

✓ 1. Basic CRUD App (with Filtering & Aggregation)

Project: Book Collection API

Create APIs to:

- ➕ Add a book (`title`, `author`, `description`, `genre`, `pages`, `publishedYear`, `rating`)
 - 📄 Get all books with:
 - Advanced filtering (by genre, author, year, rating range)
 - Pagination, search, and sorting
 - ↻ Update a book by ID
 - ✖ Delete a book by ID
 - 📊 Aggregation:
 - Get average rating per genre
 - Count books per author
 - List top 5 highest-rated books
-

✓ 2. User Auth System

Project: Auth API with JWT

Create features to:

- Register and login with JWT
- Hash passwords using bcrypt
- Protect routes using JWT tokens
- Add `role` to users (`user`, `admin`)
- Implement middleware: `authMiddleware`, `protectRoute`, `restrictTo`

✓ 3. File Upload & Media

Project: User Profile with Avatar Upload

Create features to:

- Register/login users
 - Upload and update profile picture with **Multer**
 - Store image filename/path in MongoDB
 - Serve images using Express (`app.use('/uploads', express.static(...))`)
 - Validate image type and size
-

✓ 4. Real-world App

Project: Blogging Platform API

Create features to:

- User auth (register/login)
 - Blog post CRUD (by authenticated users)
 - Each blog includes:
 - Title, content, author ID
 - Optional image (uploaded using Multer)
 - Public routes: view blogs
 - Private routes: create/edit/delete blogs (logged-in users)
 - Admin can delete any post
-

✓ 5. Complete App with All Features

Project: Job Portal API

User roles:

- **job seekers** and **employers** with role-based access

Employers can:

- Post jobs (CRUD)
- Upload company logo (Multer)

Job Seekers can:

- Apply to jobs
- Upload resume (PDF)

Authentication:

- JWT-based login
- Role-based route protection

Advanced Features:

- Pagination for job listings
- Filter/search by job title, location, type
- Admin dashboard (get all users/jobs)



Optional Bonus Projects

Project	Highlights
Image Gallery	Upload, resize, and serve images using Sharp + Multer
Task Manager	JWT auth, user-specific tasks, file attachments
E-commerce API	Auth, product CRUD, cart system, image uploads



Reusable Project Folder Structure

```
project/
├─ controllers/      # Request logic
├─ models/           # Mongoose schemas
├─ routes/           # API route definitions
├─ middleware/       # Auth, error handling
├─ uploads/          # Multer file storage
├─ utils/            # Helper functions
├─ app.js            # Entry point
├─ config.js         # DB config, env setup
```

Let me know if you want this as a **PDF**, **Notion template**, or a **GitHub starter boilerplate** for these projects!