Here’s a **step-by-step project roadmap** tailored to your PHP background and current Golang skill level. Each step includes suggested **project ideas**, the **skills you’ll build**, and **optional tech stacks** to improve your job-readiness.

**🟢 Phase 1: Build Solid Foundations (You’re Here)**

**Goal:** Get comfortable with basic Go syntax, REST APIs, Gin, GORM, and MySQL.

**✅ Project 1: Basic Todo App (CRUD)**

* **Tech Stack:** Go, Gin, GORM, MySQL
* **Skills:**
  + REST API development
  + Database migrations
  + Basic authentication (optional)

Bonus: Add Swagger for API docs

**🟡 Phase 2: Intermediate Projects – Job Ready CRUD Systems**

**Goal:** Handle larger codebases, understand Go project structure, modularization, and middleware.

**✅ Project 2: Blog Platform**

* **Features:** Posts, comments, users, likes
* **Tech Stack:** Go, Gin, GORM, MySQL, JWT, Bcrypt
* **Skills:**
  + JWT-based authentication
  + Role-based access control
  + Pagination, search, filtering

**✅ Project 3: E-commerce Backend (REST API only)**

* **Features:** Products, categories, carts, orders, users
* **Skills:**
  + Relationships in GORM
  + Transactions
  + Image uploads (to local storage or S3)

Bonus: Add Redis for caching popular products

**🟠 Phase 3: Advanced Projects – Backend & DevOps Skills**

**Goal:** Simulate real-world production-ready apps with security, Docker, CI/CD.

**✅ Project 4: Microservices Task Manager**

* **Services:** Auth, User, Task, Notification
* **Tech Stack:** Go (Gorilla Mux or Gin), MySQL, Kafka/RabbitMQ, Docker, Docker Compose
* **Skills:**
  + Microservices architecture
  + Pub/Sub with Kafka or RabbitMQ
  + Docker and Docker Compose
  + Clean architecture

**✅ Project 5: Golang + gRPC-based Chat App**

* **Skills:**
  + Real-time communication
  + gRPC vs REST
  + Protocol Buffers

Bonus: Use WebSockets or gRPC streaming for real-time chat

**🔵 Phase 4: Cloud-Native & Production-Ready**

**Goal:** Learn CI/CD, cloud deployment, monitoring

**✅ Project 6: SaaS-style Analytics Dashboard**

* **Tech Stack:** Go, Gin, PostgreSQL, Redis, Docker, Prometheus + Grafana, Kubernetes (optional)
* **Skills:**
  + Metrics instrumentation (Prometheus)
  + Deploying to AWS/GCP
  + CI/CD with GitHub Actions or GitLab

Bonus: Use Terraform for IaC or deploy to Kubernetes

**🧩 Complementary Tech Suggestions**

To land a **high-paying Go job**, consider learning:

* **Docker & Kubernetes** – for container orchestration (essential in DevOps roles)
* **gRPC** – for high-performance services
* **Redis** – for caching
* **Kafka / RabbitMQ** – for asynchronous communication
* **AWS / GCP / Azure** – basic services: EC2, RDS, S3
* **CI/CD tools** – GitHub Actions, GitLab CI
* **Testing in Go** – testing pkg, mocks, table-driven tests
* **Go Concurrency** – goroutines, channels, worker pools

**📘 Learning Strategy**

* **Hands-on First:** Build > break > fix
* **Read Source Code:** Study popular open-source Go projects on GitHub
* **Contribute to Open Source:** Great way to improve and showcase skills
* **LeetCode (Go):** Improve DSA & concurrency skills for interviews

**🚀 After the Projects**

* Build a **portfolio website** (host on GitHub Pages or Netlify)
* Write tech blogs about your projects or Go learnings
* Upload code to GitHub with clean README + Docker setup
* Practice **Go-specific interview questions**