Fullstack Engineer Assessment

PT Entropi Global Martech
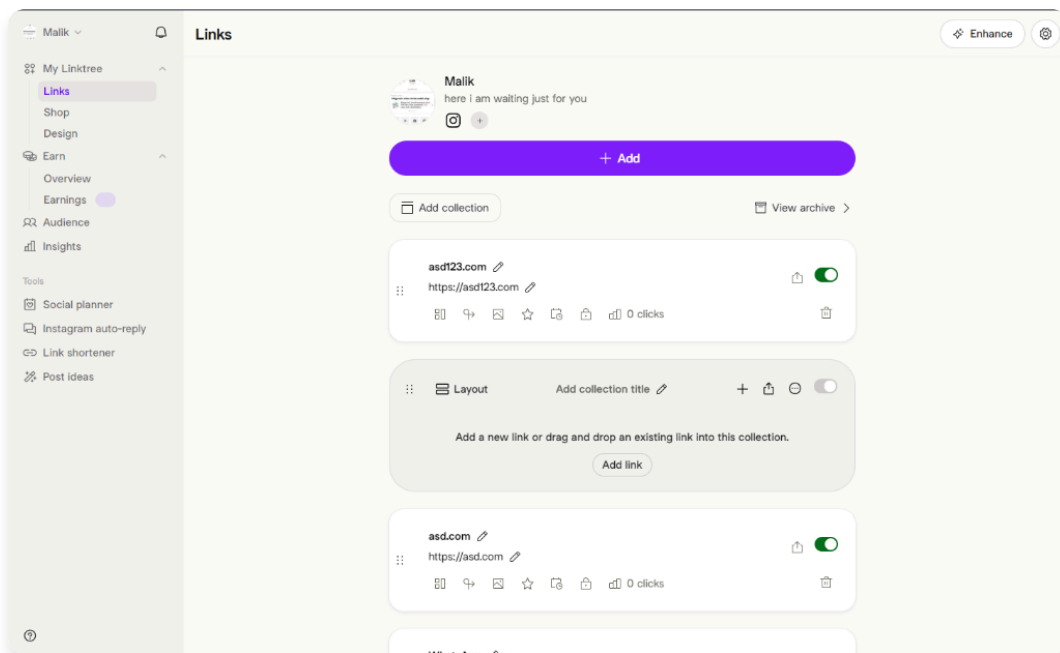
Overview

Build a **"Link in Bio" application** (similar to Linktree) with admin dashboard and public profile page.

Deadline: 2 Days

Reference Design

Use this Linktree admin interface as your visual reference:

What to Build

1. Public Profile Page (Mobile-First)

A shareable page showing:

- User avatar & bio
- List of clickable links
- Clean, modern design

2. Admin Dashboard

A simple dashboard to:

- Add/Edit/Delete links
- Toggle links on/off
- Reorder links (drag & drop optional)
- Edit profile (name, bio, avatar URL)

3. Backend API

Simple REST API for:

- CRUD operations for links
- Profile management
- Basic authentication (login/register)

Tech Stack

Layer	Technology
Frontend	Next.js 14+ (App Router)
Styling	Tailwind CSS or custom CSS
Backend	Next.js API Routes
Database	PostgreSQL + Prisma (or SQLite for simplicity)
Auth	NextAuth.js or simple JWT

Requirements

Must Have (Core)

- ☐ User can register and login
- ☐ User can add links (title + URL)
- ☐ User can edit and delete links
- ☐ User can toggle links active/inactive
- ☐ Public profile page displays user's links
- ☐ Mobile-responsive design

Nice to Have (Bonus)

- ☐ Drag & drop reorder
- ☐ Click tracking/analytics
- ☐ Dark mode
- ☐ Custom theme colors

Database Schema (Suggested)

```
model User {
  id        String   @id @default(uuid())
  email     String   @unique
  password  String
  name      String
  bio       String?
  avatar    String?
  links     Link[]
  createdAt DateTime @default(now())
}

model Link {
  id        String   @id @default(uuid())
  title     String
  url       String
  active    Boolean  @default(true)
  order     Int      @default(0)
  userId    String
  user      User     @relation(fields: [userId], references: [id])
  createdAt DateTime @default(now())
}
```

Submission

Required

1. **GitHub Repository** (public)
2. **README.md** with:
   - Setup instructions
   - What you implemented
   - Time breakdown

Bonus

- Live demo (Vercel/Railway)

Evaluation Criteria

Criteria	Weight
Functionality (everything works)	40%
Code Quality (clean, organized)	25%
UI/UX (looks good, easy to use)	25%
Documentation	10%

Tips for Success

💡 **Focus on core features first!** A working MVP with clean code beats incomplete fancy features.

Suggested 2-day breakdown:

- **Day 1:** Database setup, Auth, API routes, basic CRUD
- **Day 2:** UI polish, public profile page, testing, deploy

Contact

For questions, contact our HR team: 📧 **baby@entropimartech.com**

Good luck! We're excited to see what you build. 🚀