Software Engineering Lab

A curated collection of software engineering projects, experiments, and hands-on labs aimed at building real-world, production-ready skills.

This repository serves as a personal engineering workspace and portfolio, covering multiple domains across modern software development.

🔍Focus Areas

- **Backend Development** – APIs, databases, authentication, microservices
- **Frontend Development** – Web applications, UI/UX, frameworks
- **DevOps & Cloud Engineering** – CI/CD, containers, infrastructure as code
- **AI & Automation** – Intelligent systems, data processing, automation tools
- **Systems Design** – Scalable architectures, performance, reliability
- **Networking & Security** – Networking labs, security fundamentals, defense strategies

---

📁Repository Structure

Each directory represents an independent project or lab:

```
software-engineering-lab/
├── backend/
├── frontend/
├── devops/
├── ai-ml/
├── networking-security/
├── system-design/
└── README.md
```

Each project contains its own documentation, setup instructions, and implementation details.

🕑Objectives

- Apply theoretical concepts through hands-on implementation
- Build and document scalable, maintainable software
- Explore modern engineering tools and best practices
- Create a strong technical portfolio for academic and professional growth

---

🗜️ Getting Started

1. Clone the repository

```
git clone https://github.com/ambani-elphas/software-engineering-lab.git
```

2. Navigate to a project directory
3. Follow the project-specific README for setup and execution

---

📌 Notes

- Projects may use different languages, frameworks, and tools
- Some labs are experimental and focused on learning outcomes
- Improvements and refactoring are continuous

Author

Elphas Ambani
Software Engineer | DevOps | AI & Systems Enthusiast

📄 License

🧱 Project Components (Scaffold)

Below are the **actual components** that make up this lab, with clear responsibilities and starter templates you can copy directly into folders.

✂️ Backend

Purpose: APIs, business logic, databases

Suggested stack: Node.js (NestJS/Express) or Python (FastAPI)

Folder: `backend/`

Starter structure:

```
backend/
├── src/
│   ├── app.(ts|py)
│   ├── routes/
│   ├── services/
│   └── models/
├── tests/
├── Dockerfile
├── README.md
└── package.json / pyproject.toml
```

backend/README.md

```
# Backend
API and business logic implementations.

## Features
- RESTful APIs
- Authentication & authorization
- Database integration

## Run
npm install && npm run dev
```

✒️Frontend

Purpose: User interfaces and client-side logic

Suggested stack: React, Angular, or Vue

Folder: `frontend/`

Starter structure:

```
frontend/
├── src/
│   ├── components/
│   ├── pages/
│   └── services/
├── public/
```

```
├── README.md
└── package.json
```

frontend/README.md

```
# Frontend
User-facing web applications.

## Features
- Responsive UI
- API integration
- Component-based architecture

## Run
npm install && npm start
```

🛏️DevOps

Purpose: Automation, CI/CD, infrastructure

Folder: `devops/`

Starter structure:

```
devops/
├── docker/
├── github-actions/
├── terraform/
├── scripts/
└── README.md
```

devops/README.md

```
# DevOps
Infrastructure, automation, and deployment configurations.

## Includes
- Docker & containerization
- CI/CD pipelines
- Infrastructure as Code
```

🌂 AI / Machine Learning

Purpose: Intelligent systems and automation

Folder: `ai-ml/`

Starter structure:

```
ai-ml/
├── notebooks/
├── data/
├── models/
├── scripts/
└── README.md
```

ai-ml/README.md

```
# AI & Machine Learning
AI-driven experiments and automation projects.

## Topics
- Data preprocessing
- Model training
- Inference & evaluation
```

🌡️ Networking & Security

Purpose: Networking labs and cybersecurity practice

Folder: `networking-security/`

Starter structure:

```
networking-security/
├── labs/
├── configs/
├── scripts/
└── README.md
```

networking-security/README.md

```
# Networking & Security
Hands-on labs covering networking concepts and security fundamentals.
```

🎒System Design

Purpose: Architecture and scalability

Folder: `system-design/`

Starter structure:

```
system-design/
├── diagrams/
├── case-studies/
├── notes/
└── README.md
```

system-design/README.md

```
# System Design
Design exercises focusing on scalability, reliability, and performance.
```

🔗Next Steps

1. Create the folders above in your repository
2. Add the README templates to each folder
3. Start committing projects incrementally

This structure is **portfolio-ready**, **scalable**, and aligns with real-world engineering workflows.