Django-Based Web API System | Personal Study - Learning

* Built Back-End API using Django to control database operations management
* Evaluated REST API using Insomnia REST client for performance and accuracy.
* Implemented user authentication system with unit testing for reliability

Linux-Based OS Concepts - Design and Implementation | University Study

* Designed and implemented core operating system functionalities using C++.
* Applied paging and memory replacement algorithms for data transfer integrity.
* Wrote a multithreaded CPU scheduler for synchronizing multiple IO devices.
* Built File System with pathname resolution, symbolic linking, nested directories, memory allocation of data blocks, and secure handling of name collision.

Restaurant UI/UX Design and Development using React | Personal Study - Learning

* Designed user-centric UX/UI design from wireframe to prototype using Figma
* Built a responsive web app using React to display menus and handle reservations
* Performed unit testing with Jest, validating functionality and user interactions.

# Anarep Project

## Zoomed Out

Full Stack Fitness App (Expo, .Net9, MySQL, AWS, BLE IMU) | Personal Study – Project

A connected fitness app with real-time sensor streaming, rich UI, user-generated content , and social features. Built using Expo and .NET 9 with a full backend supported by MySQL Database.

App Development via Expo (React Native) using TypeScript and React | Core Features

* JWT Authentication System within Expo - to allow for stateless authentication and authorization of REST operations.
* Responsive & Adaptive UI System for Cross Platform Consistency – applied industry practices for global supplied screen dimensions and percentage-based sizing to scale components dynamically across platforms within components utilized UI library tokens to scale text.
* UI & UX Design – framed early designs via Figma to outline structured. Then implemented tabbed-based architecture to provide clear visual navigation throughout the app. Pages included Entry, Sign-In, and User Tabs(Home, Community, Workout, Planning, Settings, these include more pages as the user moves through activities)
* Scalable State Management Architecture – states across the application required a depth of global and local management to ensure consistency and performance optimization such that I developed a layered context system to comply with data requirements.
* Bluetooth Device Management API for Real Time Sensor Data – Designed platform specific connection classes managed in contexts to integrated live data feeds from the IMU for analytics.
* Back-End Integration – using libraries axios, react query, rxjs to connect and interact with API throughout the application to update user content and fetch application assets.
* Pop-Up and Forms – using libraries formik and yup I outlined series of forms for the user to interact with usually outlining these forms on pop up slides via Modal supplied by react native.

Back-End Development via .Net9 (Asp.Net Core) using C# | Core Features

* Scalable MySQL Database Architecture – utilizing .Net Core Entity Framework I created relational data models to create tables for authentication, user profiles, workout tracking, workout planning while maintaining the ability to migrate to new platforms as demands change.
* REST Framework and API endpoints – implemented MVC Framework to create a REST API including creating multiple controllers to handle CRUD operations categorized on activities conducted within the app. Different operations required DTO models for structuring and ensure validation.
* Documentation- Integrated Swagger Middleware to document API endpoints and provide simplified developer testing.

IMU Sensor Integration and Sensor Fusion via WitMotion WT901DCL

* Before implementing the Senor Data within the app it was essential to map out required techniques and example libraries to implement sensor fusion to derive relative motion tracker using IMU
* Preformed Sensor Calibration in python environment using Witmotion SDK and bleak for connectivity. Then used pyfilter and imufusion to preform Sensor Fusion operations.
* Developed Calibration Environment by using Animatics m5 Smart Motor on Linear Actuator to get consistent refrence motion track.
* Reversed engineered required python operations in typescript for native use on Expo App.

Animatics Smart Motor Python Library and VsCode extension

* Animatics Smart Motors Firmware is written in low level language built upon C++ named Smart Motor Executable .smx which readable as .smi .smx as limited variable names no code modularity, limited subroutines are limited to numbers and code is ran iteratively required manually breaking to be in tune with motor.   
  This has a sharp learning curve and get be difficult to debug and manage from micro controller. I wanted to develop a system that would build upon the existing structure of the
* I was developing a sub language which could be compiled in smx files with a more readable and familiar python structure. The python library would then compile these files returning meta data on smx file uploaded to the motor mapping subroutines and variables to readable python counter parts
* BREAK
* Developed python library for modular Smart Motor SubRoutines, Transpiled Firmware language compiler, and Remote Runtime Management tools.

Django-Based Web API System | Personal Study

* Designed and implemented Back-End API using Django to control management of database operations and exposure of RESTful API endpoints.
* Evaluated API using the Insomnia REST client, validating endpoints for performance and accuracy.
* Built in user authentication/registration system and conducted unit testing to ensure reliability and functionality.

Database Management E-Commerce Application in Flask | University Study

* Designed a normalized relational database schema to eliminate redundancy and ensure data integrity, tailored for an e-commerce platform.
* Implemented a MySQL database from a normalized schema and integrated it into a Flask application to enable dynamic RESTful API functionality.
* Developed a functional web interface using Flask to visualize and interact with data, including dynamic charting and query execution.

Restaurant UI/UX Design and Development using React | Personal Study

* Designed user-centric UX/UI design from wireframe to prototype using Figma following industry best practices and standards.
* Built a responsive, multi-page React web application using reusable components to display menu details and handle reservation forms efficiently.
* Ensured reliability and functionality through unit testing with Jest, validating component behavior and user interactions.

Django-Based Web API System | Personal Study

* Designed and implemented Back-End API using Django to control management of database operations and exposure of RESTful API endpoints.
* Evaluated API using the Insomnia REST client, validating endpoints for performance and accuracy.
* Built in user authentication/registration system and conducted unit testing to ensure reliability and functionality.

Linux-Based OS Concepts - Design and Implementation | University Study

* Designed and implemented core operating system functionalities using C++.
* Applied Paging and Memory Replacement Algorithms to ensure data transfer and integrity.
* Developed a thread scheduler program with different CPU scheduling policies for synchronization of operations between simulated IO devices.
* Implemented File System consisting of pathname resolution, symbolic linking, nested directories, memory allocation of data blocks, and secure handling of name collision.

Restaurant UI/UX Design and Development using React | Personal Study

* Designed user-centric UX/UI design from wireframe to prototype using Figma following industry best practices and standards.
* Built a responsive, multi-page React web application using reusable components to display menu details and handle reservation forms efficiently.
* Ensured reliability and functionality through unit testing with Jest, validating component behavior and user interactions.
* Built a cross-platform mobile and web app using Expo (React Native) for users to construct workouts, track progress, view analytics, and manage fitness goals
* Integrated a Bluetooth IMU sensor (Witmotion WT901DCL) with Sensor Fusion Techniques to compute relative motion for analytical modeling
* Constructed and integrated .Net 9 REST API with CRUD functionality, Database Management, MySQL storage