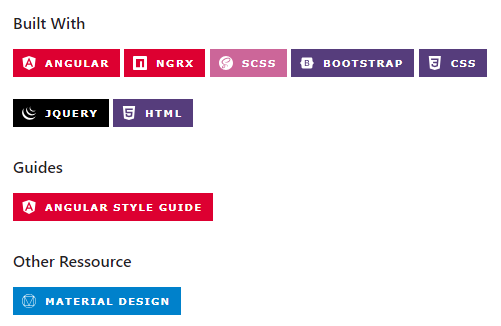
**Target Room Frontend Project**

#### 💻 About The Project

In this project ,i will show you how to build a Full Stack Application. the frontend is build using Angular V16 which represent the client side and th backend use Springboot which is server side.



## 🔐 Instalation,Configurationand Usage

#### **Instal Prerequisites:**

* Install Angular CLI : npm install -g @angular/cli
* Install Node.js and npm from official website : <https://nodejs.org/>

#### **Add Material Design:**

ng add @angular/material

#### **Add jQuery and Boostrap:**

npm install bootstrap npm install jquery

## ⚙️ Getting Started

1. **Clone repo**

https://github.com/EmnaSallemi/TARGET\_ROOM\_Stage.git

1. **Installing Dependencies**

npm install

1. **Starts the Development Server**

o For Windows: ng serve

o For Linux : npx ng serve -o

## 🚀 Usage

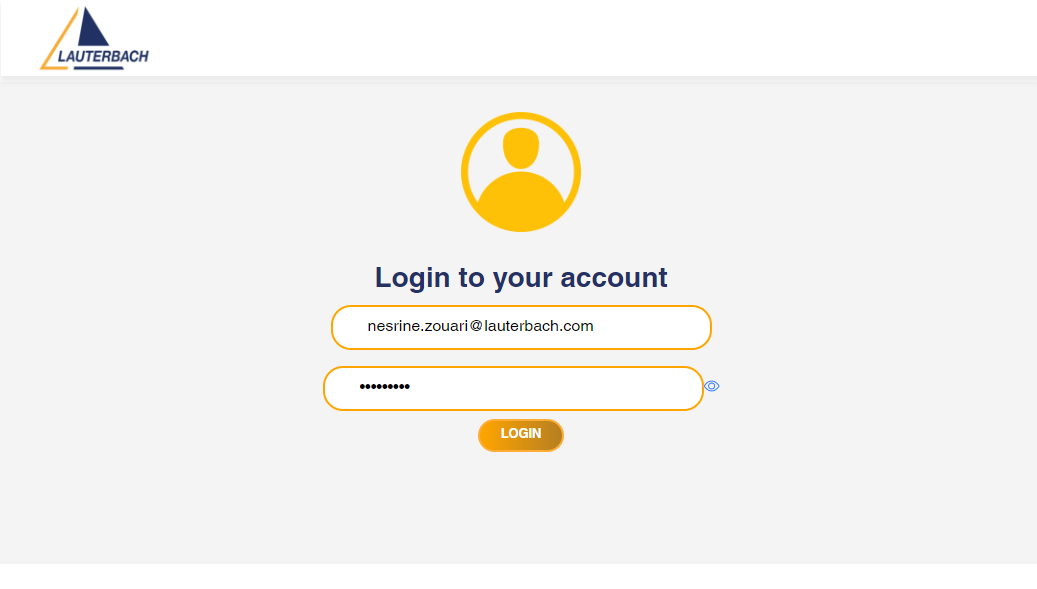
#### **Development server**

Run ng serve for a dev server. Navigate to <http://localhost:4200/>. The application will automatically reload if you change any of the source files.

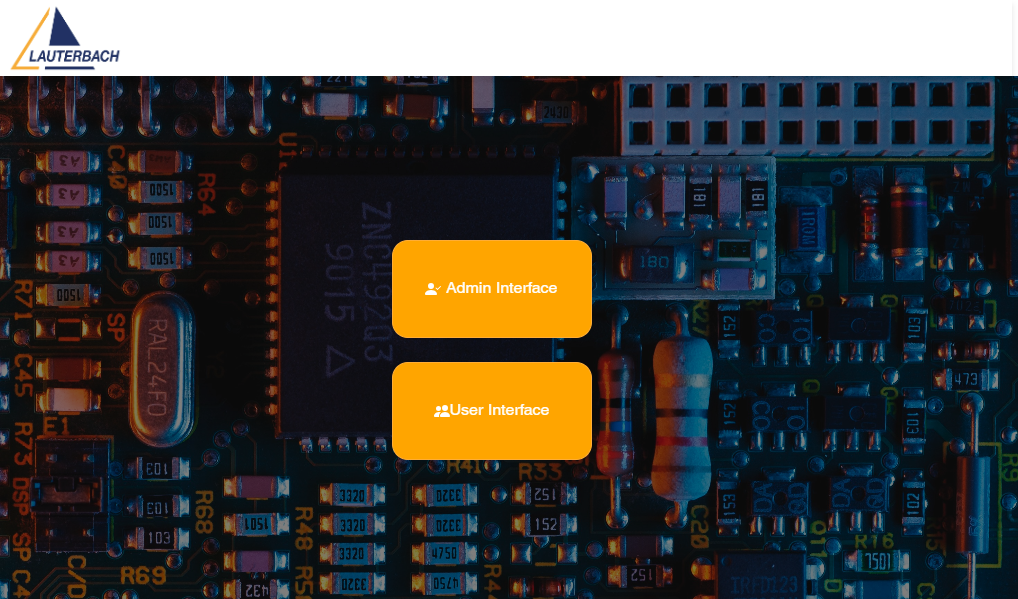
### 💪Demo

#### **Admin Interface**

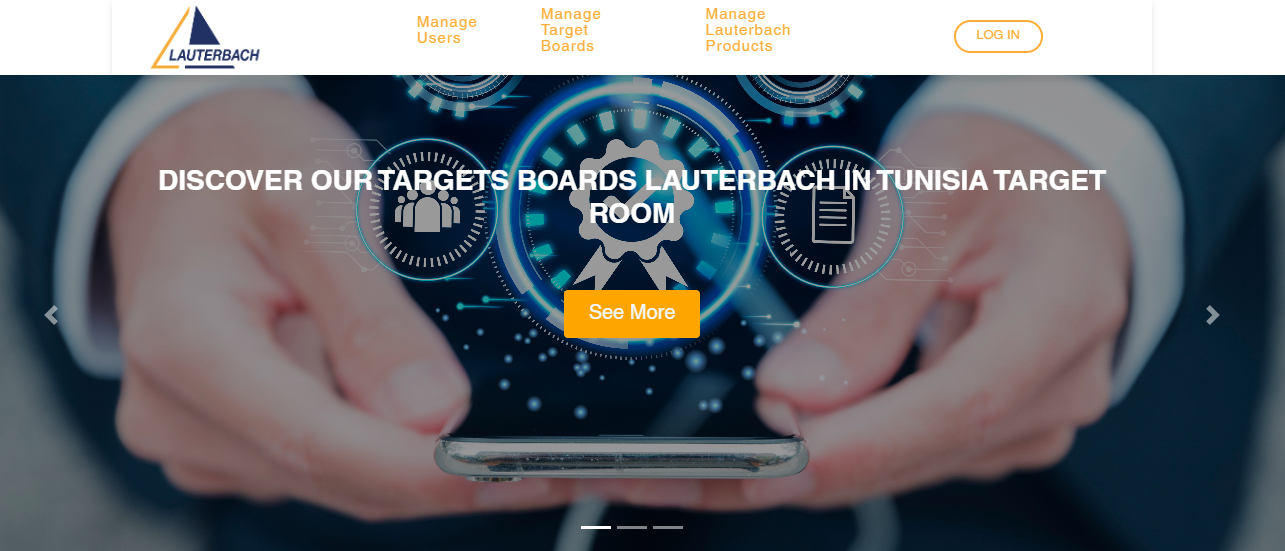
1. login page



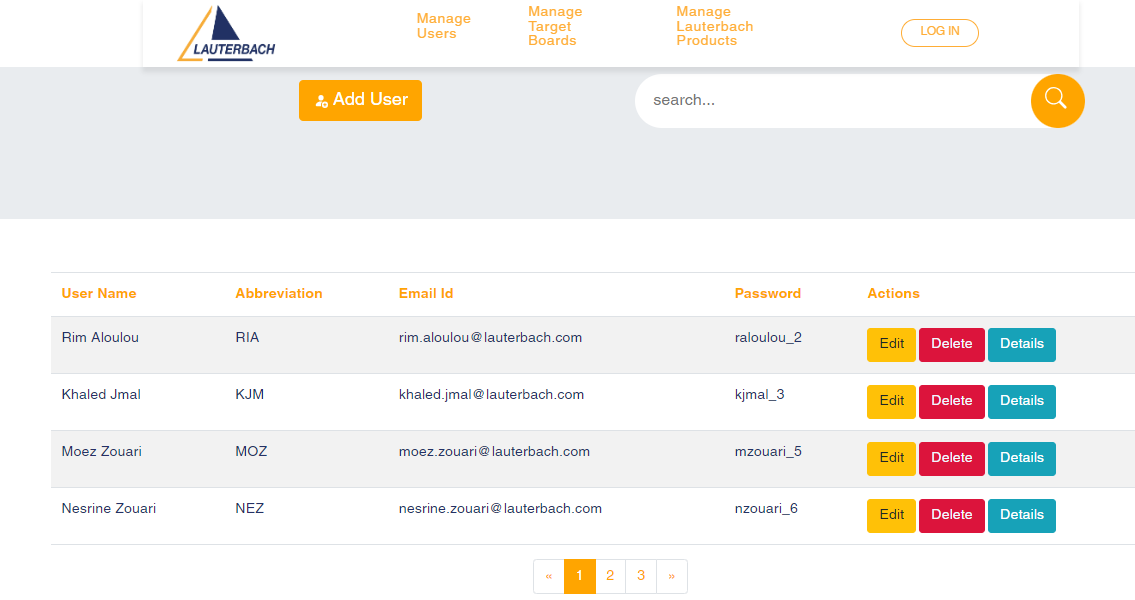
1. Choose interface **(Sauf for coder)**

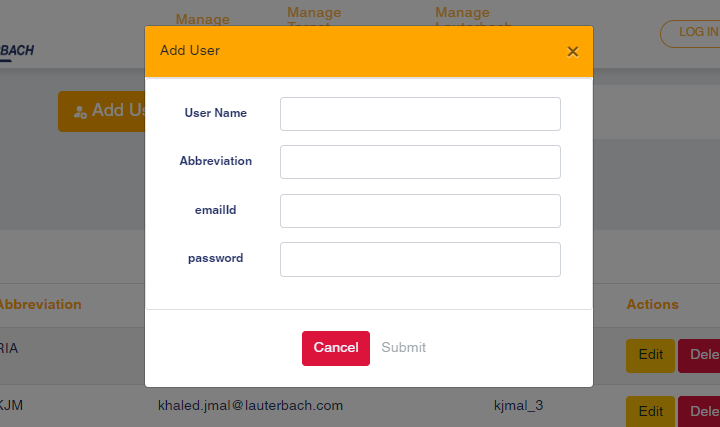
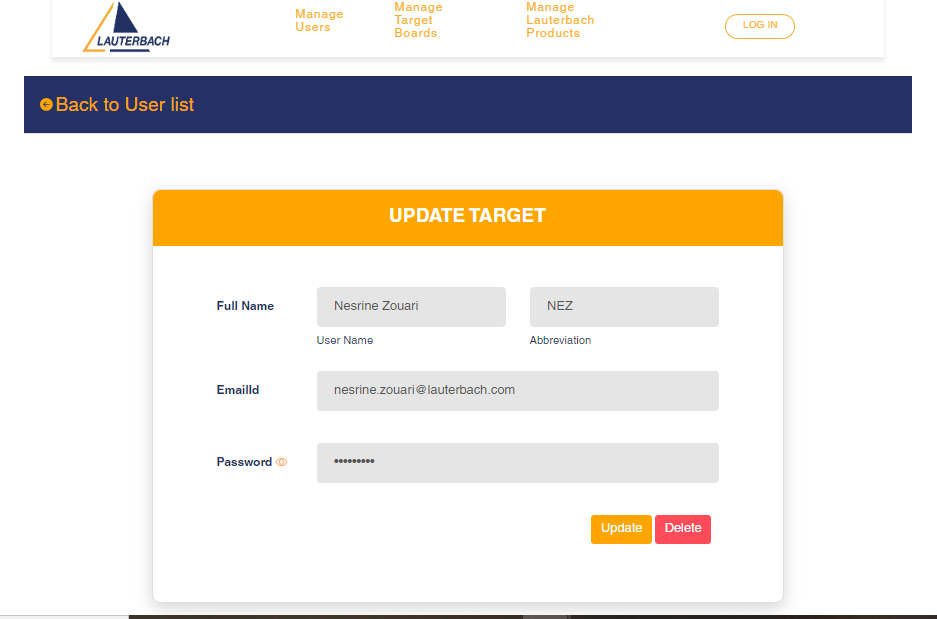
****

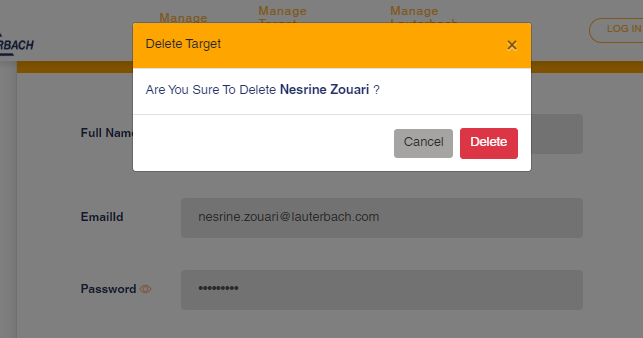
1. Admin Interface

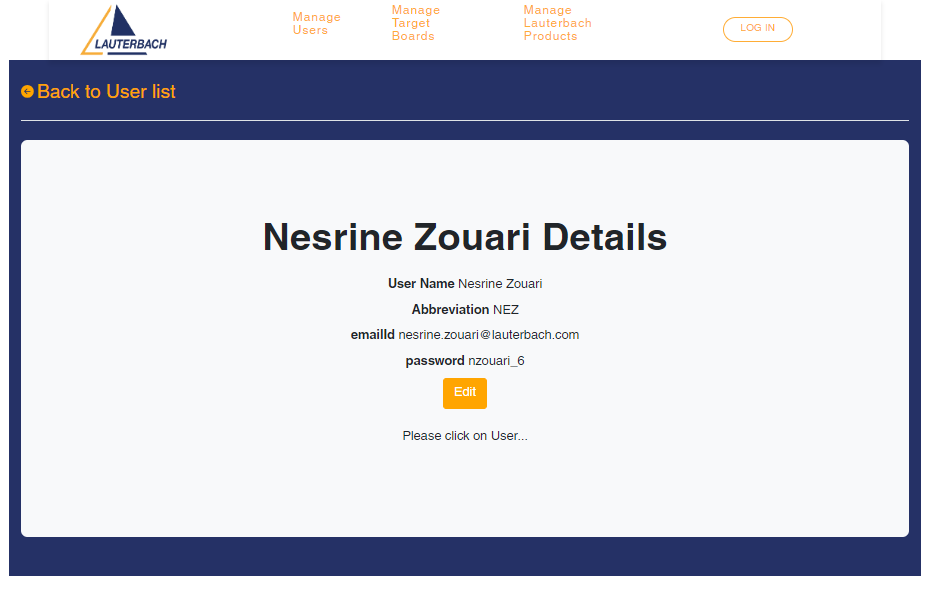


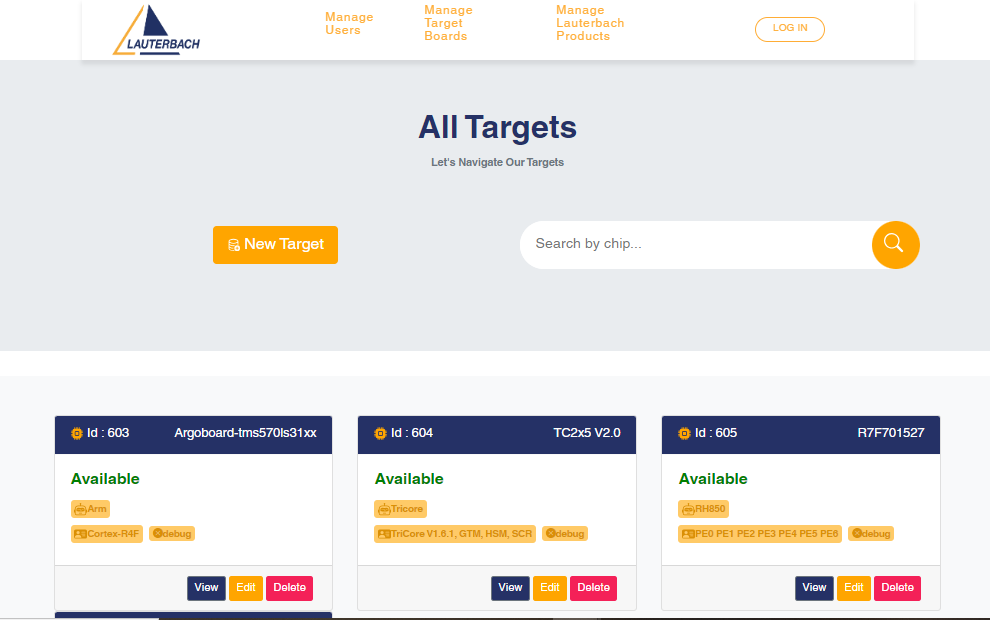
* Manage User

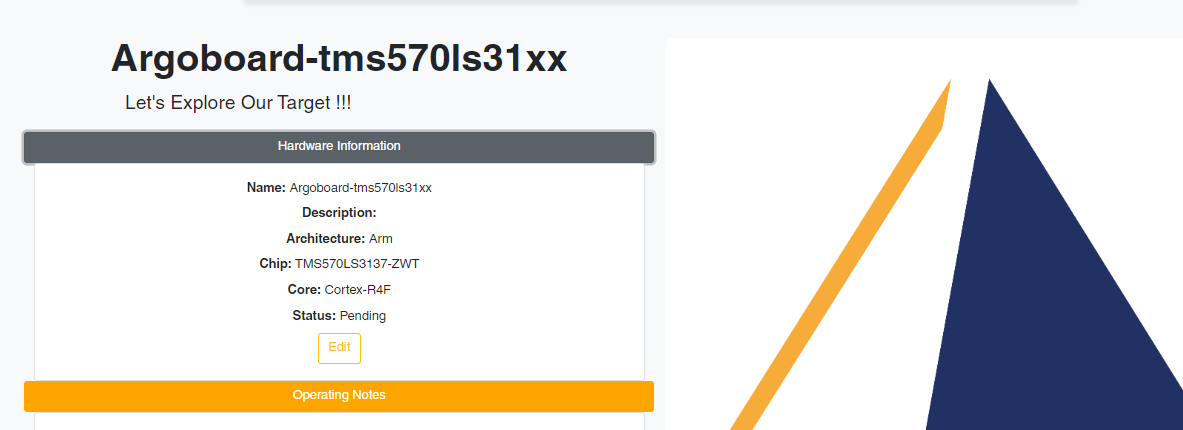
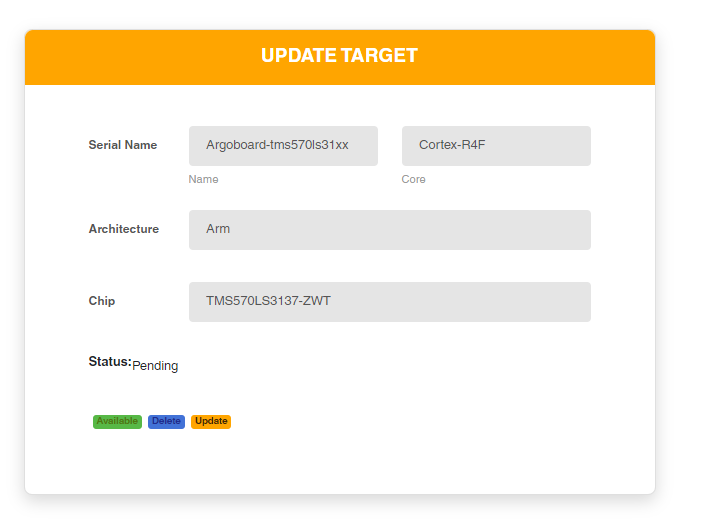


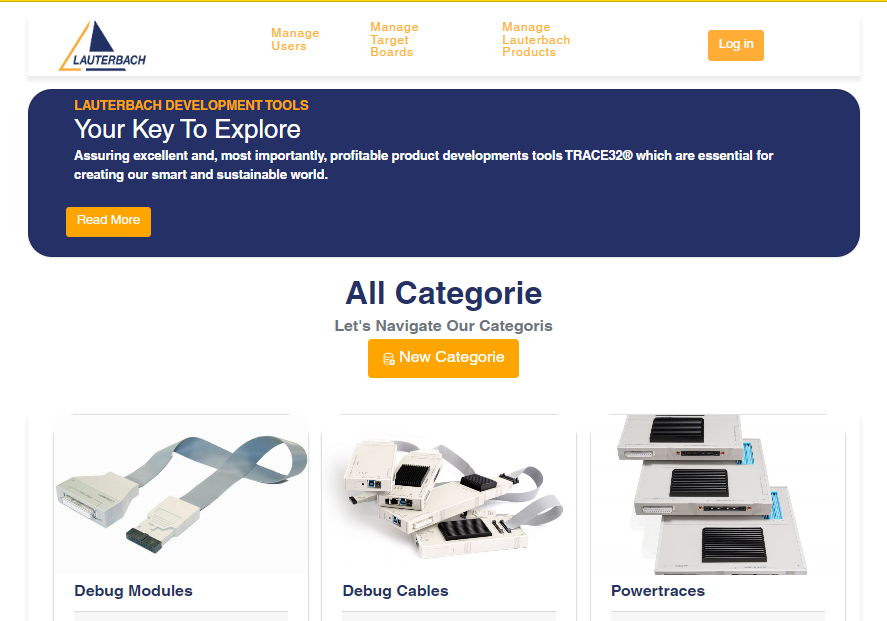
*  Add user
* Edit User
* Delete User



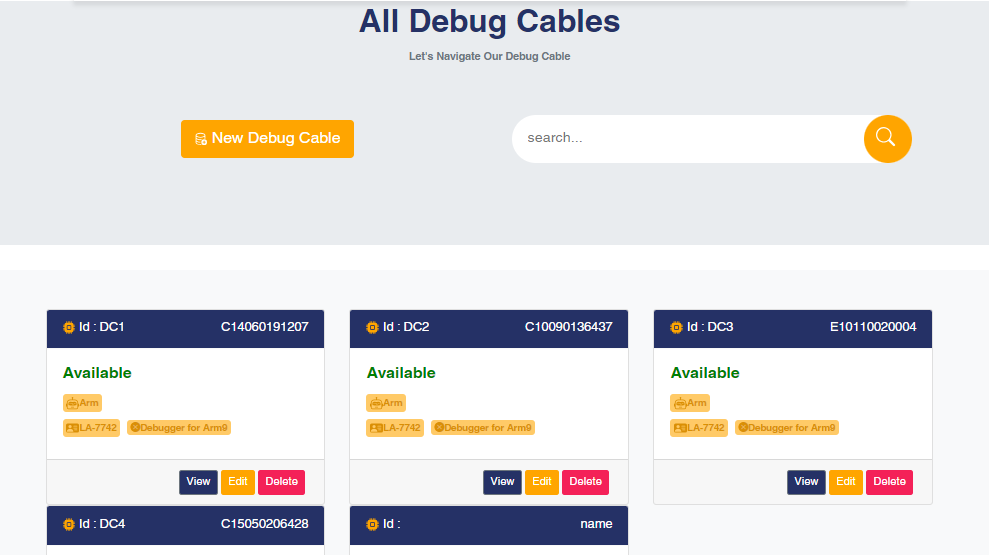
* View Details

4. Manage Target Boards

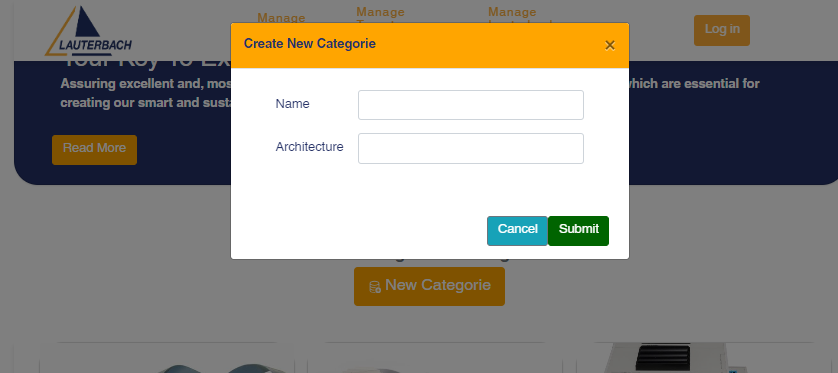
* Target Details
* Target Edit

1. Lauterbach Products

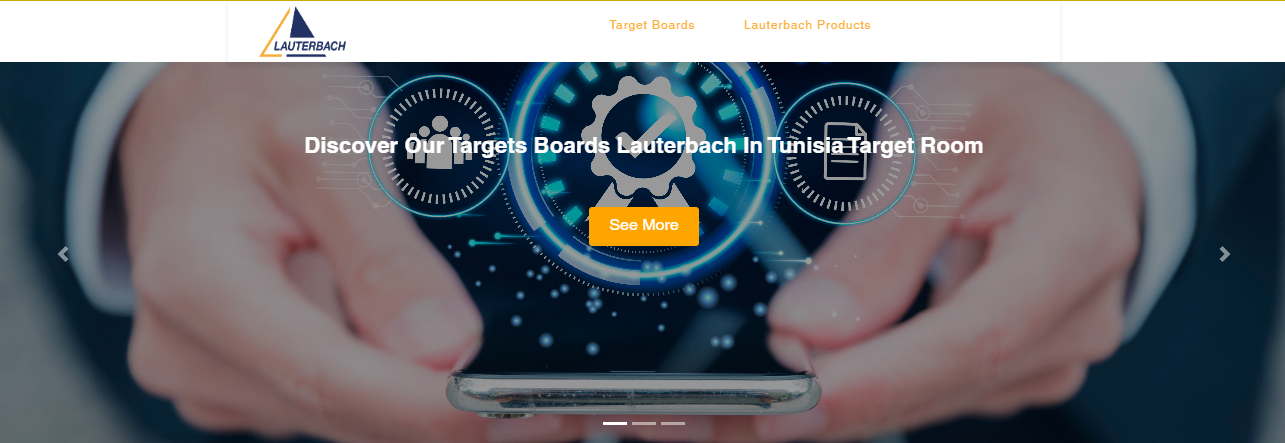
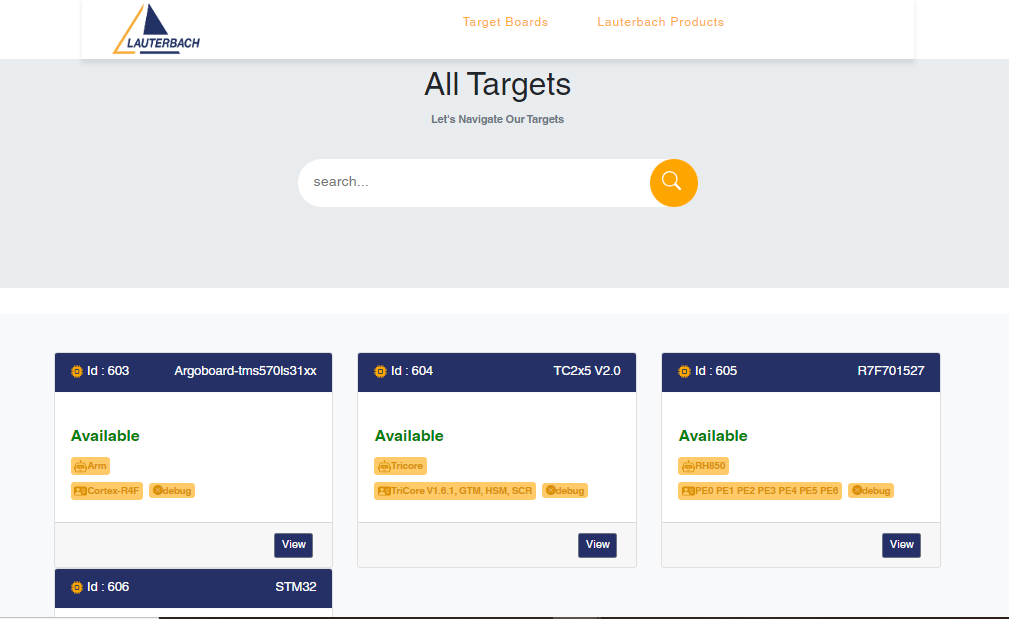
* Exemple of Debug Cable of Lauterbach Products

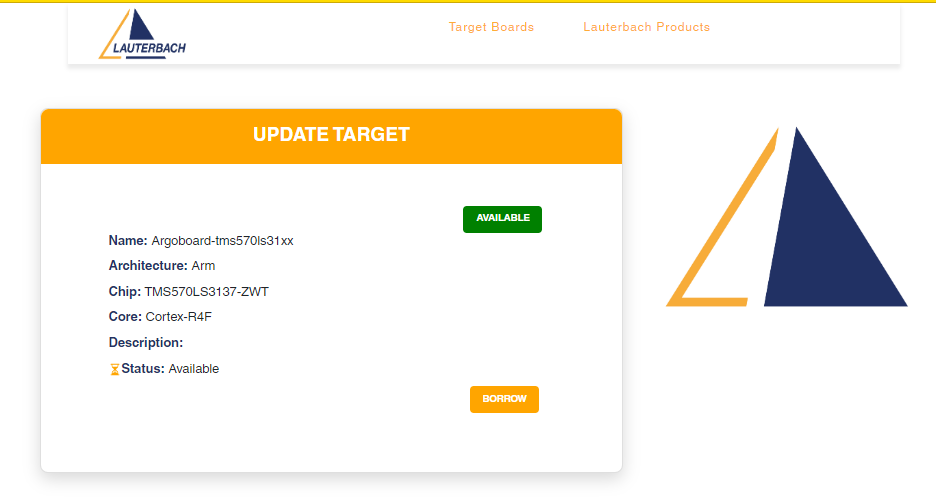


* Manage categories

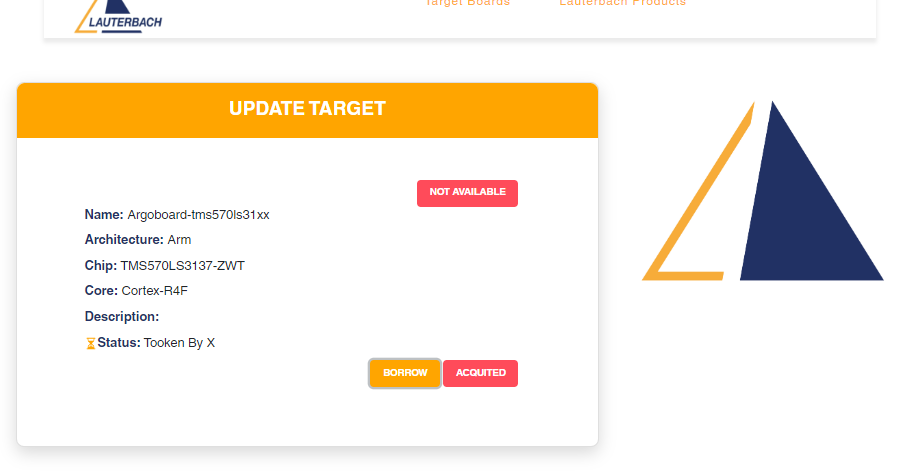


**User Interface**

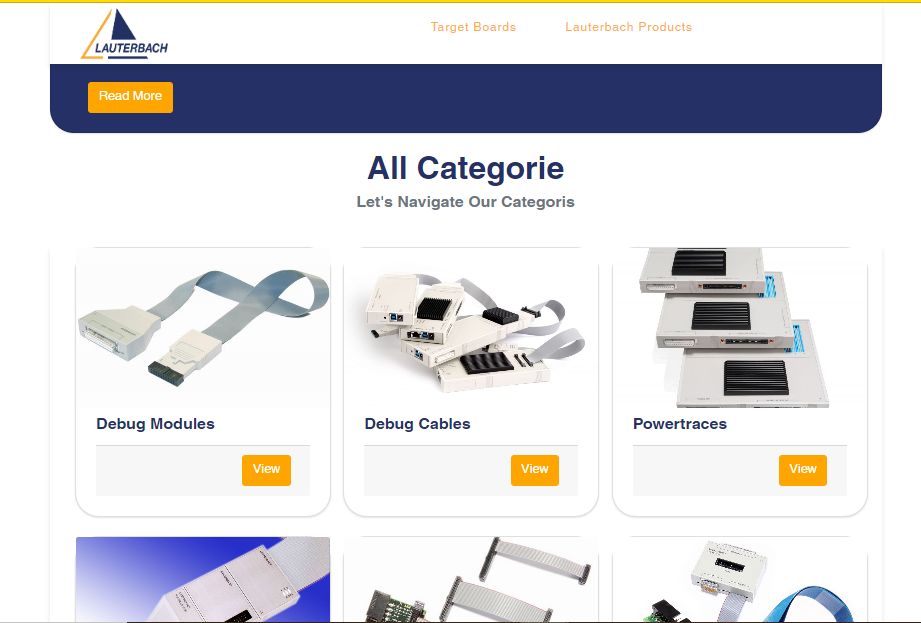
* 1. Home
  2.  Target Boards
* Target Board Details

-In case target not available :

* + In case not available :



* 1. Lauterbach Products



#### **Code scaffolding**

Run ng generate component component-name to generate a new component. You can also use ng generate directive|pipe|service|class|guard|interface|enum|module.

#### **Build**

Run ng build to build the project. The build artifacts will be stored in the dist/ directory.Use the--configuration productrion flag for a production build.

#### **Running unit tests**

Run ng e2e to execute the end-to-end tests via a platform of your choice. To use this command, you need to first add a package that implements end-to-end testing capabilities.

#### **Further help**

To get more help on the Angular CLI use ng help or go check out the [Angular CLI Overview and Command Reference](https://angular.io/cli) page.

## 🔍 Contributing

Contributions are what make the open source community such an amazing place to be learn, inspire, and create. Any contributions you make are **greatly appreciated**.

1. Fork the Project.
2. Create your Feature Branch git checkout -b feature/AmazingFea
3. Commit your Changes git commit -m 'Add some AmazingFeature)
4. Push to the Branch git push origin feature/AmazingFeature
5. Open a Pull Request.

## Hosting

1. Install Required Software:

Make sure you have the following software installed on your Linux server:

* + Node.js and npm (for Angular)
  + Java Runtime Environment (JRE) or Java Development Kit (JDK) (for Spring Boot)
  + MySQL

1. Set Up MySQL Database:

Install MySQL on your server and create a database for your application. You'll need to configure a database user and grant necessary permissions.

1. Build Angular Application:

Navigate to your Angular project directory and build the application using the ng build command. This will generate static HTML, CSS, and JavaScript files in the dist directory.

1. Build Spring Boot Application:

Build your Spring Boot application using your preferred build tool (e.g., Maven or Gradle). This will generate a JAR or WAR file containing your backend code.

1. Deploy Spring Boot Application:

Deploy your Spring Boot application by running the generated JAR or WAR file. You can use the java -jar command to start the application.

1. Configure Web Server:

Set up a web server (e.g., Apache or Nginx) to serve the Angular frontend files. Configure the server to point to the dist directory generated by the Angular build.

1. Configure Domain and Hosts:

Edit your server's hosts file to associate a domain name (e.g., myapp.local) with your server's IP address. This allows you to access the application using the domain name.

1. Test Your Application:

Open a web browser and navigate to the configured domain name. You should see your Angular frontend being served by the web server and communicating with the Spring Boot backend.

## 📧 Contact

Emna SALLEMI - [linkedin.com/in/emna-sallemi-8709a5223](https://github.com/EmnaSallemi/TARGET_ROOM_Internship/blob/main/linkedin.com/in/emna-sallemi-8709a5223)

Project Link: <https://github.com/EmnaSallemi/frontend_targetroom.git>

* Exemple of Debug Cable of Lauterbach Products

