Natalia Palej

Software Design With Artificial Intelligence For Cloud Computing  
Year 3 Semester 1

Web Development 3.1

Sprint #1 Project

Student ID: A00279259

Contents

[User Stories 2](#_Toc148610543)

[Testing 3](#_Toc148610544)

[Test Report 5](#_Toc148610545)

[Bugs Report 6](#_Toc148610546)

# User Stories

## Database/Makeup User Stories

1. As a Makeup CAO, I want to connect server and database, so that I can access my inventory.
2. As a Makeup CAO, I want to read data in JSON format for all makeup products, so that I can use it in my application.
3. As a Makeup CAO, I want to read data in JSON format for one makeup product based on its ID, so that I can use it in my application.
4. As a Makeup CAO, I want to insert new makeup product into database, so that I can expand on my inventory.
5. As a Makeup CAO, I want to update product details in database based on its ID, so that I can keep accurate information.
6. As a Makeup CAO, I want to delete product from database based on its ID, so that I can remove discontinued products.

## User User Stories

1. As a User, I want to log in, so that I can have access to the database.
2. As a Makeup CAO, I want to create new user, so that I can have more clients.
3. As a Makeup CAO, I want to delete existing user, so that I can remove inactive clients.
4. As a Makeup CAO, I want to update existing user details, so that I can keep accurate details.
5. As a Makeup CAO, I want to read user details, so that I can use it in my application.

# Testing

## Database/Makeup Test Cases

|  |  |
| --- | --- |
| User Story | Retrieve data in JSON format for all makeup products |
| Test Case ID | #1 |
| Description | Verify valid URL and table displayed |
| Given | 1. Application server and database is running 2. Postman REST Client Opened |
| When | 1. Set GET request 2. Enter URL <http://localhost:8080/MakeupInventory> 3. Press send |
| Then | 1. HTTP response 200 OK 2. All database entries are displayed in JSON format |
| Test Result | Passed Failed |

|  |  |
| --- | --- |
| User Story | Get API connection |
| Test Case ID | #2 |
| Description | Verify invalid URL |
| Given | 1. Application server and database is running 2. Postman REST Client opened |
| When | 1. Set GET request 2. Enter URL <http://localhost:8080/MyCompany> 3. Press send |
| Then | 1. HTTP response 400 NOT FOUND 2. Error is displayed |
| Test Result | Passed Failed |

|  |  |
| --- | --- |
| User Story | Retrieve data in JSON format for one makeup products by ID |
| Test Case ID | #3 |
| Description | Verify one product is retrieved |
| Given | 1. Application server and database is running 2. Postman REST Client opened |
| When | 1. Set GET request 2. Enter product ID 3. Press send |
| Then | 1. HTTP response 200 OK 2. Makeup product information displayed |
| Test Result | Passed Failed |

|  |  |
| --- | --- |
| User Story | Insert new makeup product into database |
| Test Case ID | #4 |
| Description | Verify product can be added to database |
| Given |  |
| When | 1. Set INSERT request 2. Enter product details 3. Press send |
| Then |  |
| Test Result | Passed Failed |

|  |  |
| --- | --- |
| User Story | Update product details in database |
| Test Case ID | #5 |
| Description | Verify product can be updated based on ID |
| Given |  |
| When | 1. Set UPDATE request based on ID 2. Enter details to update 3. Press send |
| Then |  |
| Test Result | Passed Failed |

|  |  |
| --- | --- |
| User Story | Delete product from database |
| Test Case ID | #6 |
| Description | Verify product can be deleted based on ID |
| Given |  |
| When | 1. Set DELETE request based on ID 2. Press send |
| Then |  |
| Test Result | Passed Failed |

## User Test Cases

|  |  |
| --- | --- |
| User Story | I want to log in |
| Test Case ID | #7 |
| Description | Verify user successfully logged in |
| Given |  |
| When |  |
| Then |  |
| Test Result | Passed Failed |

|  |  |
| --- | --- |
| User Story | I want to log in |
| Test Case ID | #8 |
| Description | Verify user not logged if doesn’t exist in database |
| Given |  |
| When |  |
| Then |  |
| Test Result | Passed Failed |

## 

## Test Report

|  |  |  |
| --- | --- | --- |
| **Test Type** | **Number Passed** | **Number Failed** |
| Manual |  |  |
| Automated |  |  |

## Bugs Report