Here's a **task list with a timeline** to create your **Blog Platform** application using **Go, Gin, GORM, MySQL, JWT, and Bcrypt**, incorporating the listed features and skills.

**🗓️ Timeline Overview (Estimated: 4-5 Weeks)**

| **Week** | **Phase** | **Description** |
| --- | --- | --- |
| Week 1 | Setup & Auth | Project setup, DB schema, user auth |
| Week 2 | Core Features | Posts, comments, likes |
| Week 3 | User Roles & Access | Role-based access, permissions |
| Week 4 | Enhancements | Pagination, search, filtering |
| Week 5 | Testing & Deployment | Final testing, deployment |

**✅ Task List with Timeline**

**Week 1: Project Setup & Authentication**

**Day 1-2: Project & DB Setup**

* Initialize Go module
* Install Gin, GORM, MySQL driver
* Set up .env and config files
* Connect to MySQL via GORM
* Define basic folder structure

**Day 3-4: User Model & Auth**

* Define User model (with roles)
* Add password hashing with Bcrypt
* Create registration & login endpoints
* Generate JWT tokens for login

**Day 5-6: Middleware & Token Handling**

* Create middleware to parse/validate JWT
* Protect routes with authentication middleware
* Add refresh token support (optional)

**Week 2: Posts, Comments, Likes**

**Day 1-2: Post Feature**

* Define Post model
* Implement CRUD routes
* Associate posts with users

**Day 3: Comment Feature**

* Define Comment model
* Implement create/list/delete comments
* Associate comments with posts & users

**Day 4: Likes Feature**

* Define Like model
* Implement like/unlike functionality
* Prevent duplicate likes

**Day 5-6: Test Core Features**

* Unit test all major endpoints
* Check basic error handling and edge cases

**Week 3: Roles & Access Control**

**Day 1: Role Definition**

* Add roles to User model (admin, user)
* Seed default admin user

**Day 2-3: Access Control Middleware**

* Middleware to enforce role-based access
* Restrict post/comment deletion/edit to owners or admins

**Day 4-5: Admin Routes**

* Add admin routes for user and post moderation
* Logging for moderation actions (optional)

**Week 4: Pagination, Search, Filtering**

**Day 1: Pagination**

* Add pagination (limit/offset) to list endpoints

**Day 2: Search**

* Enable post search by title/content
* Add full-text search indexing (optional)

**Day 3: Filtering**

* Filter posts by user, tags, date, etc.

**Day 4-5: API Cleanup**

* Standardize response structure (e.g., JSON format)
* Add Swagger/OpenAPI docs (optional)

**Week 5: Testing, Optimization & Deployment**

**Day 1-2: Testing**

* Write integration tests
* Load test core endpoints

**Day 3: Error Handling & Validation**

* Improve input validation with binding
* Add structured error responses

**Day 4-5: Deployment**

* Dockerize app
* Prepare MySQL schema/init script
* Deploy to server or cloud (e.g., Render, DigitalOcean)

**🧩 Optional Enhancements (Post-MVP)**

* Email verification & password reset
* Tagging system for posts
* Post image upload (using S3)
* Web frontend (React, etc.)