

# Developer Guide – QuickForms AI

**Course:** CSC 318 – Software Engineering Principles

**Assignment:** Final Project – QuickForms AI

**Deliverable:** *Developer Guide*

**Project Title:** QuickForms AI – Smart Form & Workflow Builder

**Student:** Eric Amoh Adjei

**Instructor:** Professor Rawad Habib

**Date:** August 2025

---

## 1. Overview

QuickForms AI is a smart online form generator designed to help small businesses, student groups, and organizations create, share, and collect form submissions efficiently. It combines a **React frontend**, an **Express backend**, a **Prisma-managed database**, and an **AI-powered schema generator**.

This guide explains how developers can:

- Set up the project locally.
- Run and test the system.
- Understand the project structure.
- Contribute new features.
- Prepare production builds.

---

## 2. Technology Stack

Layer	Technology	Notes
Frontend	React + Vite + TypeScript	Dynamic UI, drag-and-drop editor
Backend	Express.js + TypeScript	REST API, handles form requests & AI inte
Database	Prisma ORM + SQLite/Postgres	SQLite default (dev), scalable to Postgres/M
Versioning	Git + GitHub	Central repo, supports branching & PR worl

<b>Build Tools</b>	npm, pnpm, tsc, vite	Used for builds, scripts, and dependency management
<b>Testing</b>	Vitest, Jest, Supertest	Unit, integration, and system testing

---

## 3. ⚡ Getting Started

### 3.1 Clone the Repository

```
git clone https://github.com/amoheric/quickforms-ai.git
cd quickforms-ai
```

### 3.2 Install Dependencies

#### Frontend

```
cd apps/web
npm install
```

#### Backend

```
cd ../api
npm install
```

### 3.3 Environment Setup

Create a `.env` file inside `apps/api/` with the following:

```
DATABASE_URL="file:./dev.db"
PORT=4000
OPENAI_API_KEY="your_api_key_here"
```

### 3.4 Database Setup

```
cd apps/api
npx prisma migrate dev --name init
```

---

## 4. 🚀 Running the Application

**Start Backend (API):**

```
cd apps/api  
npm run dev
```

- Runs at: `http://localhost:4000`

### Start Frontend (Web):

```
cd apps/web  
npm run dev
```

- Runs at: `http://localhost:5173`

✅ Open `http://localhost:5173` to access QuickForms AI locally.

---

## 5. Testing

### 5.1 Unit Tests

```
cd apps/api  
npm test
```

- Validates functions and API endpoints.

### 5.2 Integration Tests

- Run backend: `npm run dev`
- Run frontend: `npm run dev`
- Simulate form creation + submission.

### 5.3 System Tests

- Full workflow: Create → Fill → Submit → Export CSV.
- Tested on Chrome, Edge, and Firefox.

---

## 6. Project Structure

```
quickforms-ai/
```

```
— apps/
  └─ api/      # Express backend
  └─ web/      # React frontend
— docs/        # Reports, UML, Manuals
— prisma/      # Database schema
— tests/       # Automated test cases
— README.md
— DeveloperGuide.md
— LICENSE
— .gitignore
```

- **apps/api** → Backend API with routes, controllers, and services.
- **apps/web** → React frontend with components, hooks, and views.
- **docs** → Reports, UML diagrams, and manuals for submission.
- **prisma** → Schema and migrations for SQLite/Postgres.
- **tests** → Unit and integration tests.

---

## 7. Contribution Guidelines

1. **Fork the Repository** on GitHub.
2. **Create a Feature Branch:**
3. `git checkout -b feature/my-feature`
4. **Commit with Clear Messages:**
5. `git commit -m "Add form export to CSV"`
6. **Push and Open a Pull Request** for review.

### Code Style:

- Follow ESLint + Prettier rules.
- Use camelCase for variables/functions; PascalCase for React components.
- Write unit tests for new features.

---

## 8. Building for Production

### Frontend

```
cd apps/web
npm run build
```

## Backend

```
cd apps/api  
npm run build
```

## Deployment

- Bundle artifacts with PowerShell script (Windows) or manually zip.
  - Recommended hosts: **Vercel** (frontend) + **Render/Heroku** (backend).
  - Database: Postgres for production deployment.
- 

## 9. Future Improvements

- Add **user authentication** for private forms.
- Expand AI prompts for richer form field types.
- Implement **role-based dashboards** (admin vs. respondent).
- Introduce CI/CD pipeline for automated testing + deployment.