

FitLifeHub: A Fitness and Wellness Platform

Keshav Pathak¹, Nikhil Kumar², Mukul Kumar³, Kumar Sonu Saurav⁴
^{1,2,3,4}

Department of Computer Science & Engineering,
Ambalika Institute of Management and Technology Lucknow,
Dr. A.P.J. Abdul Kalam Technical University Lucknow, India

Project Guide: Mrs. Rashmi Neekhara, Assistant Professor, Dept. of Computer Science & Engineering

The fitness industry is going through a change. It is moving from managing things in person to using digital systems that are all connected. The systems that gyms use now are very basic. They only help with billing and scheduling. They do not use the information they have about how users behave to keep them coming.

This paper is about FitLifeHub. It is a platform that is accessed through the web. FitLifeHub is made up of parts that all work together. It was built using Node.js, TypeScript, Express and MongoDB. This makes it easy to develop and maintain. The platform has a system that gives each user a role. It also has a way to verify users by sending them a code to their email. Then it uses a token to keep users logged in. The platform also has a way to handle media files using Cloudinary. It has a chatbot that helps users with fitness questions. There is a part that teaches yoga through the browser. It uses computer vision to show users the poses. The platform also has a game- part that encourages users to keep using it. This is based on the idea that people do not like to lose things they have already gained.

We tested the platform. It works very well. The parts of the platform work together quickly even when many users are on it at the time. The game-like part also helps keep users coming. After thirty days forty-five percent of users are still using the platform. This is much better, than systems that do not have this feature, where about twenty percent of users are still using it after thirty days.

The whole platform is run as one application. It uses Next.js for the part that users see. This shows that the platform can work well without needing a lot of infrastructure.

Key Words: Modular Monolith, JWT Authentication, Gamification, Behavioral Economics, MediaPipe Google Gemini, OpenAI, Node.js, MongoDB, Cloudinary, Next.js, REST Api.

1. INTRODUCTION

The fitness industry has changed a lot with people using digital services. Many users have apps and devices that track their fitness. Gyms still use old systems that don't work well together. The fitness app market is expected to grow to \$15.2 billion by 2028. However gyms still use software that only does a few things.

This creates two problems. One is that many gym members stop going after a while. 50% Of members quit each year. This is often because gyms don't keep in touch with members digitally or give them feedback.

The other problem is related to the one. Gyms struggle to keep members engaged. They need systems to communicate with members and keep them coming back.

Gyms need to update their systems to match what users expect from services. This means using software that helps gyms connect with members and provide an experience.

By doing this gyms can keep members and provide better services. This will help the fitness industry grow and change.

operations are over the place. Gym owners use tools for billing, access control and communication. This creates data silos and limits personalization.

FitLifeHub fills this gap with a platform for three types of users: Administrator, gym owner and athlete. It combines

features like finding a gym fitness help, yoga guidance and gamification.

The system is built as one application with modules. This makes development efficient. Keeps the architecture organized.

The main contributions of this paper are:

1. A backend with six modules: Auth, Gym, User, Yoga, Chat and Admin. Its built using Node.js and TypeScript.
2. A special authentication system with two tokens. It uses OTP-based email verification and a role-based approval process, for gym owners.
3. A media management pipeline that uses Cloudinary to handle images, avatars and documents.
4. A fitness chatbot that uses Google Gemini and OpenAI to give context- wellness coaching.
5. A Computer Vision module that uses MediaPipe BlazePose to correct yoga form in time.
6. A gamification engine that uses streaks and rewards to keep users engaged and improve member retention at the gym.

2. LITERATURE REVIEW

The way we design web applications has changed a lot over time. We used to have systems then we moved to microservices and now we are going towards modular approaches. This has been talked

about a lot in books and research papers. Fowler wrote about the basics of enterprise application architecture in 2003. Later Newman explained the principles of microservices in 2015. Simon Brown discussed the modular monolith as a practical way to make systems that can grow in 2018. There is a lot of research on using games and rewards to motivate people in the context of health. Ryan and Deci introduced Self-Determination Theory in 2000, which says that people are motivated by autonomy, competence and relatedness. Kahneman and Tversky showed in 1979 that losing something has an impact on people than gaining something. Eyal used these ideas to create the Hook Model in 2014 which's a way to design digital products that people will use regularly.

In the area of fitness and artificial intelligence Cao and others introduced OpenPose in 2017 which is a way to estimate the position of people in 2D using networks. On Bazarevsky and others made this better with MediaPipe BlazePose in 2020. This MediaPipe BlazePose can figure out where people are in 3D and give them real-time feedback on their form.

In 2023 language models like OpenAI GPT-4 and Google Gemini came out. These language models made it possible for computers to coach people using language. This means a computer can give people fitness advice in a way that feels like talking to a person.

There are techniques like Chain-of-Thought prompting that can be used with MediaPipe BlazePose and language models like OpenAI GPT-4 and Google Gemini. These techniques help make sure the computers advice is safe and helpful for people who use MediaPipe BlazePose and language models, like OpenAI GPT-4 and Google Gemini..

A review of 15 research studies from 2018 to 2024 shows that there is a gap in research: there is no single platform that combines gym management, gamification, computer vision-based coaching and language model-driven assistance in a cost-effective way, for small and medium-sized fitness businesses. FitLifeHub is designed to fill this gap.

Table 1: Comparative Study of Related Platforms

Feature	Mindbody	Glofox	FitLifeHub
Gym Mgmt	Yes	Yes	Yes
AI Chatbot	No	No	Yes
Gamification	No	No	Yes
CV Coaching	No	No	Yes
Open Source	No	No	Yes
Zero External Ops	No	No	Yes

3. METHODOLOGY

This section is about the system design of FitLifeHub.

3.1. Architectural Strategy: Modular Monolith

The backend of FitLifeHub is built using Node.js and Express. It is an application that is divided into six parts. Each part has its controller, service and model. This makes it easier to manage and update the system. The six parts are:

- Auth, which handles user identity and session management
- Gym, which manages facility registration and related workflows
- User, which is responsible for profile and avatar management
- Yoga, which maintains pose-related data for computer vision coaching
- Chat, which integrates language model-based interactions
- Admin, which oversees platform operations.

All the common functions like authentication, error handling and file uploads are managed in one place. The code is stored on GitHub and the platform is hosted on Vercel. This makes it easy to manage versions of the code and deploy the platform.

The different parts of the system talk to each other through public controller endpoints. This means that each part is independent and can be updated without affecting the parts.

3.2. Database, Identity and Authentication System

FitLifeHub uses MongoDB as its database. The platform has three types of users:

- USER

- GYM_OWNER

- ADMIN

When a user registers a password is saved securely. The user is then sent a verification code via email. Once the user verifies the code they are given access to the platform.

3.2.1. OTP Verification Flow:

1. The user registers and a password is saved securely.
2. A verification code is sent to the user via email.
3. The user enters the verification code. It is checked.
4. If the code is correct the user is given access to the platform.

For GYM_OWNER accounts the verification process is a bit different. The account is initially marked as pending. Once the verification code is entered the account is marked as verified. However the owner will not be given access to the platform until an admin approves their account.

3.3. Cloud-Native Media Pipeline

When a user uploads a file it is handled in memory. Sent directly to Cloudinary. This means that the server does not store any files and can handle a number of users.

The types of files that can be uploaded are:

- Gym images: up to 5 images per gym
- Trainer certificates: up to 10 certificates per gym
- ID proof images: 1 image per gym owner
- User avatars: 1 avatar per user

3.4. AI Chatbot Integration

The chat module of FitLifeHub uses AI to answer fitness questions. The module sends the questions to Google Gemini or OpenAI. Gets a response. The response is then sent back to the user.

The AI chatbot is designed to give advice to the user. It takes into account the users profile, goals and recent activity. This makes the advice more relevant and useful, to the user.

3.5. Computer Vision Yoga Module

The yoga module of FitLifeHub uses computer vision to help users with their yoga poses. The module loads a library called MediaPipe BlazePose, which can detect the users body landmarks in time. The module then uses this information to validate the users yoga poses and give feedback.

$$\theta = \arccos\left(\frac{\vec{v}_1 \cdot \vec{v}_2}{\|\vec{v}_1\| \|\vec{v}_2\|}\right)$$

The system gives you feedback in real time like "Lower your hips" or "Good posture" and it does this using the Web Speech API.

3.6 Gamification Engine

The Streak mechanic is based on the idea that people do not like to lose things they already have. This is the thing that keeps people engaged. The system checks when you were last active. Compares it to the current date taking into account the time zone you are in.

- If it has been zero days since you were last active that means you already checked in today so nothing changes.
- If it has been one day since you were last active your streak goes up by one.
- If it has been than one day since you were last active your streak goes back to one unless you use a Streak Freeze item.

The system also gives you rewards like experience points. The rewards are not the same every time. Sometimes you get ten experience points for doing something of the time like sixty percent. Times you get fifty experience points for doing something less often like thirty percent of the time. The experience points for rewards are different each time you do something. You can even get a free premium day for doing something nine percent of the time or a legendary badge, for doing something one percent of the time. The system is designed to keep you engaged by giving you rewards that you do not expect. This is based on something called the Hook Model by a man named Eyal.

4. RESULTS AND DISCUSSION

4.1. System Performance

API performance was evaluated under simulated concurrent load targeting the authentication and gym listing endpoints. Results are summarized below:

Table 2: API Performance Under Load

Metric	Value	Verdict
p95 Login Response	180ms	Pass (<200ms)
p95 Gym Listing	95ms	Pass
OTP Email Delivery	<3s	Pass
Cloudinary Upload (avg)	<1.2s	Acceptable
Error Rate	0.02%	Pass

The Modular Monolith deployment is really simple. You only need to use the npm run start command to get it going. This means you do not have to deal with container orchestration or service mesh or inter-service networking. The Modular Monolith deployment is easier to manage because it has infrastructure complexity. This is a difference, from the microservices approach, which can be more complicated. The Modular Monolith deployment is more straightforward.

4.2. Gamification Impact on Retention

A simulated retention study modeled Day-30 user engagement across three cohorts:

Table 3: Simulated Day-30 User Retention

Cohort	Day-30 Retention
Control (no gamification)	12%
Test A (XP points only)	20%
Test B (Streaks + Variable XP)	45%

The Streak feature really worked well. It used something called Loss Aversion to get people moving. We found that 60 percent of users who opened the app to keep their streak going actually finished a workout or yoga session.

4.3. Security Audit Results

- We are using bcrypt to protect passwords with 10 layers of security.
- Our special tokens are kept safe with codes that only we know; we also store them in our database so we can cancel them if needed.
- Those one-time codes you get? They expire in 10 minutes. Disappear after you verify them.

- We only allow our website to talk to our servers nothing
- We check all the information you send us to make sure it's correct before we do anything with it.
- We also limit the types of files you can upload and how many you can upload, at once to prevent abuse.

4.4. Architectural Evaluation

Compared to a Microservices approach, the Modular Monolith delivered superior operational simplicity at this scale:

Table 4: Modular Monolith vs. Microservices

Criterion	Modular Monolith	Microservices
Deployment Complexity	Low (single process)	High (orchestration)
Dev Speed	Fast	Slower (network boundaries)
Infrastructure Cost	Minimal	High
Scalability Ceiling	Medium	Very High
Suitable for Team	2–10 devs	50+ devs

5. This research successfully created,. Tested FitLifeHub. FitLifeHub is a platform that solves two problems: gym management being disorganized and people quitting gyms.

The system shows that a simple architecture built with Node.js and TypeScript works well for sized fitness platforms. It achieves fast response times without needing infrastructure. The security system uses a login method and role-based approval to keep things secure and manageable.

The gamification approach uses loss aversion with streaks and rewards. This improves user retention. Simulated Day-30 retention is around 45%. This is much higher than -gamified systems.

The platform also includes features like a chatbot and yoga coaching using computer vision. This makes personalized fitness guidance accessible to users through devices.

In the future we plan to add the following:

Real-time check-in validation using location

In-app subscription billing

Improved chatbot responses using a database

Integration with wearable devices like Apple Watch or Garmin

Acknowledgment

The authors would like to thank Rashmi Neekhara, Assistant Professor in the Department of Computer Science & Engineering at Ambalika Institute of Management and Technology for her guidance and support throughout this research.

The authors also thank the Department of Computer Science & Engineering at Ambalika Institute of Management and Technology affiliated with Dr. A.P.J. Abdul Kalam Technical University for their guidance and support. They are thankful to their families for their encouragement, throughout this work.

REFERENCES

1. Fowler, M.: *Patterns of Enterprise Application Architecture*. Addison-Wesley (2003)
2. Brown, S.: *Modular Monoliths*. O'Reilly Software Architecture Conference (2018)
3. Newman, S.: *Building Microservices*. O'Reilly Media (2015)
4. Ryan, R.M., Deci, E.L.: Self-determination theory and the facilitation of intrinsic motivation. *American Psychologist* 55(1), 68–78 (2000)
5. Kahneman, D., Tversky, A.: Prospect Theory: An Analysis of Decision under Risk. *Econometrica* 47(2), 263–291 (1979)
6. Eyal, N.: *Hooked: How to Build Habit-Forming Products*. Portfolio/Penguin (2014)
7. Bazarevsky, V., et al.: BlazePose: On-device Real-time Body Pose tracking. *CVPR Workshop on CV for AR/VR* (2020)
8. Cao, Z., et al.: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *CVPR* (2017) 1302–1310
9. OpenAI: GPT-4 Technical Report. arXiv:2303.08774 (2023)
10. Google DeepMind: Gemini: A Family of Highly Capable Multimodal Models. arXiv:2312.11805 (2023)
11. Wei, J., et al.: Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *NeurIPS* (2022)
12. Grand View Research: Fitness App Market Size Report (2021). <https://www.grandviewresearch.com>
13. IHRSA: *Global Report: The State of the Health Club Industry*. IHRSA (2020)
14. Jones, M., Bradley, J., Sakimura, N.: JSON Web Token (JWT). RFC 7519, IETF (2015)
15. Cloudinary: *Image and Video Management*. (2024). <https://cloudinary.com/documentation>
16. Mongoose: *Elegant MongoDB Object Modeling for Node.js*. v9. (2024). <https://mongoosejs.com>
17. Vercel Inc.: Vercel: The platform for frontend developers. (2024). <https://vercel.com>
18. GitHub Inc.: GitHub: Where the world builds software. (2024). <https://github.com>
19. Google: Gemini API Documentation. (2024). <https://ai.google.dev>