Nexus Universe 3.0 – Master Blueprint Document

# 📘 Table of Contents

1. 1. Overview and Vision
2. 2. Feature Roadmap (Phases 1–5)
3. 3. Roles and Responsibilities
4. 4. Project Timeline and Milestones
5. 5. GitHub Project Structure
6. 6. Swagger + Postman Integration
7. 7. Content Generation (AI/Manual)
8. 8. Living Document Guidelines

# 1. Overview and Vision

Nexus Universe 3.0 is a globally scalable, AI-powered virtual school ecosystem for ages 4–16. It blends traditional schooling structures with cutting-edge technologies—covering multi-curriculum alignment (NG, UK, IPC, IB, CAIE, SG), gamified learning, AI-driven auto-assessments, DRM protection, and marketing automation. All built with CI/CD, container-ready deployment, and strong observability tooling.

# 2. Feature Roadmap (Phases 1–5)

* Phase 1: Repo Scaffold, JWT/2FA Auth, Basic Routes, Zod Validation, Swagger & Postman Setup
* Phase 2: Course/Module/Lesson Schema, Timetables, Lesson Engines, Frontend Zod Forms
* Phase 3: VR Integration, Math & Tracing Tools, Notifications, Cron Jobs
* Phase 4: Proctoring, Assessment AI, Badging, Completion/Progress Tracking
* Phase 5: Payments, Global Onboarding, Admin Dashboards, Marketing Auto-Campaigns

# 3. Roles and Responsibilities

* 🧑‍💻 Developer – Implements APIs, frontend, and test suites.
* 📚 Curriculum Lead – Ensures multi-curricula content is aligned.
* 🎨 Content Creator – Generates VR, games, media, and lesson content.
* 🛡️ QA/Security – Tests auth, DRM, CI/CD pipelines, and blockers.
* 🧠 AI Integrator – Connects GPT + auto-gen tools + AI content logic.
* 📊 Analytics – Ensures observability + reporting + Paystack integration.

# 4. Project Timeline and Milestones

Estimated 90-day execution window:  
• Phase 1: Week 1–3  
• Phase 2: Week 4–6  
• Phase 3: Week 7–8  
• Phase 4: Week 9–11  
• Phase 5: Week 12+

# 5. GitHub Project Structure

Structure:  
- backend/: API logic, auth, routes, Prisma schemas  
- frontend/: Next.js + Tailwind + Zod + RHF forms  
- docs/: Swagger YAMLs, Markdown references  
- postman/: Postman collections (auto-testable)  
- .github/workflows/: CI/CD automation with GitHub Actions

# 6. Swagger + Postman Integration

• Swagger generated via swagger-jsdoc and swagger-ui-express.  
• Postman test suite auto-generated for CRUD flows.  
• Newman CLI used for CI/CD testing.  
• Postman collection file stored in /postman/ directory.

# 7. Content Generation (AI/Manual)

• Use GPT to generate quizzes, modules, auto-assessments, and ad copy.  
• Bannerbear API to generate banners for marketing.  
• Buffer API to schedule campaigns.  
• Manual upload areas documented in the docs/ folder.

# 8. Living Document Guidelines

• Keep this doc and Notion copy in sync.  
• Update each feature/phase as it ships.  
• Version and link new commits using semver in README.md and changelog.  
• Use standard-version or similar tool to tag releases.  
• Sync Swagger + Postman regularly as new endpoints are added.