**Setup Instructions**

1. **Install Prerequisites**:
   * Java Development Kit (JDK 11 or higher).
   * MySQL database.
   * Redis server.
   * Maven or Gradle for project management.
2. **Clone or Create the Project**:
   * Clone the project repository or initialize a new Spring Boot application with the structure provided.
3. **Configure MySQL Database**:
   * Create a MySQL database named demo\_db or adjust the application.properties file to reflect your database setup.
   * Use the following credentials or update them:

properties

spring.datasource.url=jdbc:mysql://localhost:3306/demo\_db?useSSL=true&requireSSL=true

spring.datasource.username=root

spring.datasource.password=yourpassword

1. **Start Redis Server**:
   * Ensure the Redis server is running on localhost:6379.
2. **Build the Project**:
   * For Maven:

mvn clean install

* + For Gradle:

./gradlew build

1. **Run the Application**:
   * Use the following command:

mvn spring-boot:run

* + Access the API at http://localhost:8080.

1. **Test the Application**:
   * Run unit and integration tests:

mvn test

**Code Architecture**

1. **Core Components**:
   * **Entity Layer**: Defines the database schema (e.g., User class).
   * **Repository Layer**: Interfaces for database operations using Spring Data JPA.
   * **Service Layer**: Business logic for interacting with the repository and handling caching.
   * **Controller Layer**: RESTful API endpoints to expose CRUD operations.
2. **Caching**:
   * Implemented using Redis for performance optimization.
   * Integrated with Spring Cache annotations (@Cacheable, @CacheEvict).
3. **Concurrency**:
   * ThreadService handles multithreading with ExecutorService for concurrent task execution.
4. **Performance Monitoring**:
   * Spring Boot Actuator is enabled to expose performance metrics for endpoints.
5. **Testing**:
   * **Unit Tests**:
     + Mock repository dependencies using Mockito.
     + Test CRUD operations in isolation.
   * **Integration Tests**:
     + Validate the REST API using TestRestTemplate.

**Directory Structure**

src/

├── main/

│ ├── java/com/example/demo/

│ │ ├── DemoApplication.java # Main application class

│ │ ├── config/RedisConfig.java # Redis configuration

│ │ ├── entity/User.java # Entity class for User

│ │ ├── repository/UserRepository.java # Repository interface

│ │ ├── service/ # Business logic

│ │ │ ├── UserService.java # User service

│ │ │ └── ThreadService.java # Multithreading logic

│ │ ├── controller/UserController.java # REST controller

│ │ └── performance/PerformanceLogger.java # AOP for logging

│ └── resources/

│ ├── application.properties # Application configuration

├── test/

│ ├── java/com/example/demo/

│ │ ├── service/UserServiceTest.java # Unit tests

│ │ └── controller/UserControllerIntegrationTest.java # Integration tests