SLB Proxy Modeling Platform
User Manual

Date: December 22, 2025

Submitted By: Hung-Jui Chen, Guanghao Xu

Table of Content

1. Introduction

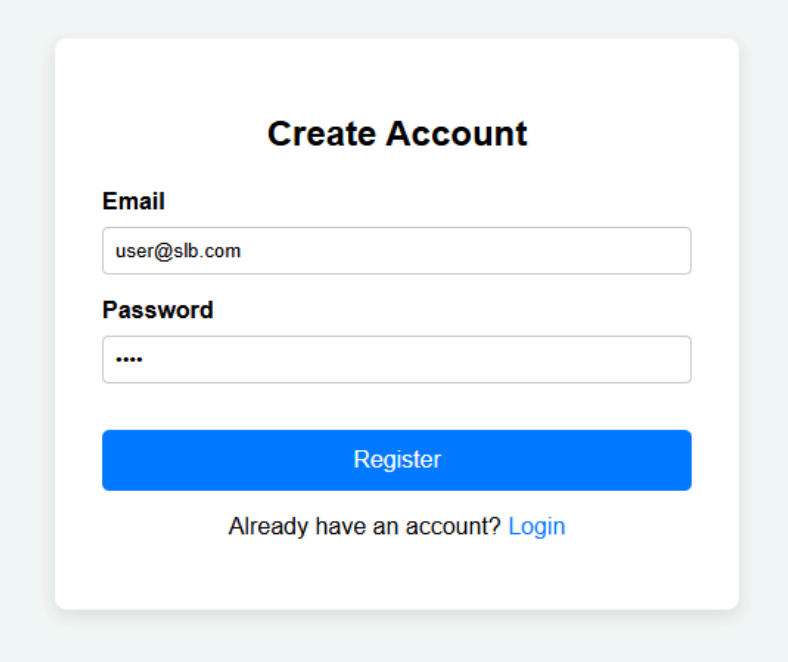
The **SLB Proxy Modeling Platform** is a cloud-based engineering tool designed to replace time-consuming spreadsheet or FEA calculations. It provides instant analysis for Pressure Vessels, allowing engineers to validate designs against **Von Mises** failure criteria in under 0.1 seconds.

2. Access & Authentication

2.1 Registration

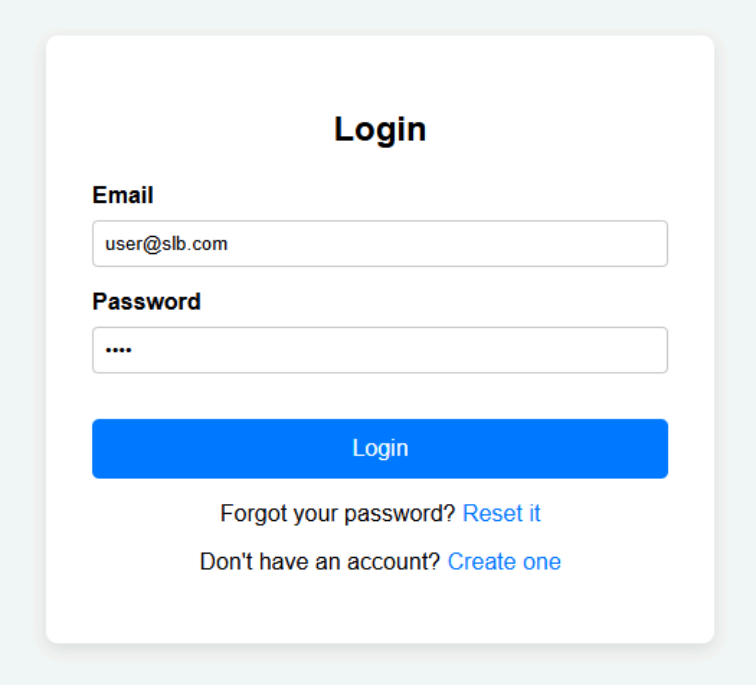
New users must create an account to access the calculation engine.

1. Navigate to the **SLB Proxy Modeling Platform** at: https://red-sea-01a27a51e.3.azurestaticapps.net
2. Click the **"Create one"** link at the bottom of the Login box.
3. Enter a valid email and a secure password.
4. Click **Register**.



2.2 Login

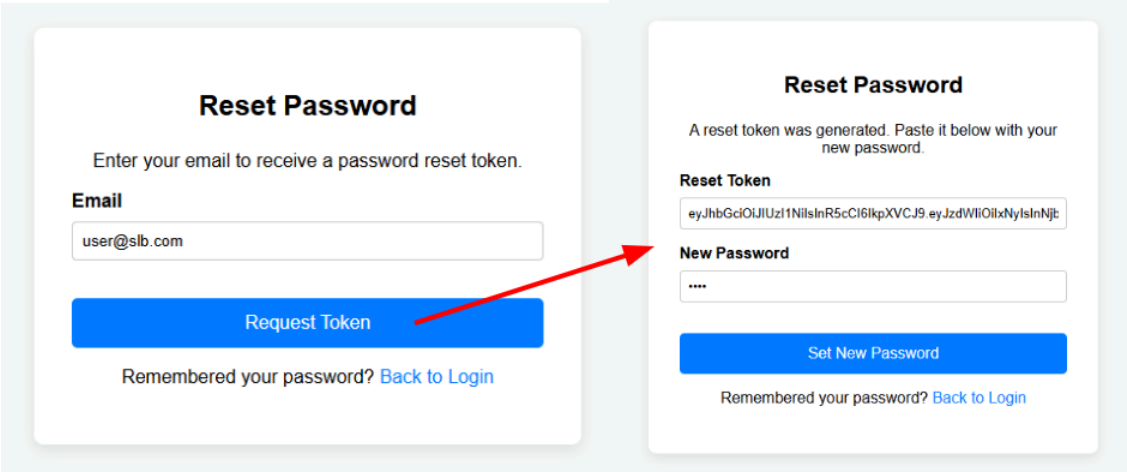
1. Enter your registered Email.
2. Enter your Password.
3. Click **Login** to enter the workspace.



2.3 Password Reset

If you have forgotten your credentials:

1. Click **"Reset it"** on the login screen.
2. Enter your email and click **Request Token**.
3. Copy the token generated by the system.
4. Paste it into the **Reset Token** field along with your **New Password**.

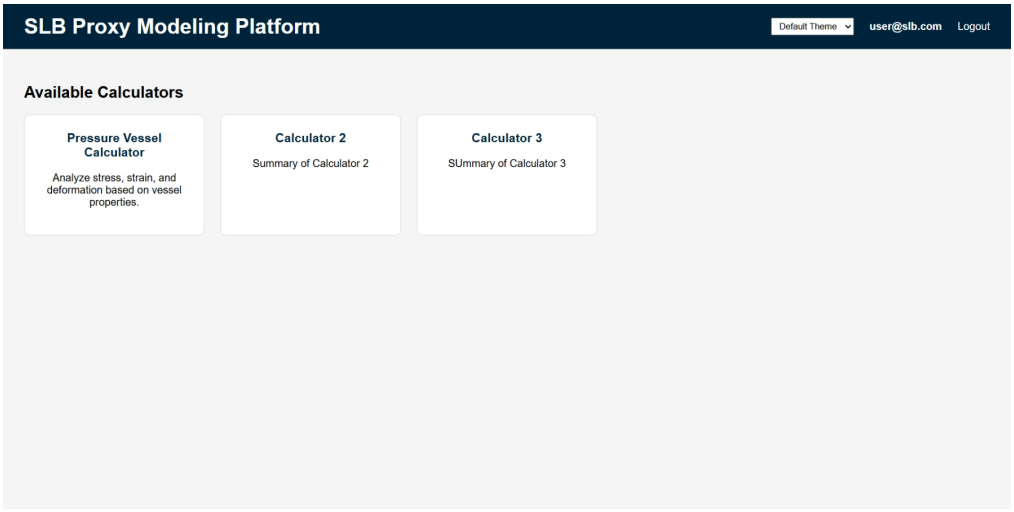


3. The Workspace Interface

3.1 The Landing Page (Dashboard)

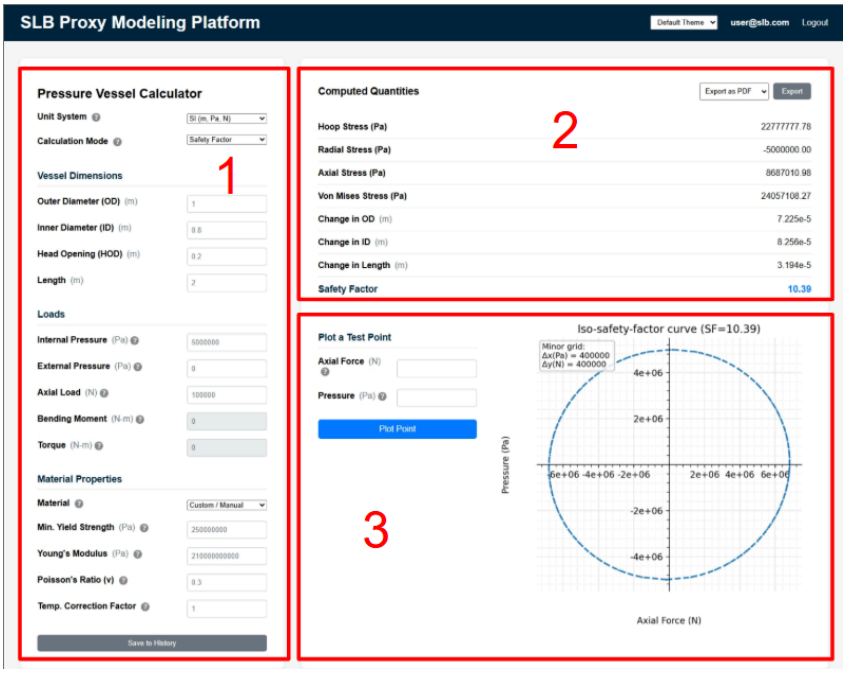
Upon logging in, you are greeted by the **Calculator Dashboard**. This page serves as the central hub for the platform.

- **Calculator Grid:** Displays cards for all available engineering modules (e.g., Pressure Vessel, Calculator 2, etc.).
- **Navigation Bar:** Provides quick access to **User Management** (Admins only), **Theme Settings**, and **Logout**.
- **Action:** Click on the **"Pressure Vessel Calculator"** card to launch the workspace.



3.2 The Calculator Page

Upon logging in, you will see the main **Pressure Vessel Calculator**. The interface is divided into three zones.



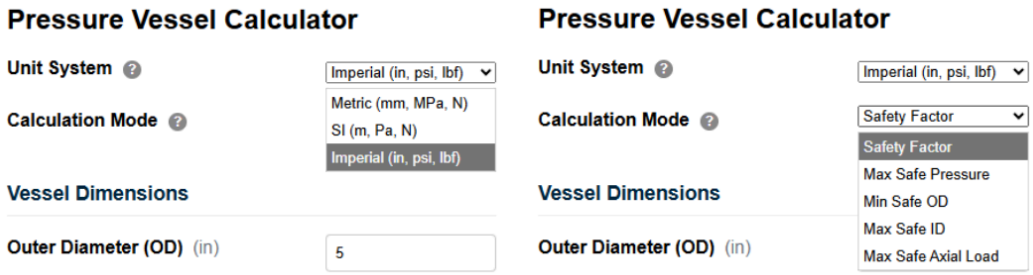
- **Zone 1: Input Panel (Left)** – Where you define units, modes, geometry, and loads. It also contains the **Save to History** button at the bottom for persisting your data.
- **Zone 2: Results Panel (Top right)** – Displays output, calculated Stresses (Hoop, Radial, Axial) and Deformations.
- **Zone 3: Visualization (Bottom right)** – Displays the "Iso-Safety Factor" failure envelope graph.

4. Step-by-Step Instructions

4.1 Setting Units & Mode

Before entering data, define your working environment.

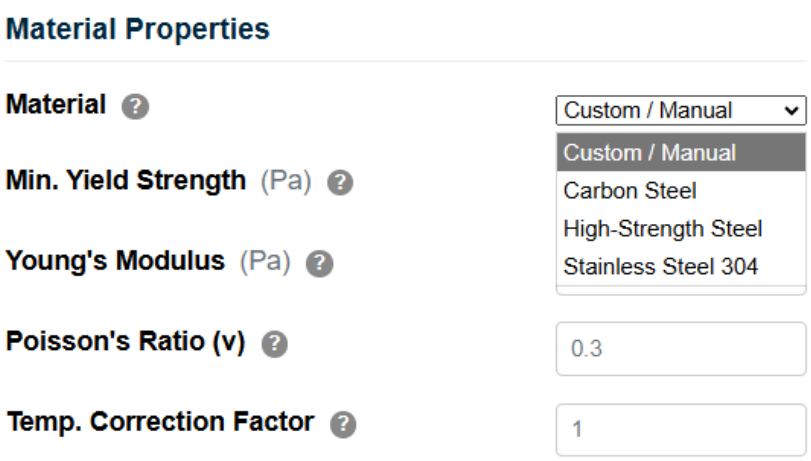
1. **Unit System:** Click the dropdown to select **SI** (m, Pa), **Metric** (mm, MPa), or **Imperial** (in, psi).
2. **Calculation Mode:** Select what you want to solve for (e.g., **Safety Factor**).



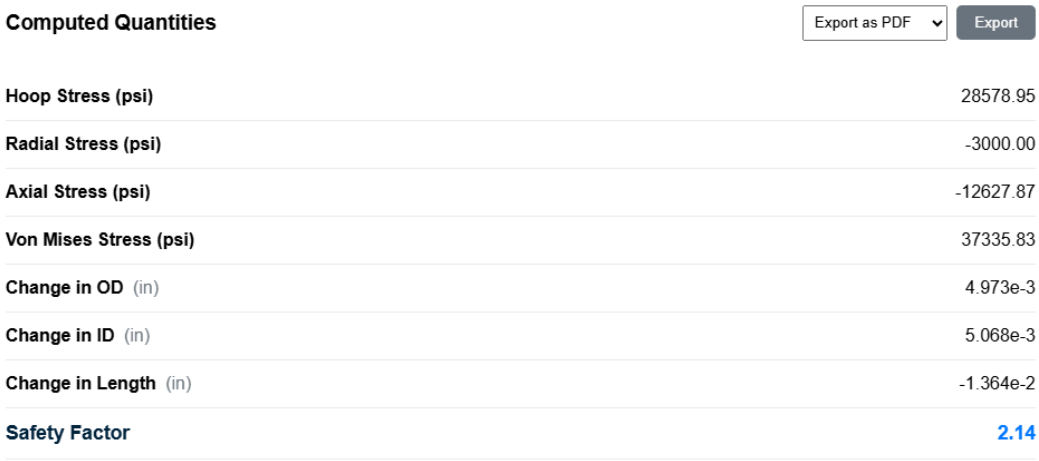
4.2 Calculating Safety Factor (Forward)

Use this mode to check if a specific design is safe.

1. Set Mode to **Safety Factor**.
2. Enter **Vessel Dimensions** (OD, ID, Length).
3. Enter **Loads** (Internal/External Pressure, Axial Load).
4. Enter **Material Properties**.
    - **Material Selection:** Use the **Material** dropdown menu in the "Material Properties" section.
    - **Preset Materials:** Select a standard material (e.g., "Carbon Steel", "Stainless Steel 304") to auto-fill the Yield Strength and Young's Modulus.
    - **Custom:** Select "Custom / Manual" (or simply type in the number fields) to define your own material properties.



5. **Review Results:** The platform features **Real-Time Calculation**. As you type in the input fields, the results in the Center Panel will update automatically (after a brief 300ms pause). You do not need to press a button to see the preview.



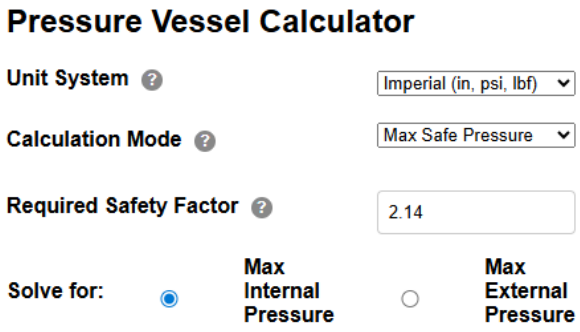
6. **Save to History:** To permanently record this calculation, click the **Save to History** button located at the very bottom of the Input Panel.
    - **Action:** This saves your current inputs and results to the centralized database.
    - **Confirmation:** A browser alert will confirm "Calculation saved to your history!
7. **Data Storage Note:** When you click **Save to History**, the system automatically converts all values into **SI Units** (Meters, Pascals, Newtons) before storing them in the database. This ensures consistency across the organization, regardless of whether you calculated using Metric or Imperial units.

4.3 Inverse Calculation

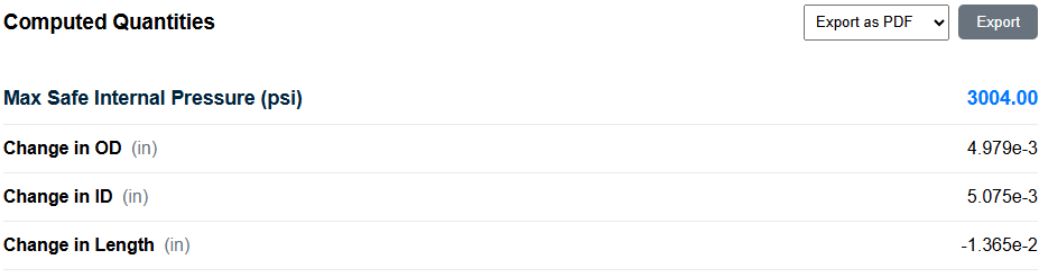
4.3.1 Max Safe Pressure

Use this mode to find the pressure limit for a required safety margin.

1. Set Mode to **Max Safe Pressure**.
2. Enter the **Required Safety Factor** (e.g., 2.14).
3. Select **Solve for: Max Internal/External Pressure**.



4. **Result:** The Center Panel will display the maximum pressure allowed (e.g., 3000 psi) to maintain that safety factor.



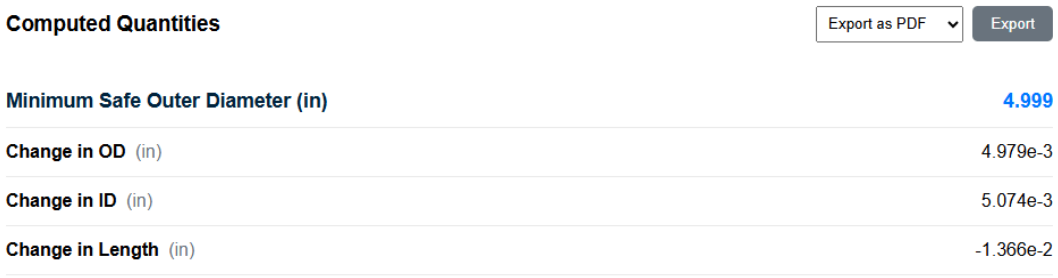
> **Tip:** You can also save these inverse calculation results to the database by clicking **Save to History**. The target pressure or optimized geometry will be stored in SI units.

4.3.2 Optimizing Geometry (Min OD / Max ID)

Use this to find the thinnest possible wall thickness that is still safe.

1. Set Mode to **Min Safe OD** (Minimum Outer Diameter).
2. Enter **Required Safety Factor** (e.g., 2.14).
3. Enter your fixed value (e.g., Inner Diameter, Pressure, Material Properties, etc.).

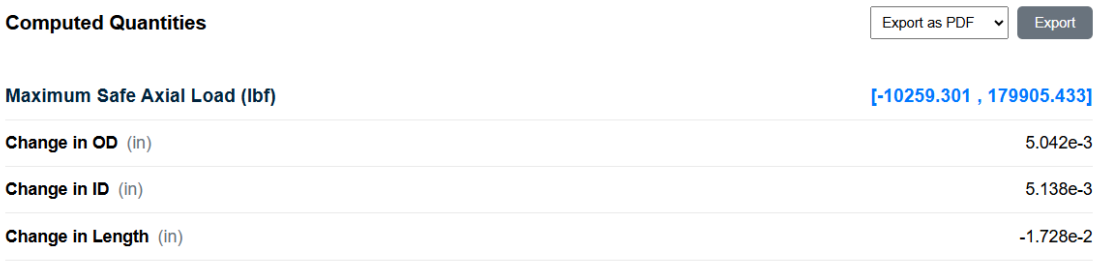
4. **Result:** The system calculates the smallest safe OD.



4.3.3 Max Safe Axial Load

This mode determines the structural limits of the vessel under longitudinal forces, given a fixed pressure. It answers: *"How much can I pull (tension) or push (compression) this vessel before it fails?"*

1. Set **Calculation Mode** to **Max Safe Axial Load**.
2. Enter the **Required Safety Factor** (e.g., 2.14).
3. Enter the existing **Pressure** (Internal or External), Dimensions and Material Properties.
4. **Result:** The system displays a **Safe Load Range** (e.g., [-10,260 lbf, +180,000 lbf]).
    - **Negative Value:** Maximum allowable **Compression**.
    - **Positive Value:** Maximum allowable **Tension**.

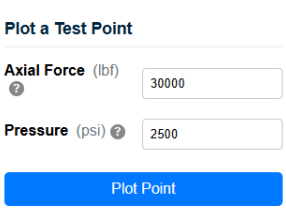


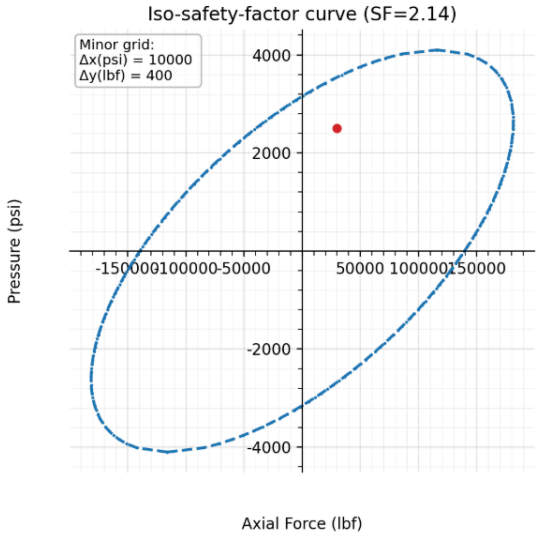
4.4 Visualizing Results (Graphing)

The graph shows the safe operating limits (Failure Envelope).

1. Look at the **Iso-safety-factor curve** on the bottom right.
2. **Test a Point:**
    - Enter an **Axial Force** (e.g., 30000 lbf) and **Pressure** (e.g., 2500 psi) in the "Plot a Test Point" box.

    - Click **Plot Point**.
3. **Interpret:**
    - **Red Dot Inside Loop:** Safe.
    - **Red Dot Outside Loop:** Failure.





5. Exporting Reports

The platform allows you to export your analysis in three different formats, depending on whether you need a visual report or raw data for further processing.

5.1 Export Formats

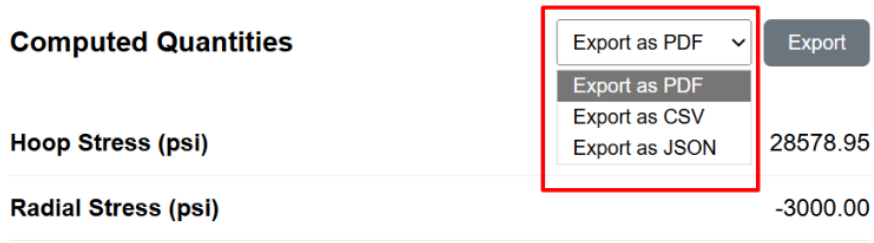
- **PDF Report:** Best for documentation and sharing. It captures a snapshot of the entire workspace, including input tables, calculated results, and the Iso-Safety graph.
- **CSV (Comma-Separated Values):** Best for Excel. It downloads a structured table of all inputs and outputs, allowing you to copy-paste data into your own engineering spreadsheets.
- **JSON (JavaScript Object Notation):** Best for programming. It provides a machine-readable file that can be easily parsed by Python, MATLAB, or other software for automated workflows.

5.2 How to Export

1. Locate the **Export controls** at the top-right of the Results Panel.
2. Click the dropdown menu (default is "Export as PDF").
3. Select your desired format:
    - **Export as PDF**
    - **Export as CSV**

- **Export as JSON**

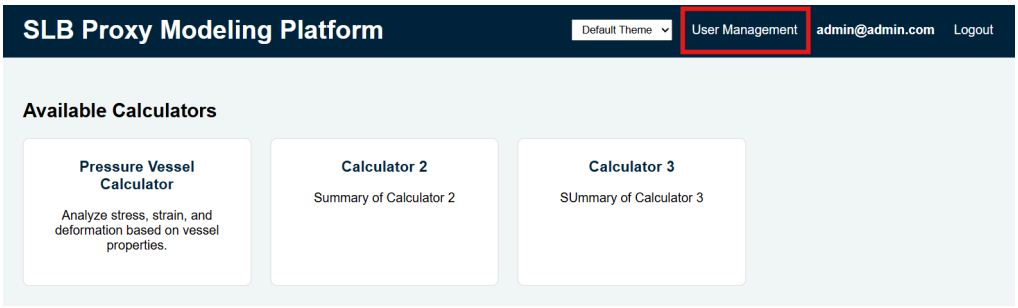
4. Click the **Export** button.
5. The file will automatically download to your computer's default downloads folder (e.g., slb-vessel-report.csv).



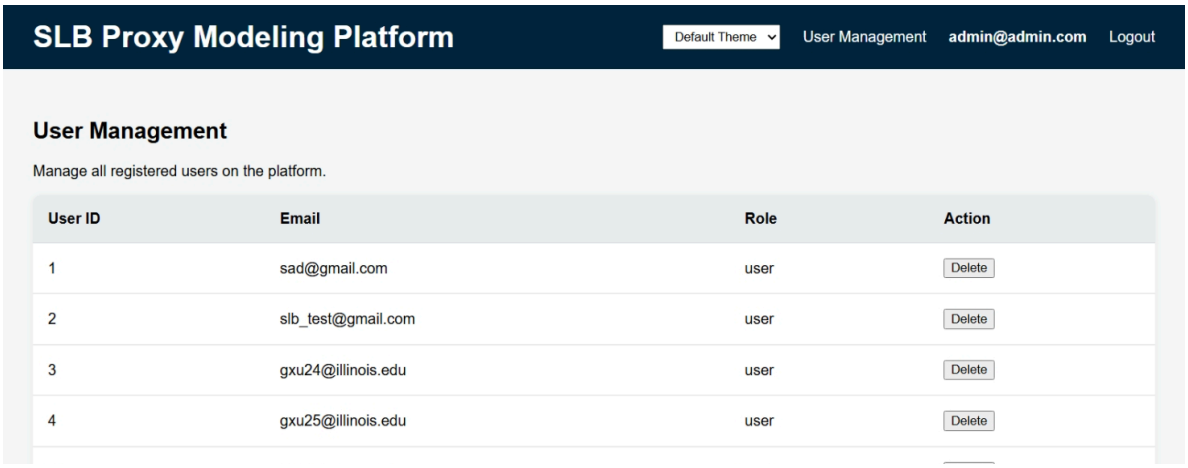
6. Administrator Tools

(Only accessible to accounts with Admin role)

1. Click **User Management** in the top navigation bar.



2. **View Users:** See a list of all registered emails and IDs.



3. **Delete User:** Click the **Delete** button to remove a user and their history. **Warning: This cannot be undone.**

7. Troubleshooting

Issue	Solution
"Outer diameter must be greater than inner diameter"	Check your inputs. The OD cannot be smaller than the ID.
"Inner diameter must be greater than 0"	Check your inputs. The ID cannot be smaller than 0.
Graph does not appear	Ensure "Yield Strength" and "Target Safety Factor" are positive numbers.
Login fails repeatedly	Click "Logout" to clear old sessions, then try logging in again.
Export button not working	Ensure pop-ups are allowed in your browser settings.

8. Advanced Administration & Maintenance

Note: These operations require direct access to the MySQL database server. They should only be performed by IT personnel or authorized system administrators.

8.1 Accessing Azure Database for MySQL Flexible Server

First, download and install MySQL Community Server from the official MySQL website. During the installation process, the MySQL Server package automatically installs the MySQL command-line client (mysql), which is required to interact with MySQL databases from the terminal.

After installation, the client was verified by checking the MySQL version:

```
mysql --version
```

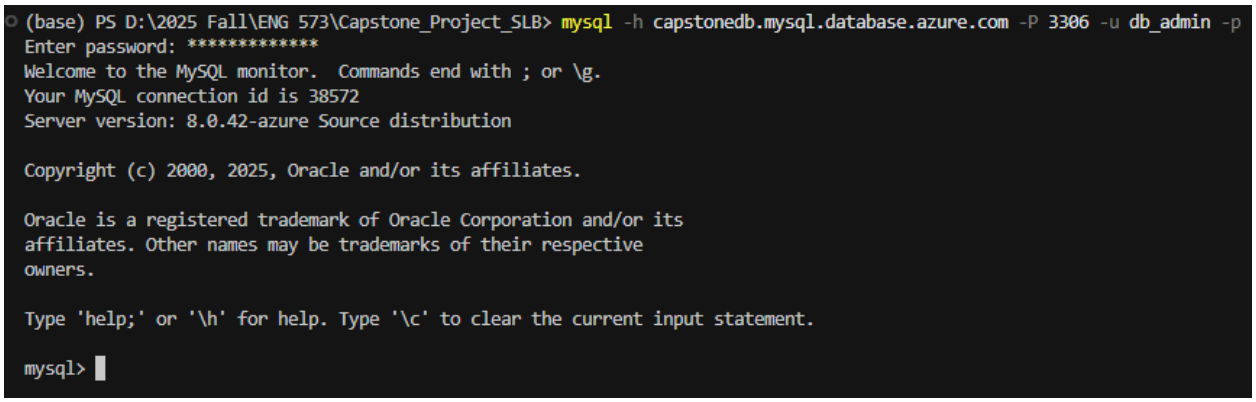
If the version information is displayed, the MySQL command-line client has been successfully installed and configured.

Once the client is available, establish a secure connection to the **Azure Database for MySQL Flexible Server** using the MySQL command-line interface:

```
mysql -h <MySQL server hostname> -P <MySQL port> -u <database administrator
username> -p
```

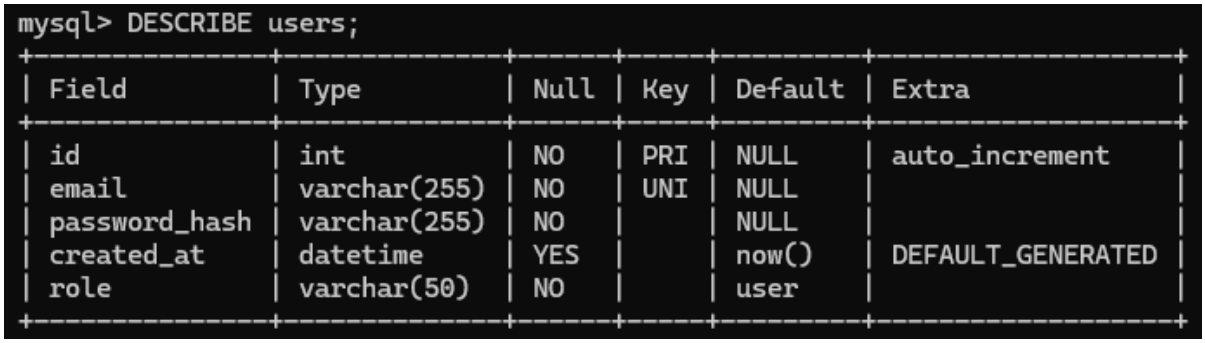
After executing the command, enter the database password when prompted.

If the connection is successful, the MySQL monitor will be displayed, allowing direct interaction with the remote Azure MySQL database.



8.2 promoting a User to Admin

By default, all new registrations are assigned the user role. To grant a user access to the **User Management Dashboard** (Section 6), you must update their role in the database.



SQL Command:

```
UPDATE users

SET role = 'admin'

WHERE email = 'target_user_email@slb.com';
```

Action: Execute this query in your database client. The user must log out and log back in for the change to take effect.

8.3 Managing the Material Library

The **"Material"** dropdown menu in the Calculator Interface is populated dynamically from the materials database table. To add new standard materials to the global list, you must insert them into this table.

```
mysql> DESCRIBE materials;
+-----------------------+--------------+------+-----+---------+-------------------+
| Field                 | Type         | Null | Key | Default | Extra             |
+-----------------------+--------------+------+-----+---------+-------------------+
| id                    | int          | NO   | PRI | NULL    | auto_increment    |
| key                   | varchar(50)  | NO   | UNI | NULL    |                   |
| display_name          | varchar(100) | NO   |     | NULL    |                   |
| yield_strength        | float        | NO   |     | NULL    |                   |
| youngs_modulus        | float        | NO   |     | NULL    |                   |
| poissons_ratio        | float        | NO   |     | NULL    |                   |
| temp_correction_factor | float       | NO   |     | NULL    |                   |
| created_at            | datetime     | YES  |     | now()   | DEFAULT_GENERATED |
+-----------------------+--------------+------+-----+---------+-------------------+
```

To Add a New Material: Execute an SQL INSERT statement with the material's properties (in SI Units: Pascals for Pressure, decimal for Ratio).

SQL Example (Adding "Titanium Alloy"):

```
INSERT INTO materials (
    `key`,
    `display_name`,
    `yield_strength`,
    `youngs_modulus`,
    `poissons_ratio`,
    `temp_correction_factor`
) VALUES (
    'ti_6al_4v',          -- Unique internal key
    'Titanium Ti-6Al-4V', -- Name shown in Dropdown
    880000000,            -- Yield Strength (880 MPa in Pa)
```

```
    113800000000,         -- Young's Modulus (113.8 GPa in Pa)
    0.34,                 -- Poisson's Ratio
    1.0                   -- Temp Factor
);
```

Result: "Titanium Ti-6Al-4V" will immediately appear in the material dropdown for all users.

To Remove a Material:

```
DELETE FROM materials WHERE `key` = 'ti_6al_4v';
```

8.4 Access to Calculation History

All pressure vessel calculation results are stored persistently in the Azure Database for MySQL Flexible Server to support traceability and result verification. Each record represents a single calculation request and includes input parameters, computed stresses, deformation results, safety factors, and a timestamp.

The calculation history was inspected directly using the MySQL command-line interface. The following SQL command was an example used to retrieve the most recent calculation records:

```
SELECT * FROM pressure_vessel_calculations ORDER BY id DESC LIMIT 50\G
```

The **\G** option enables vertical output formatting, which improves readability for tables with many fields by displaying each record as a structured key–value block.

```
mysql> SELECT * FROM pressure_vessel_calculations ORDER BY id DESC LIMIT 50\G
*************************** 1. row ***************************
                    id: 28
               user_id: 3
                  mode: safety_factor
        outer_diameter: 1
        inner_diameter: 0.8
 head_opening_diameter: 0.2
                length: 2
     internal_pressure: 50000000
     external_pressure: 0
           axial_load: 100000
        bending_moment: 0
                torque: 0
        yield_strength: 250000000
        poissons_ratio: 0.3
        youngs_modulus: 210000000000
temp_correction_factor: 1
         material_name: NULL
      calculation_mode: internal
  target_safety_factor: 10.3919
          hoop_stress: 227778000
        radial_stress: -50000000
         axial_stress: 83687000
      von_mises_stress: 240619000
         safety_factor: 1.03899
       target_pressure: NULL
              delta_od: 0.000727008
              delta_id: 0.000829225
               delta_l: 0.000289083
          min_safe_od: NULL
          max_safe_id: NULL
   max_safe_axial_load: null
            created_at: 2025-12-15 13:44:20
```

9. Cloud Setup

This section describes the cloud infrastructure setup and deployment process for the system. The application adopts a cloud-based architecture in which the database, backend service, and frontend interface are deployed as independent but interconnected components on the Microsoft Azure platform.

This section covers:

Database: Azure Database for MySQL Flexible Server
Backend: Azure App Service
Frontend: Azure Static Web App

9.1 Creation and Deployment of Azure Database for MySQL Flexible Server

Azure Database for MySQL is a fully managed relational database service designed to run, manage, and scale high-performance MySQL databases in the cloud. The Flexible Server deployment model offers maximum control and customization and is optimized for mission-critical workloads.

9.1.1 Creating an Azure Database for MySQL Flexible Server Instance

The following are the key steps for creating a Flexible Server instance using the Azure portal:

Step 1: Search for the Service

1. Log in to the Azure Portal.
2. Search for and select "Azure Database for MySQL flexible servers" in the top search bar.
3. Select "Create".

Step 2: Configure Basic Information

In the "Basics" tab, you need to configure the following essential settings:

- Subscription and Resource group: Select or create a resource group to manage your server resources.
- Server details:
  - Server name: Enter a unique name (e.g., *capstonedb*).
  - Region: Select the geographical location for server deployment (e.g., *West US 2*).
  - MySQL version: Choose the desired MySQL version.
- Compute + storage:
  - Select the Workload type and Pricing tier. This determines the number of vCores, memory, IOPS, and storage size (e.g., Burstable or General Purpose).
- Administrator account:
  - Admin username: The default is typically *db_admin*.
  - Password: Set a strong administrator password.

Step 3: Configure Networking

Networking configuration is crucial for connection security:

- Connectivity method:
  - Public access (allowed IP addresses): The server is accessible via a public IP address. You must specify the allowed client IP address ranges in the Firewall rules.
  - Private access (VNet Integration): The server is only accessible via a Virtual Network (VNet), offering higher security.
- For Public Access, in the Firewall rules section:
  - Add current client IP address: This adds your current device's public IP address to the whitelist to allow local connection.

- (Optional) Allow public access from any Azure service to this server: Check this if your application is hosted on other Azure services (like Azure App Service or Azure Functions).

Step 4: Review and Create

1. Select "Review + create".
2. Review all configuration details, and if correct, select "Create".
3. Wait for the deployment to complete.

9.1.2 Post-Deployment: Viewing Server Overview and Connection Information

After deployment, navigate to your MySQL Flexible Server resource page:

- The Overview page provides key information:
  - Server name: E.g., *capstonedb.mysql.database.azure.com*. This is the Hostname/Endpoint required for connection.
  - Server admin login name: E.g., *db_admin*.
  - Configuration: Shows details like vCore, Memory, Storage, and IOPS.

9.1.3 Creating the Application Database

After the Azure Database for MySQL Flexible Server instance was successfully deployed, an application-specific database was created to store project data.

Using the MySQL command-line client, the database was created with the following command:

```
CREATE DATABASE <NAME>

CHARACTER SET utf8mb4

COLLATE utf8mb4_0900_ai_ci;
```

This database serves as the primary data store for the application. All tables and application data are created and managed within this database.

9.1.4 Configuring the Database Connection

After the application database was created, the database connection was configured using an environment variable in the **.env** file. The backend service connects to the Azure Database for MySQL Flexible Server through the **DATABASE_URL** variable, which specifies the server endpoint, credentials, and database name.

```
DATABASE_URL =

mysql+pymysql://<username>:<password>@<hostname>:<port>/<database_name>?charset=utf8mb4
```

The connection string follows this format:This configuration enables the backend application to establish a secure connection to the cloud-hosted MySQL database without hardcoding connection details.

9.2 Creation and Deployment of Azure App Service

The backend service of the system was deployed using **Azure App Service**, a fully managed platform for hosting web applications and APIs. Azure App Service provides built-in scalability, security, and integration with other Azure services, making it suitable for deploying the backend component of the application.

9.2.1 Creating an Azure App Service Instance

The following steps describe the creation of the backend App Service using the Azure Portal:

Step 1: Create a Web App

1. Log in to the Azure Portal.
2. Select Create a resource and search for Web App.
3. Click Create to begin the configuration.

Step 2: Configure Basic Settings

In the *Basics* tab, configure the following options:

- Subscription and Resource Group: Select the existing resource group used for the project (e.g., *SLB_project_mysql*).
- Instance Details:
  - Name: Enter a unique application name (e.g., *slb-capstone-api*).
  - Publish: Select Code.
  - Runtime stack: Select Python 3.11.
  - Operating System: Select Linux.
  - Region: Select the same region as the database server (e.g., *West US*).
- App Service Plan:
  - Select or create an App Service Plan (e.g., *ASP-SLBprojectmysql-b779*).

9.2.2 Endpoint Configuration

After the deployment was completed, two Azure resources became visible in the resource group: an Azure App Service (Web App) and an associated App Service Plan.

The App Service hosts the backend application code, while the App Service Plan defines the underlying compute environment used to run the application.

Within the Overview page of the App Service, a Default Domain is provided by Azure. This domain represents the public endpoint of the backend service and serves as the base URL for all exposed RESTful API endpoints.

Example:

```
slb-capstone-api-huatesgzcmbpa9ef.westus-01.azurewebsites.net
```

To enable communication between the frontend and backend components, the frontend configuration must be updated to reference this backend service address. Specifically, the constant variable **API_BASE** defined in the **scripts.js** should be set to the Default Domain of the Azure App Service. This ensures that the frontend application can correctly locate and interact with the backend API.

9.2.3 Environment Variables

Use Azure App Service environment variables to configure the backend application.

Step 1: Navigate to Environment Variables

- In the Azure Portal, open the backend App Service.
- From the left navigation menu, go to Settings → Environment variables.

Step 2: Add variables from the .env file

- Under App settings, create entries for all variables defined in the local .env file (e.g., *DATABASE_URL, SECRET_KEY, ALGORITHM, etc.*).

- Save the configuration changes. Azure will apply these settings to the running application process (a restart may be triggered depending on the changes).

Step 3: Configure deployment build behavior

In addition to the **.env** variables, the following setting was added:

```
SCM_DO_BUILD_DURING_DEPLOYMENT = 1
```

This variable instructs Azure App Service (Kudu/Oryx build system) to perform the build steps during deployment, such as installing dependencies (e.g., from **requirements.txt**) and preparing the runtime environment. Without enabling build during deployment, the application may be deployed as raw source files without dependency installation, which can cause startup failures or missing-module errors when the backend service launches.

9.2.4 Configuration

In Azure App Service, the backend service is configured by navigating to **Settings** → **Configuration**. Under **General settings**, the runtime stack is set to **Python 3.11**. The application is started using the following startup command:

```
gunicorn -w 4 -k uvicorn.workers.UvicornWorker server:app
```

This command launches the FastAPI application using **Gunicorn** as the process manager with four **Uvicorn** worker processes for handling concurrent requests. The **server:app** argument specifies the Python module and FastAPI application instance to be executed. This configuration enables reliable and scalable production deployment.

9.2.5 Backend Source Code Deployment

Deploy the backend source code using **Deployment Center** in Azure App Service.
You may choose any supported deployment method. In this project, a ZIP file upload was used as an example.

```
app.zip
├── calculator_vessel.py
├── data_shapes.py
├── db_init.py
├── security.py
├── server.py
├── requirements.txt
└── README.md
```

- Open the backend App Service in the Azure Portal.
- Navigate to Deployment → Deployment Center.
- Select Upload ZIP file as the deployment source.
- Upload the backend source code package and complete the deployment.

After deployment, the backend service becomes accessible through the App Service Default Domain.

9.2.6 Backend Service Verification

Once deployment was completed, the backend service was verified by accessing the default application URL provided by Azure App Service.

You can use the following command for activation test:

```
https://slb-capstone-api-huatesgzcmbpa9ef.westus-01.azurewebsites.net/healthz
```

Correct Response:

{"status":"ok"}

Successful access to the service confirmed that the backend application was running correctly and ready to handle requests from the frontend and database layers.

9.3 Creation and Deployment of Azure Static Web App

The frontend of the system was deployed using **Azure Static Web Apps**, which provides a managed hosting service optimized for static frontend applications with built-in support for continuous deployment.

9.3.1 Creating an Azure Static Web App

- Log in to the Azure Portal.
- Select **Create a resource** and search for **Static Web App**.
- Click **Create** to start the configuration process.

In the *Basics* configuration tab, set the following options:

- **Subscription and Resource Group**:
  Select the existing project resource group.
- **Static Web App details**:
  - Name: Enter a unique name (e.g., *slb-proxy-frontend*).
  - Hosting plan: Choose from demand
  - Region: Select an appropriate region.
- **Deployment details**:
  - Source: **GitHub**.
  - Repository and branch: Select the frontend repository and deployment branch.

```
frontend
├── index.html
├── scripts.js
├── style.css
```

- Build presets: Select the appropriate framework preset based on the frontend project.

9.3.2 Frontend Deployment and Access

After the Static Web App was created, Azure automatically configured a **GitHub Actions workflow** to build and deploy the frontend application. Each update pushed to the selected branch triggers an automatic redeployment.

Once deployment is completed, the frontend application becomes accessible through the default URL provided by Azure Static Web Apps:

```
https://<static-app-name>.azurestaticapps.net
```

This URL serves as the public entry point of the frontend application and communicates with the backend service through configured API endpoints.