**Updated Service Endpoints**

1. User Registration

* Endpoint: POST /users/register
* Description: Registers a new user in the application.
* Request Body:

json

Copy code

{

"username": "cfreeman",

"email": "cfreeman@email.com",

"password": "password123"

}

* Response:

json

Copy code

{

"id": 1,

"message": "User registered successfully."

}

* Error Response:

json

Copy code

{

"error": "User registration failed."

}

1. User Login

* Endpoint: POST /users/login
* Description: Authenticates a user and returns session data.
* Request Body:

json

Copy code

{

"username": "cfreeman",

"password": "password123"

}

* Response:

json

Copy code

{

"user\_id": 1,

"message": "Login successful."

}

* Error Response:

json

Copy code

{

"error": "Invalid username or password."

}

1. Add Calculation

* Endpoint: POST /calculations
* Description: Saves a new calculation for the authenticated user.
* Request Body:

json

Copy code

{

"user\_id": 1,

"type": "Ohm's Law",

"voltage": "120",

"current": "10",

"resistance": "12",

"result": "Voltage (V) = 120 V"

}

* Response:

json

Copy code

{

"id": 1,

"message": "Calculation saved successfully."

}

* Error Response:

json

Copy code

{

"error": "Failed to save calculation."

}

1. Get All Calculations

* Endpoint: GET /calculations
* Description: Retrieves all calculations for the authenticated user.
* Response:

json

Copy code

[

{

"id": 1,

"type": "Ohm's Law",

"voltage": "120",

"current": "10",

"resistance": "12",

"result": "Voltage (V) = 120 V",

"calculated\_at": "2024-11-09T12:00:00"

}

]

* Error Response:

json

Copy code

{

"error": "No calculations found."

}

1. Get Last Calculation

* Endpoint: GET /calculations/last
* Description: Retrieves the most recent calculation for the authenticated user.
* Response:

json

Copy code

{

"id": 1,

"type": "Ohm's Law",

"voltage": "120",

"current": "10",

"resistance": "12",

"result": "Voltage (V) = 120 V",

"calculated\_at": "2024-11-09T12:00:00"

}

* Error Response:

json

Copy code

{

"error": "No calculations available."

}

1. Mark Calculation as Favorite

* Endpoint: POST /favorites
* Description: Marks a specific calculation as a favorite.
* Request Body:

json

Copy code

{

"user\_id": 1,

"calculation\_id": 1

}

* Response:

json

Copy code

{

"message": "Calculation marked as favorite."

}

* Error Response:

json

Copy code

{

"error": "Failed to mark as favorite."

}

1. Get All Favorites

* Endpoint: GET /favorites
* Description: Retrieves all favorited calculations for the authenticated user.
* Response:

json

Copy code

[

{

"id": 1,

"calculation\_id": 1,

"favorited\_at": "2024-11-09T12:10:00"

}

]

* Error Response:

json

Copy code

{

"error": "No favorites found."

}

1. Data Export

* Endpoint: GET /export
* Description: Exports the user's calculation history as a CSV file.
* Response:

json

Copy code

{

"message": "Data export initiated."

}

* Error Response:

json

Copy code

{

"error": "Failed to export data."

}

1. Analytics

* Endpoint: GET /analytics
* Description: Provides insights on user activity, such as the most frequently used calculation type.
* Response:

json

Copy code

{

"most\_used\_type": "Ohm's Law",

"calculation\_count": 50

}

* Error Response:

json

Copy code

{

"error": "Unable to retrieve analytics."

}

**Purpose, Implementation, and Interactions**

* **User Management:** The service layer handles user registration, login, and session management, allowing personalized experiences and data security.
* **Calculation Management:** Provides CRUD operations for user calculations, enabling users to save, view, edit, and delete their work.
* **Favorites:** Allows users to mark and retrieve favorite calculations for quick access.
* **Data Export and Analytics:** Enhances user experience by offering export options and activity insights.