

Method	Endpoint	Description
POST	/bookings	Create booking (customer)
GET	/bookings	Get user bookings (customer)
GET	/bookings/:id	Get booking detail
POST	/bookings/:id/check-in	Check-in to class
POST	/bookings/:id/check-out	Check-out from class

Method	Endpoint	Description
GET	/user-packages	Get user packages
GET	/user-packages/class/:id	Get packages valid for class

Method	Endpoint	Description
POST	/payments	Create payment (customer)
GET	/payments/me	Get user payment history
GET	/payments/me/:id	Get payment detail
POST	/payments/stripe/notification	Stripe webhook
GET	/admin/payments	Get all payments (admin)
GET	/admin/payments/:id	Get payment detail (admin)

08. API ENDPOINT

Notifications & Review System

Method	Endpoint	Description
GET	/reviews/:classId	Get reviews by class
POST	/reviews/:id	Submit review from booking

Method	Endpoint	Description
GET	/notifications	Get user notifications
PATCH	/notifications/read	Mark all as read
GET	/notifications/settings	Get notification settings
PUT	/notifications/settings	Update notification settings
POST	/admin/notifications/broadcast	Send broadcast (admin)

09. DEPLOYMENT CHALLANGES

Obstacles

- Complex recurring schedule logic
- Class-package eligibility validation
- Real-time notification accuracy
- Role-based access enforcement
- Sync issues between payment & booking

Solutions

- ron + validation layer for schedule
- Booking middleware for package checks
- Event-driven notification system
- Policy-based role access
- Webhook-first payment handling

10. CONCLUSION

Project Impact

- Simplified booking operations
- Reduced manual effort
- Secure & scalable backend architecture

Future Improvements

- Mobile app integration
- Real-time instructor chat
- AI-based class recommendations

THANK YOU !!!

Know More About project

Live Preview : https://sweatup.ahmadfiqrioemry.com

Postman Collection : https://test99-3397.postman.co/workspace/fitness_booking_app~129e8344-0751-4fda-9364-204b522c762f/collection/39450267-e723e170-c727-46d8-a620-7de9fd77ebff?action=share&source=copy-link&creator=39450267

Project Documentation : https://www.notion.so/FITNESS-BOOKING-APPLICATION-206f51e9c94d80948a42cc8c89ba98c0

Source code : https://github.com/fiqrioemry/fitness_booking_application

AHMAD FIQRI OEMRY

DIBIMBING CAPSTONE