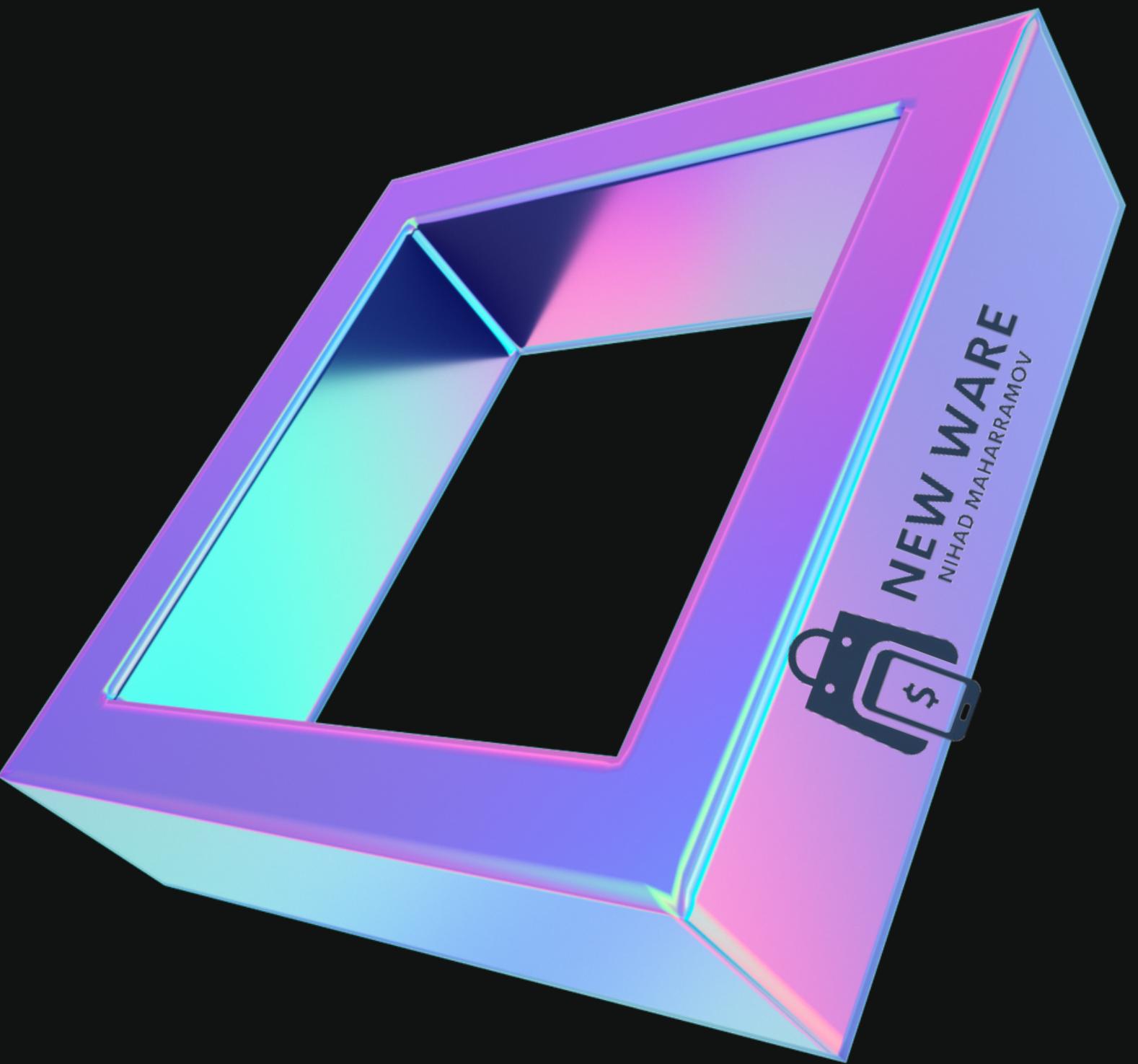
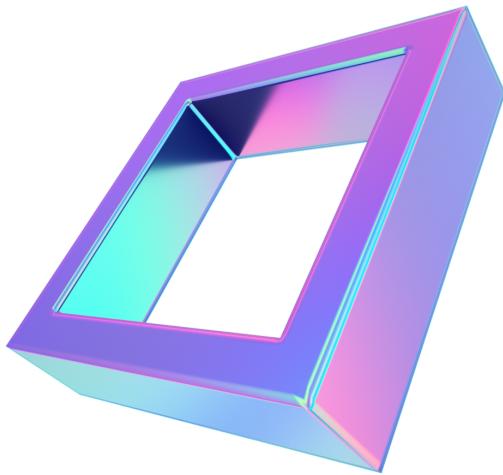


NEW WARE

Nihad Maharramov (INHA IT-3)



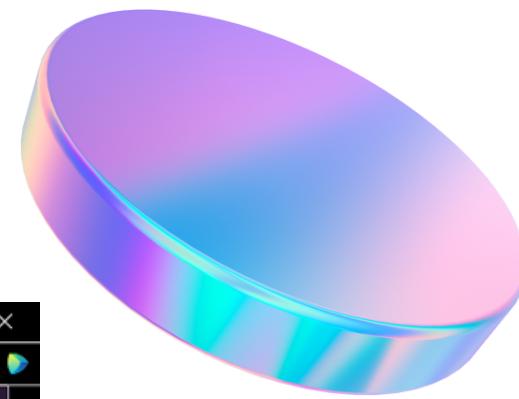
01



NEW WARE
NIHAD MAHARRAMOV

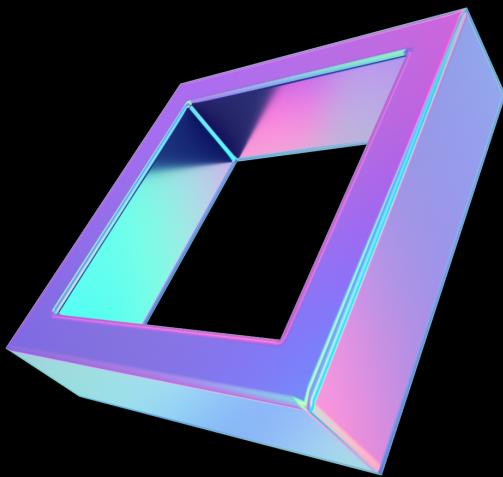
1.WORKS

Project works correctly

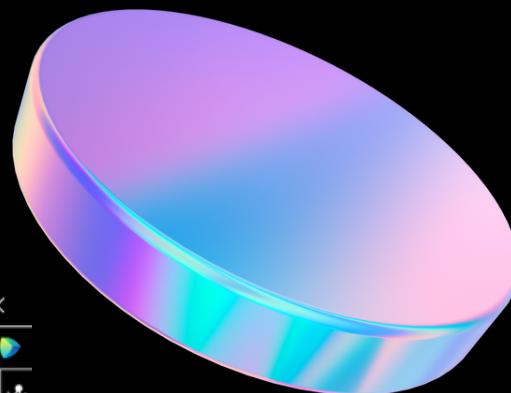


The screenshot shows a Java IDE interface with the following details:

- Top Bar:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help, ware-house-app [D:\Desktop\ware-house-app] - ProductController.java [ware-house-app.main]
- Project Explorer:** Shows a project structure with packages like `AuthController`, `CategoryController`, and `ProductController` under `com.example.warehouseapp.controller`.
- Code Editor:** The `ProductController.java` file is open, showing code related to product requests and category mapping.
- Database Explorer:** A database connection to `postgres@localhost [2]` is shown, with the `public` schema selected.
- Console:** The output of the application startup is displayed, showing logs from `StartupInfoLogger`, `SpringApplication`, `RepositoryConfigurationDelegate`, and other components. It includes the Spring Boot version (v3.2.1) and various initialization steps for the application.
- Bottom Status Bar:** Shows the command bar with Version Control, Run, TODO, Problems, Terminal, Services, Profiler, Database Changes, and Build buttons. It also displays a message about synchronization and a system status bar with date, time, and battery information.



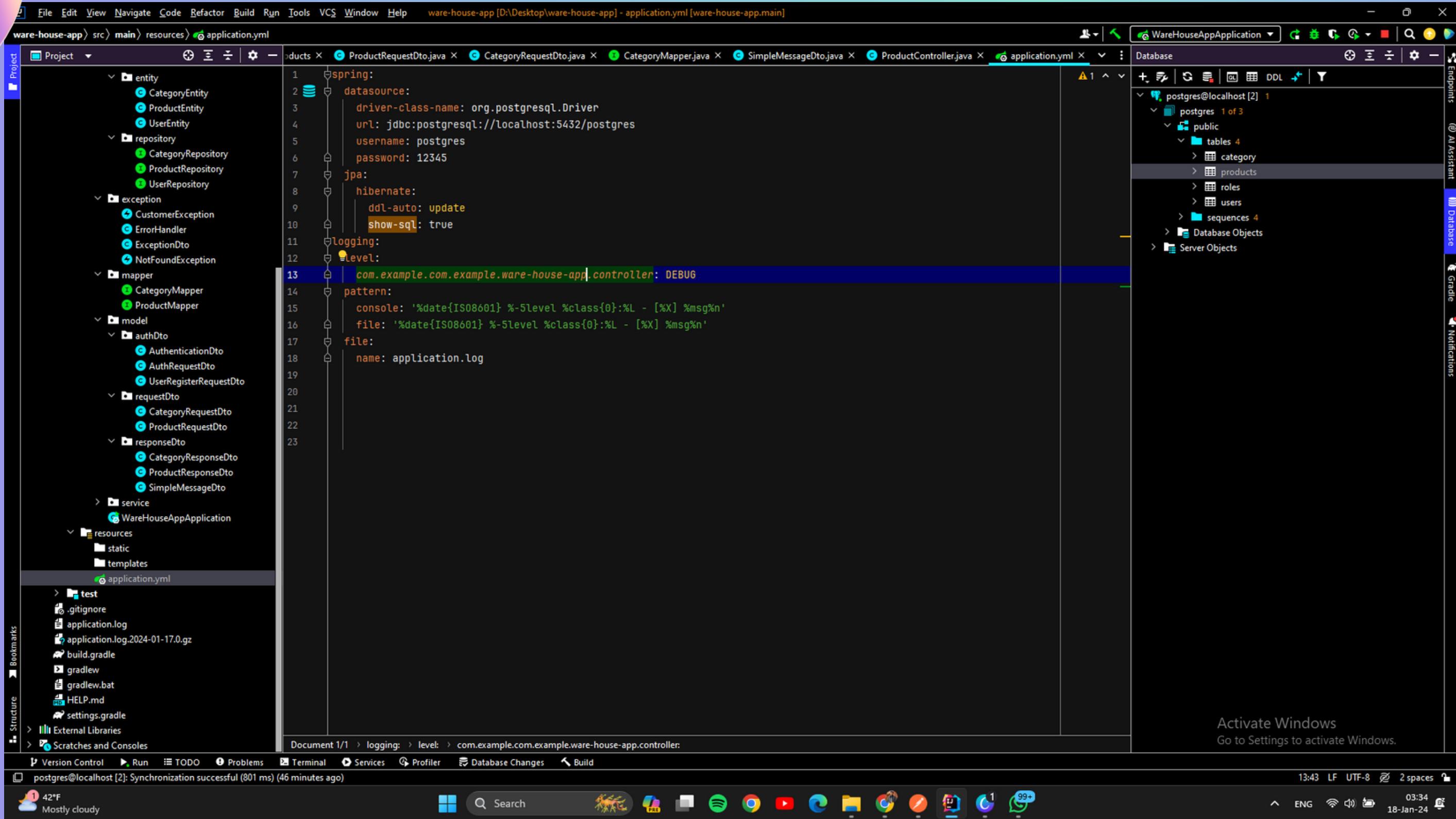
2. CLEAN STRUCTURE



The screenshot shows a Java development environment with the following details:

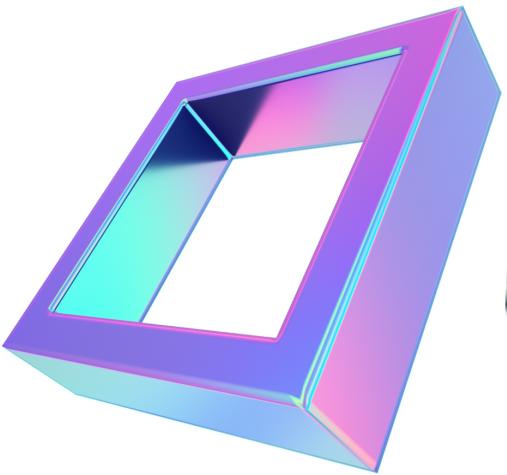
- Project Explorer:** Shows the project structure under `com.example.warehouseapp`. The `ProductController` class is selected.
- Code Editor:** Displays the `ProductController.java` file. The code implements a REST controller for products, using `@RestController` and `@RequestMapping` annotations. It interacts with a `productService` and handles requests for adding products, getting products, checking product counts, and updating products.
- Database Explorer:** Shows the database schema for `postgres@localhost [2]`. It includes tables for `category`, `products`, `roles`, and `users`, along with sequences and various database objects.
- Bottom Bar:** Includes tabs for Version Control, Run, TODO, Problems, Terminal, Services, Profiler, Database Changes, and Build. A message at the bottom right says "Activate Windows Go to Settings to activate Windows."
- System Tray:** Shows icons for battery level (42%), weather (Cloudy), and system status.

3. DB CONNECTIONS



The screenshot shows a Java IDE interface with the following details:

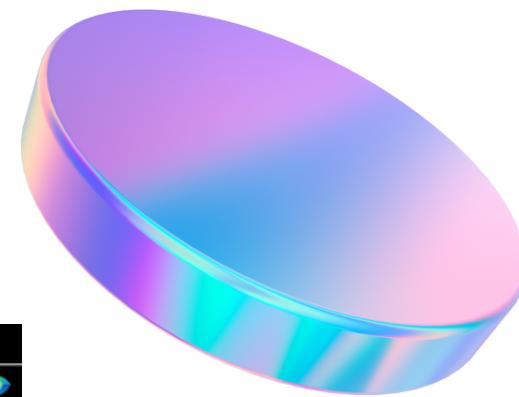
- Project Structure:** The project structure on the left includes entities like CategoryEntity, ProductEntity, and UserEntity; repositories like CategoryRepository, ProductRepository, and UserRepository; exception classes like CustomerException, ErrorHandler, ExceptionDto, and NotFoundException; mappers like CategoryMapper and ProductMapper; and models like authDto, requestDto, responseDto, and service classes.
- application.yml Configuration:** The main configuration file is displayed in the center. It defines a `spring` section with a `datasource` (PostgreSQL driver, URL, username, password), `jpa` (hibernate ddl-auto update, show-sql true), `logging` (level DEBUG, pattern for console and file), and `file` (name application.log).
- Database Browser:** On the right, a database browser shows a connection to `postgres@localhost [2]`. It lists tables such as `category`, `products`, `roles`, and `users` under the `public` schema.
- IDE Status:** At the bottom, there are tabs for Version Control, Run, TODO, Problems, Terminal, Services, Profiler, Database Changes, and Build. A message indicates "Document 1/1 > logging: > level: > com.example.com.example.ware-house-app.controller:"
- System Status:** The bottom bar shows system information including weather (42°F, Mostly cloudy), system status (Synchronization successful), and system time (13:43, 18-Jan-24).



NEW WARE
NIHAD MAHARRAMOV

4.CRUD

Create, Read, Update, Delete



```
ware-house-app > src > main > java > com > example > warehouseapp > service > ProductService.java [ware-house-app.main]
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help ware-house-app [D:\Desktop\ware-house-app] - ProductService.java [ware-house-app.main]

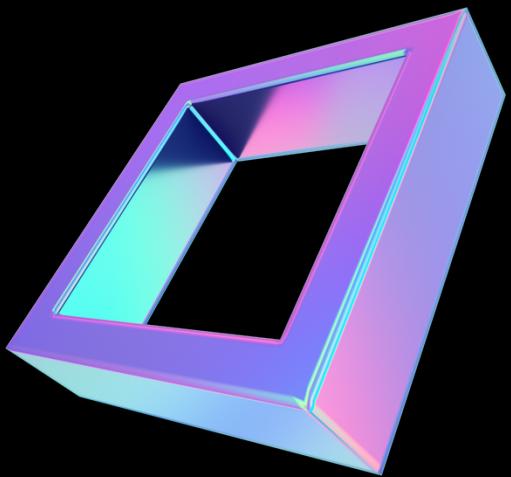
Project Database
+ postgres@localhost [2] 1
  + public 1 of 3
    + tables 4
      + category
      + products
      + roles
      + users
    + sequences 4
    + Database Objects
    + Server Objects

WareHouseAppApplication Database
Database Objects
Server Objects

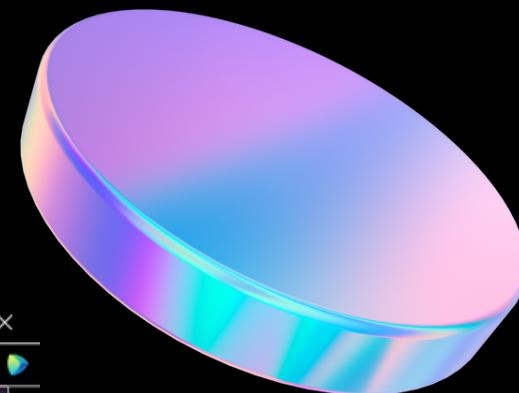
ProductService.java
private final ProductMapper productMapper;
public ProductService(ProductRepository productRepository, ProductMapper productMapper) {
    this.productRepository = productRepository;
    this.productMapper = productMapper;
}
public void addProduct(ProductRequestDto productRequestDto) {
    log.info("ActionLog.start product add method ");
    var id=productRequestDto.getCategoryId();
    String base64 = Base64.getEncoder().encodeToString(String.valueOf(id).getBytes());
    ProductEntity productEntity = productMapper.dtoToEntity(productRequestDto);
    productEntity.setQrCode(base64);
    productRepository.save(productEntity);
    log.info("ActionLog.end product add method ");
}
public List<ProductResponseDto> getProducts() {
    log.info("ActionLog.start product get method ");
    var result = productMapper.entityToDtos(productRepository.findAll());
    log.info("ActionLog.end product get method ");
    return result;
}
public ProductResponseDto getProduct(Long id) {
    log.info("ActionLog.start product get method with id: {}", id);
    ProductEntity productEntity = productRepository.findById(id).orElseThrow(() -> {
        throw new NotFoundException("PRODUCT_NOT_FOUND");
    });
    log.info("ActionLog.end product get method with id: {}", id);
}

Activate Windows
Go to Settings to activate Windows.

Version Control Run TODO Problems Terminal Services Profiler Database Changes Build
postres@localhost [2]: Synchronization successful (801 ms) (52 minutes ago) 38:56 CRLF UTF-8 4 spaces
1 42°F Mostly cloudy Search Spotify YouTube Chrome File 1 99+
```



5. Logging



The screenshot shows a Java IDE interface with multiple panes. The top navigation bar includes File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help, and a central title bar for 'ware-house-app [D:\Desktop\ware-house-app] - application.log [ware-house-app]'. The left sidebar has sections for Project, Bookmarks, and Structure. The main area has tabs for Project, application.java, application.yml, CustomerException.java, CategoryService.java, ProductService.java, application.log (which is active), application.log.2024-01-17.0.gz, and application.log.2024-01-17.0.gz. The right side features a Database pane connected to 'WareHouseAppApplication' on 'postgres@localhost [2]' with 1 connection, showing tables like category, products, roles, and users. A status bar at the bottom shows 'Activate Windows' and system information.

Project

ware-house-app > application.log

application.java X application.yml X CustomerException.java X CategoryService.java X ProductService.java X application.log X application.log.2024-01-17.0.gz X application.log.2024-01-17.0.gz X

Database

WareHouseAppApplication

PostgreSQL

Endpoints

Assistants

Database Grid

Notifications

Activate Windows
Go to Settings to activate Windows.

Version Control Run TODO Problems Terminal Services Profiler Database Changes Build

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help ware-house-app [D:\Desktop\ware-house-app] - application.log [ware-house-app]

42°F Mostly cloudy

Search

Cloudinary

Spotify

YouTube

Microsoft Edge

Google Chrome

File

ENG

03:47

18-Jan-24



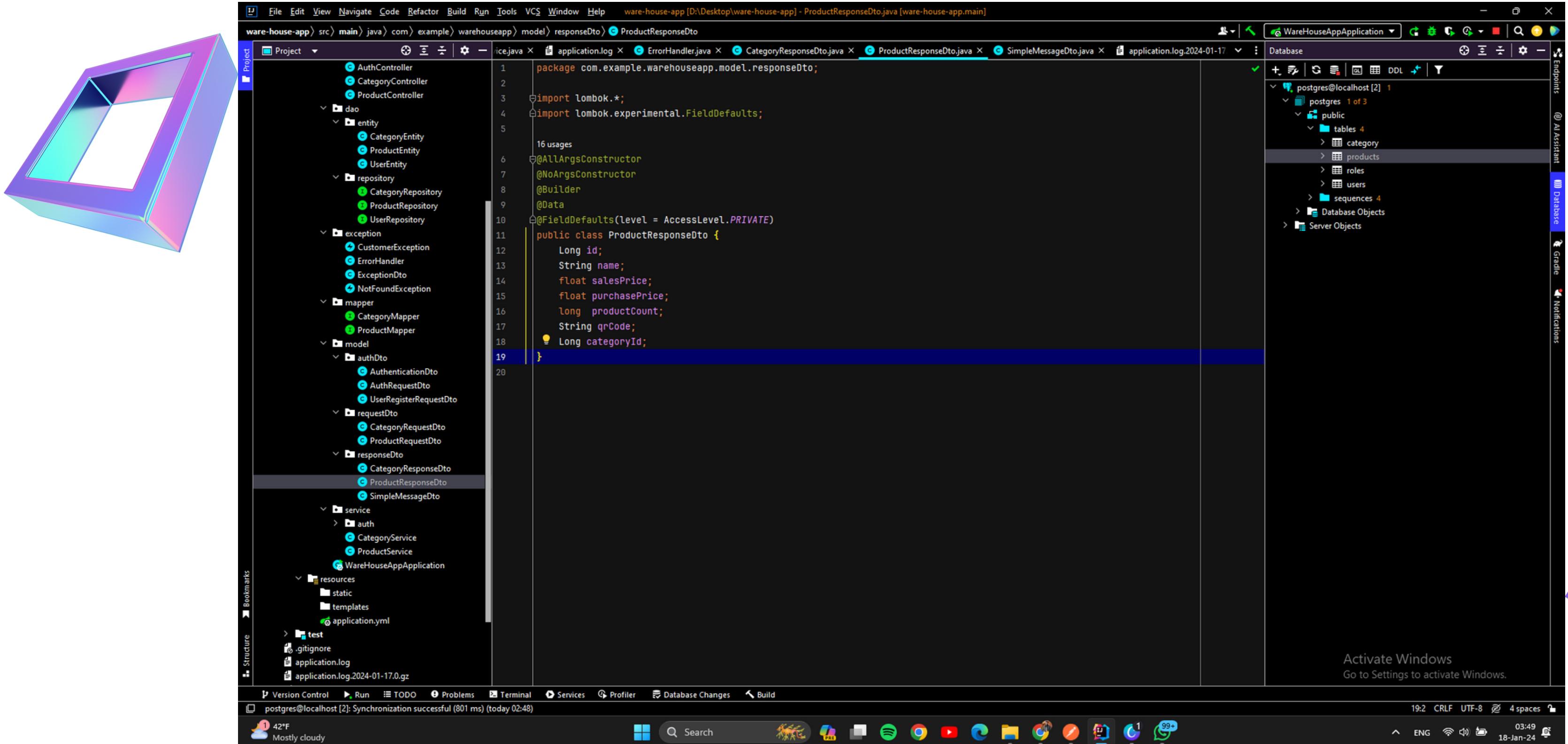
6. ERROR HANDLING

The screenshot shows a Java development environment with the following details:

- Project Structure:** The project is named "ware-house-app". It contains several packages: config, controller, dao, exception, mapper, model, service, and resources.
- Open File:** The file "ErrorHandler.java" is open in the editor. It is annotated with `@RestControllerAdvice` and handles various exception types using `@ExceptionHandler`. The code includes methods for handling `NotFoundException`, `CustomerException`, `AuthenticationException`, `MethodArgumentNotValidException`, and `ConstraintViolationException`.
- Database:** A database connection to "postgres@localhost [2]" is established. The schema includes tables for category, products, roles, and users, along with sequences and other objects.
- Toolbars and Status:** The bottom of the screen features toolbars for Version Control, Run, TODO, Problems, Terminal, Services, Profiler, Database Changes, and Build. The status bar at the bottom right shows the date as 18-Jan-24 and the time as 03:48.

7. STRUCTURED RESPONSE

Create, Read, Update, Delete



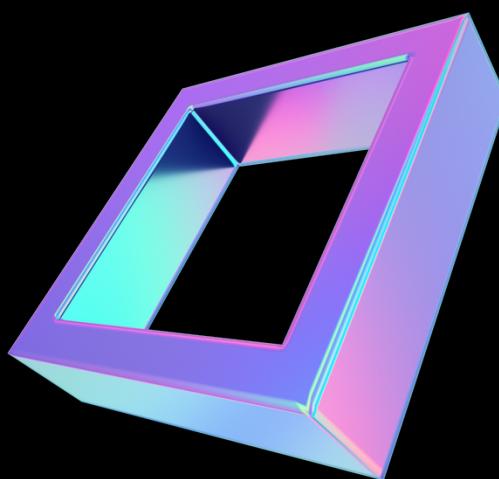
The screenshot shows a Java development environment with the following details:

- Project Structure:** The project is named "ware-house-app". The `src/main/java/com/example/warehouseapp/model/responseDto` package contains the `ProductResponseDto.java` file.
- Code Editor:** The `ProductResponseDto.java` file is open, showing the following code:

```
package com.example.warehouseapp.model.responseDto;

import lombok.*;  
import lombok.experimental.FieldDefaults;  
  
@FieldDefaults(level = AccessLevel.PRIVATE)  
public class ProductResponseDto {  
    Long id;  
    String name;  
    float salesPrice;  
    float purchasePrice;  
    long productCount;  
    String qrCode;  
    Long categoryId;  
}
```
- Database View:** On the right, the "Database" tool window shows a connection to "postgres@localhost [2]". It displays the schema structure, including the `public` schema with tables `category` and `products`.
- System Tray:** At the bottom left, it shows the weather as 42°F and mostly cloudy. At the bottom right, it shows system status like battery level (99%), network, and date (18-Jan-24).

8. CUSTOM ADDRESS/PORT



ware-house-app [D:\Desktop\ware-house-app] - application.yml [ware-house-app.main]

```
ware-house-app> main> resources> application.yml
```

Project Database Endpoints AI Assistant Database Gradle Notifications

```
spring:
  datasource:
    driver-class-name: org.postgresql.Driver
    url: jdbc:postgresql://localhost:5432/postgres
    username: postgres
    password: 12345
  jpa:
    hibernate:
      ddl-auto: update
      show-sql: true
  logging:
    level:
      com.example.com.example.ware-house-app.controller: DEBUG
    pattern:
      console: '%date{ISO8601} %-5level %class{0}:%L - [%X] %msg%n'
      file: '%date{ISO8601} %-5level %class{0}:%L - [%X] %msg%n'
    file:
      name: application.log
```

Document 1/1 > spring: > datasource:

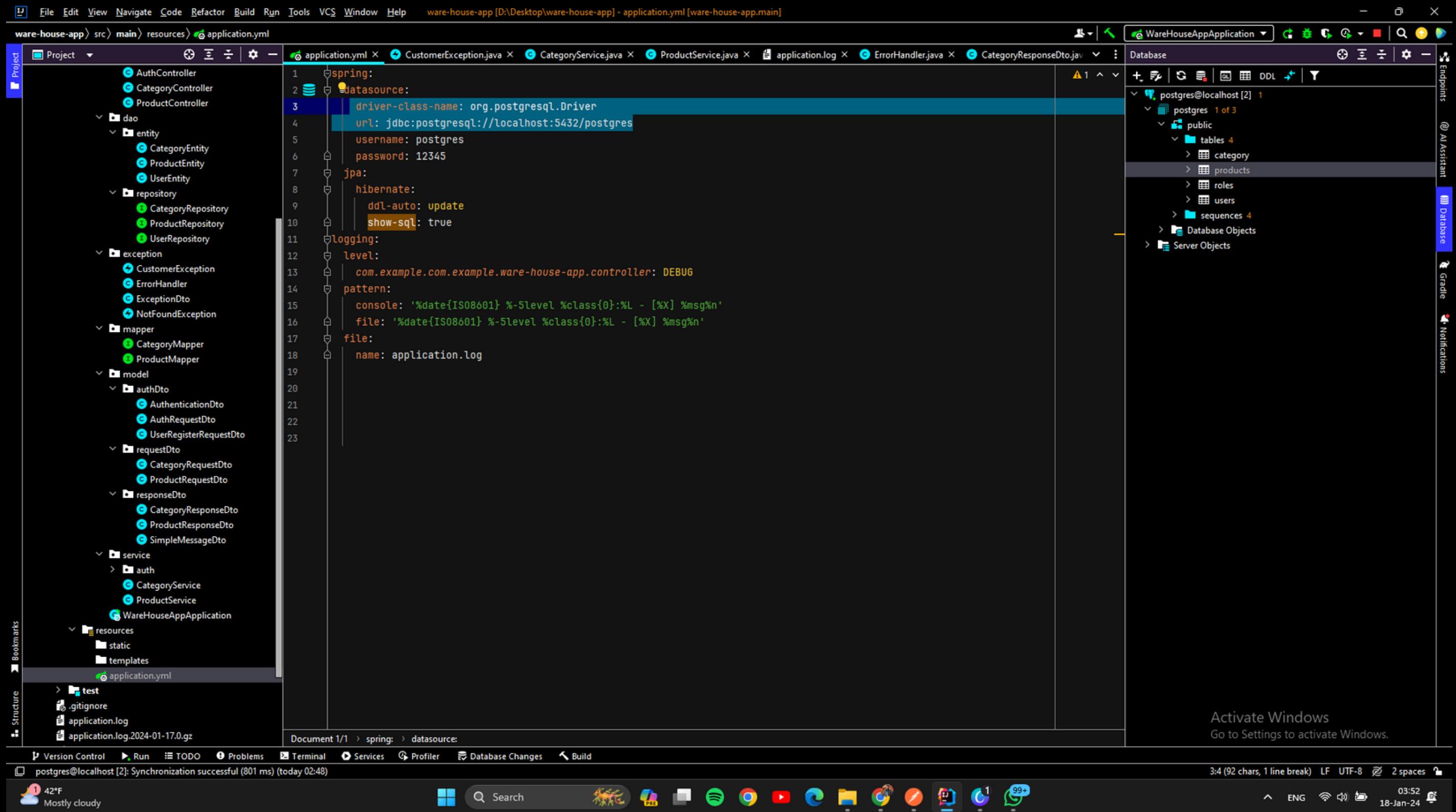
Activate Windows
Go to Settings to activate Windows.

Version Control Run TODO Problems Terminal Services Profiler Database Changes Build

postgres@localhost [2]: Synchronization successful (801 ms) (today 02:48) 3:4 (92 chars, 1 line break) LF UTF-8 2 spaces

42°F Mostly cloudy Search

18-Jan-24 03:52 ENG





9. USING METHODS

GET/POST/PUT/DELETE

```
public ProductController(ProductService productService) {this.productService = productService;}

@PostMapping
public SimpleMessageDto<ProductRequestDto> addProduct(@Valid @RequestBody ProductRequestDto categoryRequestDto) {
    productService.addProduct(categoryRequestDto);
    return new SimpleMessageDto<>(HttpStatus.CREATED.getReasonPhrase(), HttpStatus.CREATED.value());
}

@GetMapping
public SimpleMessageDto<List<ProductResponseDto>> getProducts() {
    return new SimpleMessageDto<>(HttpStatus.OK.getReasonPhrase(), HttpStatus.OK.value(), productService.getProducts());
}

@GetMapping("/{id}")
public SimpleMessageDto<ProductResponseDto> getProduct(@PathVariable Long id) {
    return new SimpleMessageDto<>(HttpStatus.OK.getReasonPhrase(), HttpStatus.OK.value(), productService.getProduct(id));
}

@PostMapping("/{base64}")
public SimpleMessageDto<String> checkCount(@PathVariable String base64) {
    productService.checkCount(base64);
    return new SimpleMessageDto<>(HttpStatus.CREATED.getReasonPhrase(), HttpStatus.CREATED.value());
}

@PutMapping("/{id}")
public SimpleMessageDto<ProductRequestDto> updateProduct(@PathVariable Long id, @Valid @RequestBody ProductRequestDto productRequestDto) {
    productService.updateProduct(productRequestDto, id);
    return new SimpleMessageDto<>(HttpStatus.OK.getReasonPhrase(), HttpStatus.OK.value());
}

@DeleteMapping("/{id}")
public SimpleMessageDto<String> deleteProduct(@PathVariable Long id) {
    productService.deleteProduct(id);
    return new SimpleMessageDto<>(HttpStatus.OK.getReasonPhrase(), HttpStatus.OK.value());
}
```

Activate Windows
Go to Settings to activate Windows.

42°F Mostly cloudy

45.29 CRLF UTF-8 4 spaces

ENGLISH 03:54 18-Jan-24

10. SWAGGER UI

localhost:8080/swagger-ui/index.html#/

product-controller

- GET /v1/product/{id}
- PUT /v1/product/{id}
- DELETE /v1/product/{id}
- GET /v1/product
- POST /v1/product
- POST /v1/product/{base64}

category-controller

- GET /v1/category/{id}
- PUT /v1/category/{id}
- DELETE /v1/category/{id}
- GET /v1/category
- POST /v1/category

auth-controller

- POST /v1/auth/register
- POST /v1/auth/login

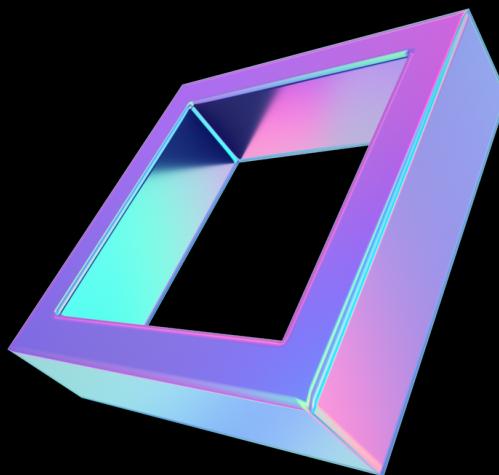
Activate Windows
Go to Settings to activate Windows.

42°F Mostly cloudy

Search

03:55 18-Jan-24

11. SECURITY



ware-house-app [D:\Desktop\ware-house-app] - SecurityConfiguration.java [ware-house-app.main]

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help ware-house-app [D:\Desktop\ware-house-app] - SecurityConfiguration.java [ware-house-app.main]

ware-house-app > src > main > java > com > example > warehouseapp > config > SecurityConfiguration > securityFilterChain > Lambda

Project Database

application.yml ProductController.java AuthService.java SecurityConfiguration.java JwtService.java application.log ErrorHandler.java

ware-house-app D:\Desktop\ware-house-app .gradle .idea build gradle src main java com.example.warehouseapp config ApplicationConfig JwtAuthenticationFilter SecurityConfiguration SwaggerConfig controller AuthController CategoryController ProductController dao entity CategoryEntity ProductEntity UserEntity repository CategoryRepository ProductRepository UserRepository exception CustomerException ErrorHandler ExceptionDto NotFoundException mapper CategoryMapper ProductMapper model authDto AuthenticationDto AuthRequestDto UserRegisterRequestDto requestDto CategoryRequestDto ProductRequestDto responseDto CategoryResponseDto

13 import org.springframework.security.provisioning.InMemoryUserDetailsManager;
14 import org.springframework.security.web.SecurityFilterChain;
15 import org.springframework.security.web.authentication.UsernamePasswordAuthenticationFilter;
16
17 @Configuration
18 @EnableWebSecurity
19 @RequiredArgsConstructor
20 public class SecurityConfiguration {
21
22 private final JwtAuthenticationFilter jwtAuthFilter;
23 private final AuthenticationProvider authenticationProvider;
24
25 @Bean
26 public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
27 http.csrf(csrf -> csrf.disable())
28 .sessionManagement(session -> session.sessionCreationPolicy(SessionCreationPolicy.STATELESS))
29 .authorizeHttpRequests(auth ->
30 auth.requestMatchers(permitSwagger).permitAll()
31 .requestMatchers(v1AuthMatchers).authenticated()
32 .requestMatchers(v1Matchers).authenticated()
33 .anyRequest().authenticated());
34
35 http.authenticationProvider(authenticationProvider);
36 http.addFilterBefore(jwtAuthFilter, UsernamePasswordAuthenticationFilter.class);
37
38 return http.build();
39 }
40
41 @Bean
42 public UserDetailsService userDetailsService() {
43 UserDetails user =
44 User.withDefaultPasswordEncoder()
45 .username("user")
46 .password("password")
47 .roles("ADMIN")
48 .build();
49
50 return new InMemoryUserDetailsManager(user);
51 }
52 public static String[] permitSwagger = {
53 "/api/v1/auth/**",

Activate Windows
Go to Settings to activate Windows.

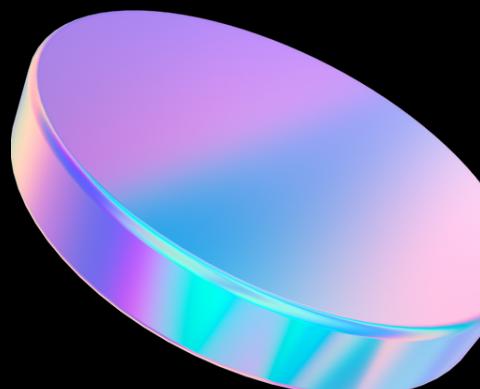
Version Control Run TODO Problems Terminal Services Profiler Database Changes Build

postgres@localhost [2]: Synchronization successful (801 ms) (today 02:48)

30:33 (285 chars, 3 line breaks) CRLF UTF-8 4 spaces

42°F Mostly cloudy Search Spotify Chrome YouTube Edge Google Sheets 1 99+ 03:57 ENG 18-Jan-24

The screenshot shows a Java Spring Security configuration file named SecurityConfiguration.java. The code defines a SecurityFilterChain that disables CSRF, sets session creation policy to stateless, and authorizes requests based on their paths. It uses a JWT authentication filter and an in-memory user details manager. A static array permitSwagger contains the path '/api/v1/auth/**'. The IDE interface includes a Project tool window, a Database tool window showing a PostgreSQL database structure, and various navigation and search tools at the bottom.



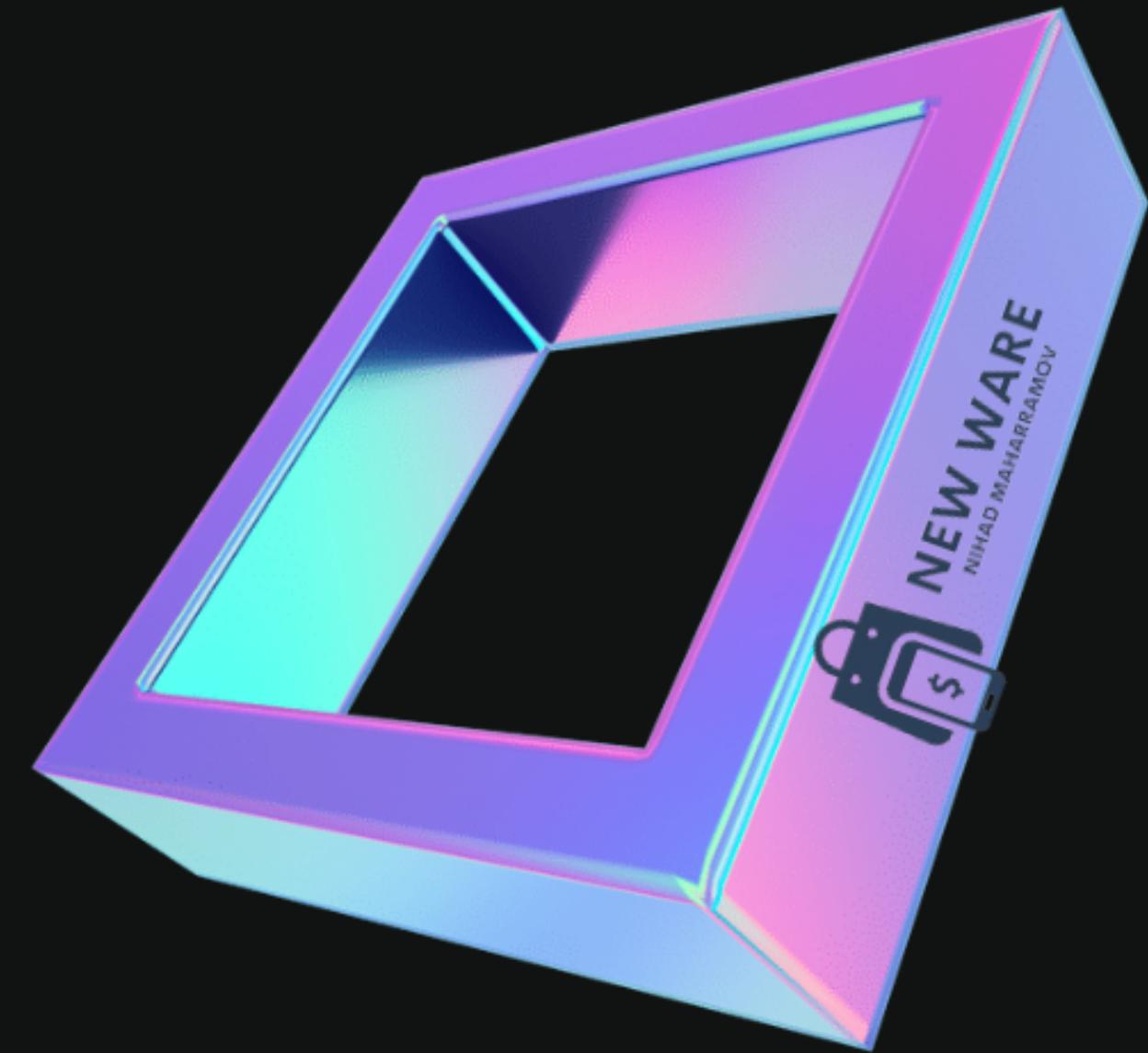


NEW WARE
NIHAD MAHARRAMOV

12. PRESENTATION

NEW WARE

Nihad Maharramov (INHA IT-3)



01

13. GIT COMMITS



Screenshot of a GitHub repository page for "prj-warehouse-count-system-app-nihadishi-1". The repository is private and was created by GitHub Classroom.

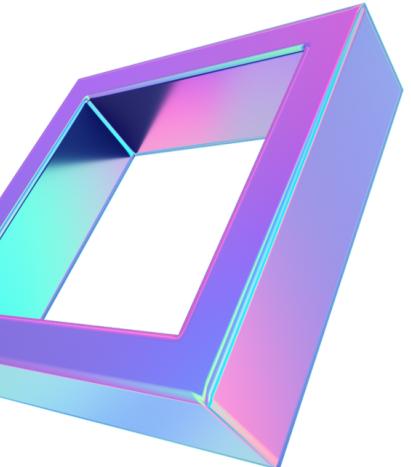
The main repository page shows the following commit history:

Author	Commit Message	Date
nihadishi	bug fix	f4aae09 · 11 minutes ago
	project structure	3 months ago
	permit all	31 minutes ago
	project structure	54 minutes ago
	bug fix	11 minutes ago
	bug fix	11 minutes ago
	project structure	3 months ago
	project structure	3 months ago
	project structure	3 months ago
	project structure	3 months ago

The repository also contains a README file and several configuration files like gradle/wrapper, .gitignore, application.log, build.gradle, gradlew, gradlew.bat, and settings.gradle.

On the right side of the page, there are sections for About, Releases, Packages, Languages, and Suggested workflows.

At the bottom of the screen, there is a taskbar with various icons and a system tray showing the date and time.





I'm ready to work with you!

Thank you!

PHONE

077-637-38-38

EMAIL

nmeherremov1@std.beu.edu.az

WEBSITE

www.nihadishi.vercel.app