FitSloth Junior Software Engineer Technical Challenge

Overview

Welcome! This technical challenge is designed to assess your ability to build a full-stack web application using technologies that align with FitSloth's tech stack. You will have **3 weeks** to complete this challenge.

We're looking for engineers who can build meaningful products in the health, fitness, or positive behavior change space - domains that align with our mission to improve people's wellbeing.

Code Ownership & Transparency

You retain full ownership of all code you create for this challenge.

We want to be completely transparent:

- FitSloth will **NOT** claim any ownership or rights to your code
- This challenge is **solely for evaluating your technical skills** and approach
- You are free to continue developing your project after the challenge
- You can showcase this project in your portfolio, GitHub, resume, or anywhere else
- You can open-source it, monetize it, or do whatever you wish with it

This is your project, your idea, and your work. We're just excited to see what you build!

Challenge: Build Your Own Health/Fitness/Wellness Application

Your Mission

Design and build a **full-stack web application** related to health, fitness, wellness, or positive behavior change. The specific idea is entirely up to you - be creative!

Example Ideas (you are NOT limited to these):

- Habit tracking app (water intake, exercise, meditation)
- Workout planning and logging system
- Sleep tracking and analysis

What We're Looking For:

- A thoughtful, well-scoped idea that solves a real problem
- Demonstration of your technical skills
- Clean, maintainable code
- Understanding of full-stack development patterns
- Ability to work with modern development tools

Technical Requirements

Backend (Choose ONE)

- **Java 17+ with Spring Boot 3.x** OR
- **Node.js with Express.js**

Must Include:

- RESTful API design
- PostgreSQL database
- Proper authentication system (JWT recommended)
- Input validation and error handling
- At least 3 related database tables/entities
- Repository-Service-Controller pattern (or equivalent MVC pattern)

Security Requirements:

- **Password Hashing**: Use bcrypt, Argon2, or scrypt (NEVER store plain-text passwords)
- **JWT Authentication**: Implement token-based authentication for protected routes
- **Environment Variables**: Store all secrets (database credentials, JWT secret, API keys) in environment variables
- **Input Validation**: Validate and sanitize all user inputs to prevent injection attacks
- **HTTPS**: Use HTTPS for your deployed application (most free hosting services provide this)
- **No Hardcoded Secrets**: Never commit passwords, API keys, or secrets to GitHub

Frontend

- **Next.js 14+ with React 18+**
- **TypeScript** (strongly recommended)

Must Include:

- Responsive UI design
- Form validation

- Proper error handling and user feedback
- Integration with your backend API
- Authentication flow (login/register/logout)

Database

- **PostgreSQL**
- Proper schema design with relationships
- Database migrations (Flyway for Java, Sequelize/Prisma migrations for Node.js recommended)

Deployment

- **Deploy your backend** using **PM2** process manager
- Provide clear deployment documentation
- Application should be accessible via public URL (you may use free hosting services like Railway, Render, DigitalOcean, AWS free tier, etc.)

Version Control

- **Public GitHub repository**
- Clean, meaningful commit messages
- Proper .gitignore (no secrets, no node_modules, etc.)
- Regular commits showing development progress

AI-Assisted Development

We ENCOURAGE the use of AI coding assistants such as:

- Claude Code
- Cursor
- GitHub Copilot
- Codex
- Other AI pair programming tools

IMPORTANT: You must document:

1. Which AI tool(s) you used
2. How you used them

We want to see that you can:

- Effectively leverage AI tools to accelerate development
- Critically evaluate AI-generated code

- Make informed decisions about how to use AI assistance

Note: You will submit a separate document about your AI tool usage - see "Submission Requirements" below.

Submission Requirements

1. GitHub Repository

Your repository must include:

README.md

- **Project Overview**: What does your app do? What problem does it solve?
- **Tech Stack**: List all technologies used
- **Features**: What can users do with your app?
- **Database Schema**: Document your database structure (ERD or table descriptions)
- **API Documentation**: List your endpoints with request/response examples (or link to Swagger/OpenAPI)
- **Setup Instructions**:
  - Prerequisites (Node.js version, Java version, PostgreSQL, etc.)
  - How to install dependencies
  - How to set up the database
  - Environment variables needed
  - How to run the application locally
- **Deployment Guide**: How you deployed the application

.env.example

- Template for environment variables (with dummy values)
- Clear comments explaining each variable

Clean Code Structure

- Organized folder structure
- Separation of concerns (controllers, services, repositories, models, etc.)
- Consistent naming conventions
- Comments where necessary (but code should be self-documenting where possible)

2. Demo Application

Provide the following:

- **Live Deployment URL** - where we can test your application
- **Video Walkthrough** (3-10 minutes) - demonstrating all features

For a video, please:

- Show the application in action (user flows, features)
- Upload to YouTube and include a link in the submission email.

3. Separate Documents (Submit via Email)

Create **two separate documents** to submit via email (these should NOT be in your public GitHub repository):

AI-Assisted Development Document

A document (PDF, Google Doc, or Markdown) describing:

- Which AI tool(s) you used (Claude Code, Cursor, GitHub Copilot, etc.)
- How you used them throughout the development process
    - Examples: code generation, debugging, learning new concepts, refactoring, test writing
- What worked well and what didn't when using AI assistance
- What you learned from using these tools
- Your thoughts on AI-assisted development

Challenges & Learnings Document

A personal reflection document (PDF, Google Doc, or Markdown) covering:

- **Challenges Faced**: What was difficult during this project? Technical blockers? Learning curves?
- **How You Overcame Them**: Problem-solving approaches, resources used, breakthroughs
- **Technical Learnings**: What new technologies or concepts did you learn?
- **Mistakes & Iterations**: What did you try that didn't work? What would you do differently?
- **Key Takeaways**: What are the most important things you learned from this challenge?
- **Future Improvements**: If you had more time, what would you add or improve?

4. Email Submission

Send an email to hr@fitsloth.co.th with:

- Subject: "Junior SE Technical Challenge - Your Name"
- Link to your GitHub repository
- Link to deployed application and demo video
- **Attached**: AI-Assisted Development document
- **Attached**: Challenges & Learnings document

Submission Deadline: 3 weeks from assignment

What Happens Next?

- We will review your submission and aim to respond within **7 business days**
- If your submission passes our evaluation, you will receive an email to proceed with the interview process
- All candidates will receive feedback on their submission

Evaluation Criteria

Your submission will be evaluated on the following:

- Functionality
- Application Idea
- Code Quality
- API Design
- Database Design
- Authentication & Security
- Frontend Quality
- Documentation
- Git Usage
- Deployment
- AI Tool Usage
- Testing (Nice to Have)

Questions?

If you have questions about the requirements or need clarification, please email hr@fitsloth.co.th with:

- Subject: "Question: Junior SE Technical Challenge"
- Your specific question

We aim to respond within 24 hours.

Good Luck!

We're excited to see what you build! Remember:

- **Quality over quantity** - a well-built simple app is better than a broken complex one
- **Show your thinking** - we want to understand your approach and decision-making
- **Be creative** - this is your chance to show us how you think about product development

- **Have fun** - build something you're proud of!

This challenge is your opportunity to demonstrate not just your coding skills, but your ability to:

- Understand user needs
- Make technical decisions
- Build production-ready software
- Learn and adapt using modern tools
- Communicate your work effectively

We're rooting for you!

FitSloth Tech Team