Backend Development Learning Checklist

Stage 1: Java Core Foundation

- [ ] Classes, objects, and constructors
- [ ] Inheritance, abstraction, interfaces
- [ ] Exception handling (try-catch, custom exceptions)
- [ ] Collections (List, Map, Set)
- [ ] Generics (List<T>, methods with <T>)
- [ ] Streams and Lambda expressions
- [ ] Java enum, static, final usage

Stage 2: Spring Boot Basics

- [ ] Project structure: main(), @SpringBootApplication
- [ ] Dependency injection with @Autowired
- [ ] Build tool (Maven): add/edit pom.xml
- [ ] REST API: @RestController, @RequestMapping, @GetMapping, etc.
- [ ] Use application.properties for configuration

Stage 3: JPA and Database

- [ ] Define entities with @Entity, @Id, @GeneratedValue
- [ ] Repository interface (extends JpaRepository)
- [ ] Basic queries: findBy, save, deleteById, etc.
- [ ] Custom queries using @Query
- [ ] Relationships: @OneToMany, @ManyToOne (basic)
- [ ] Connect to a PostgreSQL/MySQL database

Stage 4: REST & JSON Fundamentals

- [ ] Understand HTTP verbs: GET, POST, PUT, DELETE
- [ ] HTTP status codes (200, 201, 400, 404)
- [ ] JSON structure and data mapping
- [ ] Use Postman to test endpoints

Stage 5: DTOs & Validation

- [ ] Create DTO classes for API requests and responses

Backend Development Learning Checklist

[ ] Use @Valid and validation annotations like @NotBlank, @Size

[ ] Convert between DTO and Entity in service layer

Stage 6: Error Handling

[ ] Create a @ControllerAdvice class

[ ] Use @ExceptionHandler for specific exceptions

[ ] Return custom error messages and status codes

Stage 7: Pagination & Sorting

[ ] Use Pageable and PageRequest in service layer

[ ] Accept pagination parameters in controller

[ ] Return paginated DTOs

[ ] Add optional sorting using Sort.by(...)

Stage 8: Unit Testing

[ ] Use @SpringBootTest and @MockBean

[ ] Write service tests with Mockito.when(...).thenReturn(...)

[ ] Assert outputs with assertEquals, etc.

[ ] Test controller separately with @WebMvcTest (optional)

Stage 9: Authentication (JWT)

[ ] Add Spring Security dependency

[ ] Setup basic JWT-based login/register/auth flow

[ ] Secure endpoints with roles

[ ] Use JWT filter and token provider

[ ] Pass token in Postman Authorization header

Stage 10: Git & Project Deployment

[ ] Initialize Git repo and commit code

[ ] Push code to GitHub

[ ] Deploy backend on: Railway or Render or Fly.io

[ ] Set environment variables (DB_URL, JWT_SECRET, etc.)

Backend Development Learning Checklist

Optional Concepts (Later)

- [ ] Caching with @Cacheable
- [ ] File upload and download
- [ ] Swagger/OpenAPI documentation
- [ ] Environment-based configuration (application-dev.properties, etc.)
- [ ] Email sending / notifications

Suggested Mini Projects

Project	Concepts Covered
Library System	CRUD, DTO, pagination, error handling
Course Manager	Relationships, validations, sorting
User Auth API	JWT, roles, login/register
Task Tracker	Filtering, custom queries, soft delete