https://www.halvorsen.blog

ASP.NET Core RFID Application

Hans-Petter Halvorsen

Contents

- In this Tutorial we will create an RFID Web Application in ASP.NET Core that can scan RFID Tags
- The Application will also Add, Edit and Delete Tag Information stored in a SQL Server Database

RFID System

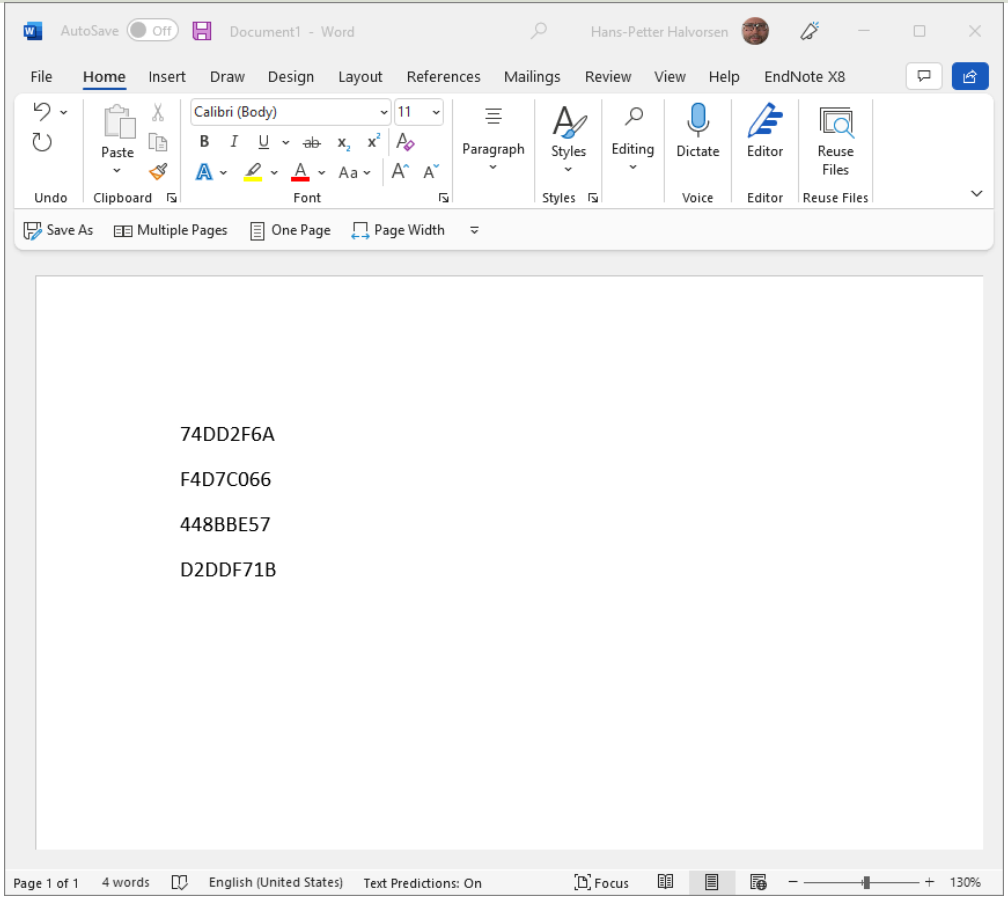
Desktop Reader NEO 2

MIFARE Classic 1K (ISO 14443A) Tags

https://en.idtronic-rfid.com/rfid-readers/rfid-hf-readers/desktop-reader-neo-2/

https://www.elfadistrelec.no/en/rfid-tag-hf-red-13-56mhz-nxp-mifare-idtronic-kf-mfs50-rd/p/30182163?trackQuery=RFID&pos=30&origPos=30&origPageSize=50&track=true

Testing



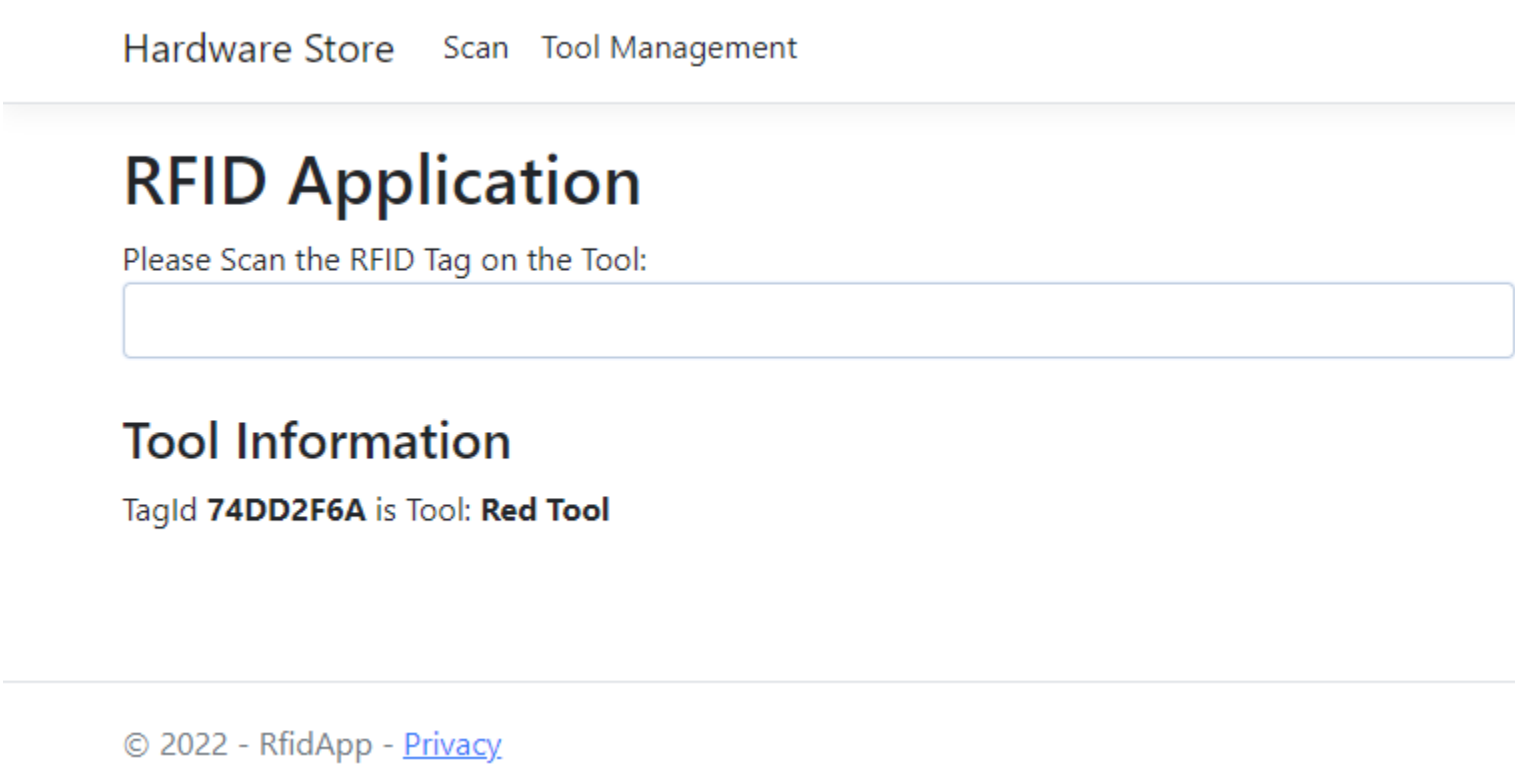
- Plug in the RFID Reader into your PC
- Open MS Word, Notepad, etc.
- Put a RFID Tag on top of the Reader
- Observe that the unique Tag UID is written into MS Word

ASP.NET Application

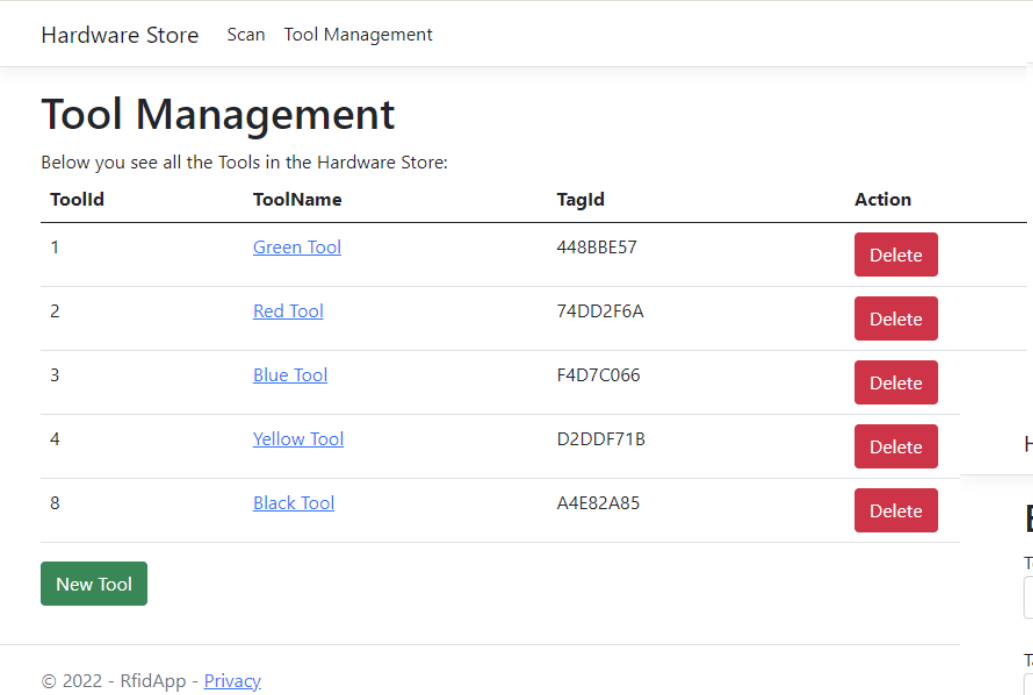
Hans-Petter Halvorsen

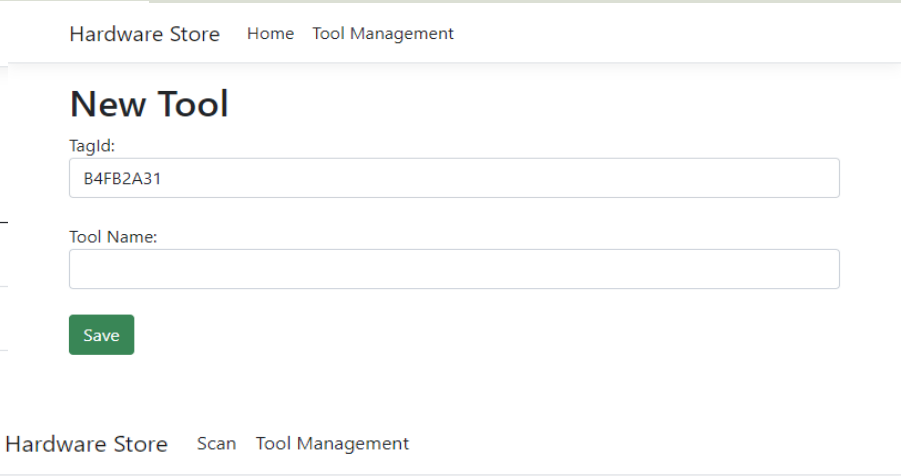
Table of Contents

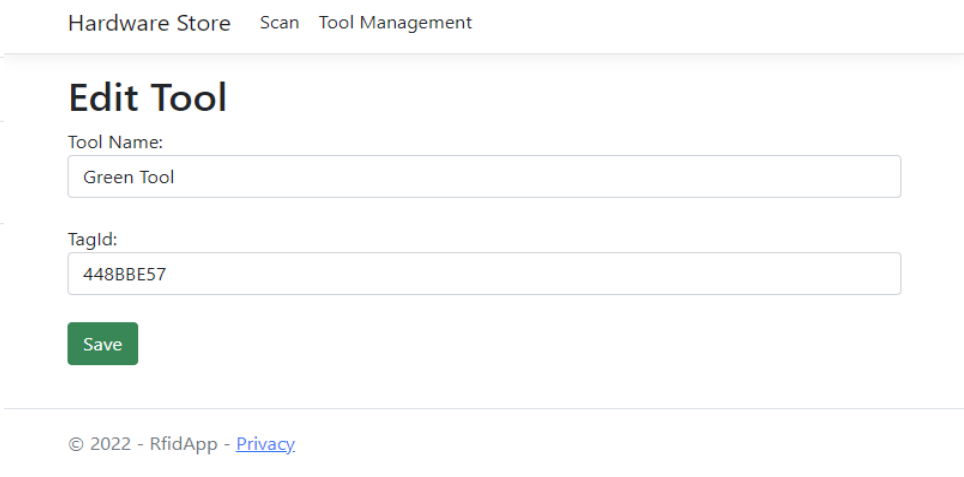
ASP.NET Application



ASP.NET Application







Step by Step

1. Scanning RFID Tags into a Textbox
2. Retrieve Information based on the scanned Tag
3. Retrieve Information from Database based on the scanned Tag
4. Get List of available Tools in the Database
5. Add New Tools into the Database
6. Edit Existing Tool in the Database
7. Delete Existing Tool in the Database

Step 1

Scanning RFID Tags into a Textbox

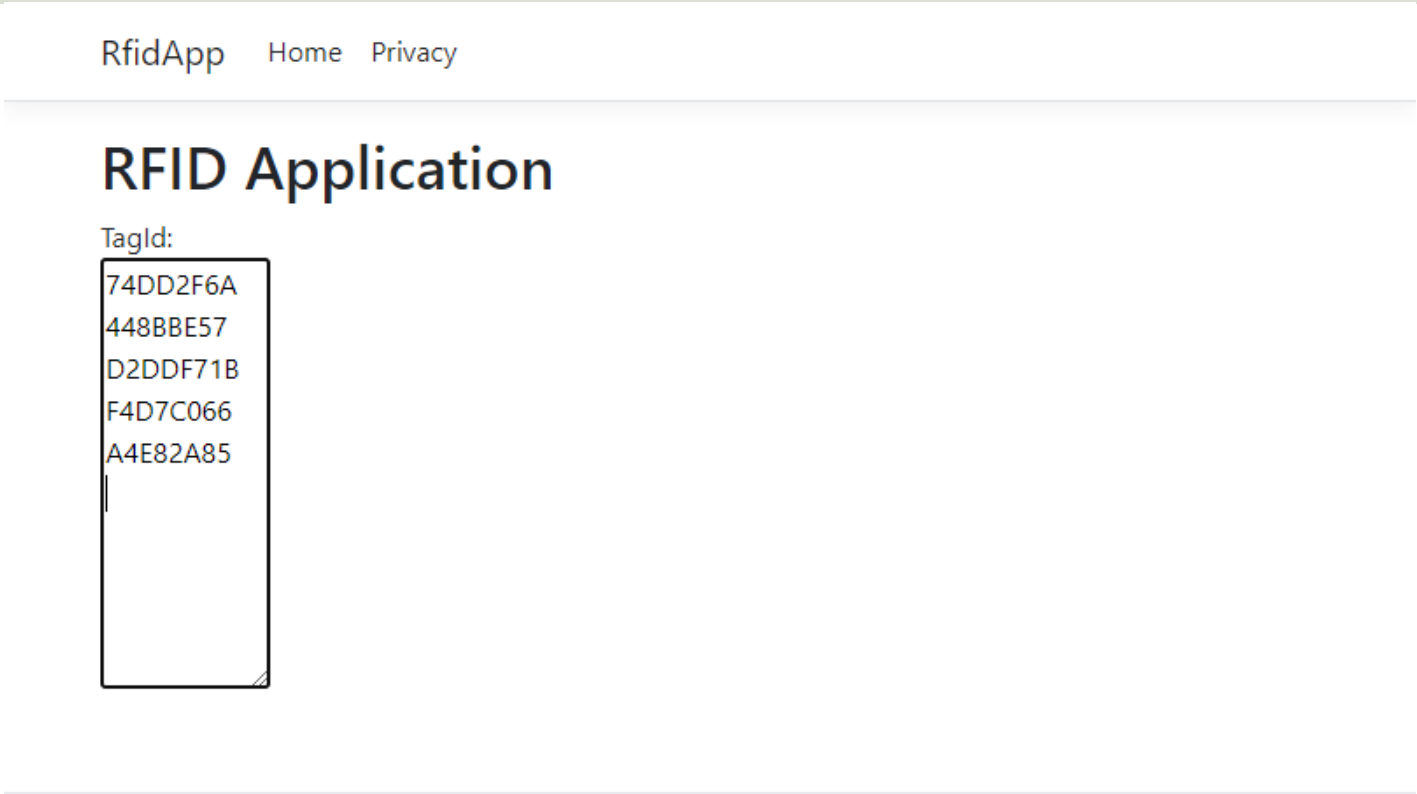
Hans-Petter Halvorsen

Table of Contents

Step by Step

1. **Scanning RFID Tags into a Textbox**
2. Retrieve Information based on the scanned Tag
3. Retrieve Information from Database based on the scanned Tag
4. Get List of available Tools in the Database
5. Add New Tools into the Database
6. Edit Existing Tool in the Database
7. Delete Existing Tool in the Database

ASP.NET Application



Step 2

Retrieve Information based on the scanned Tag

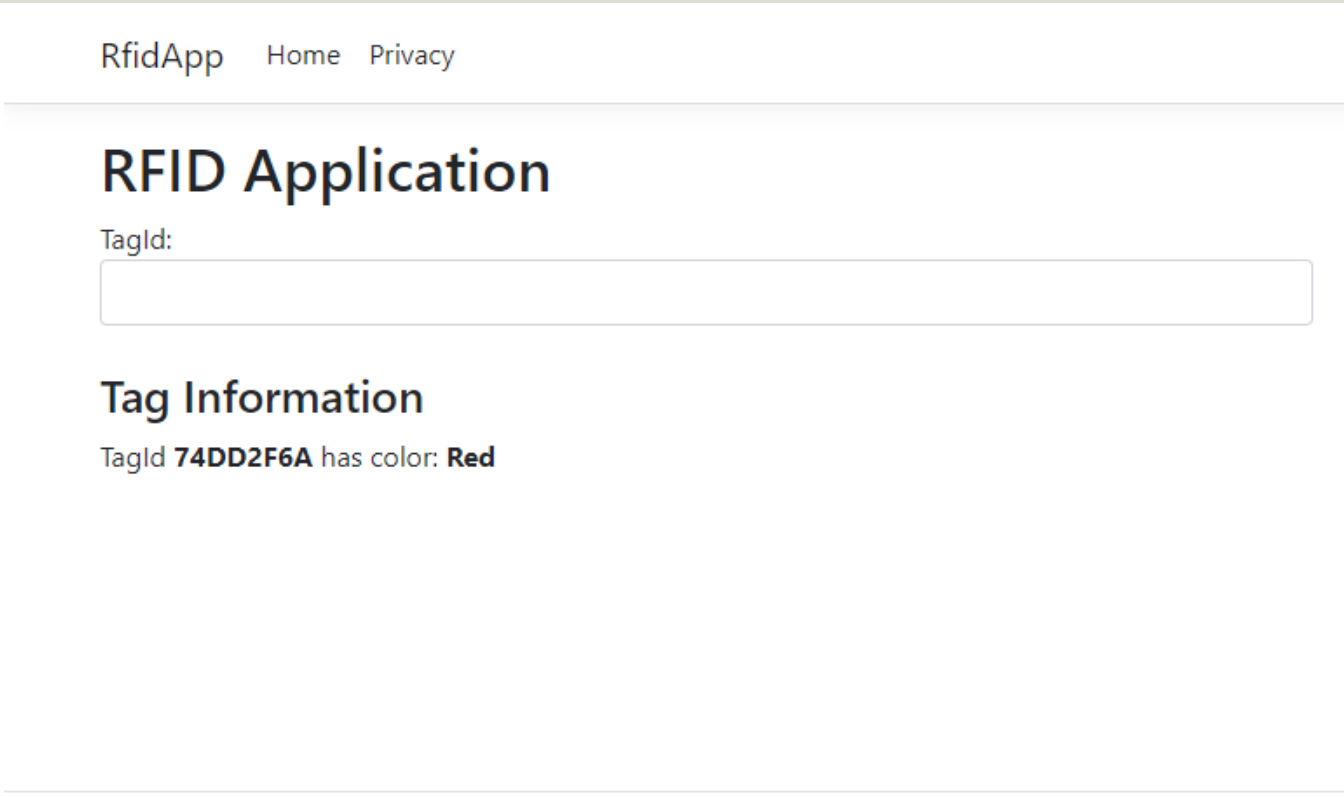
Hans-Petter Halvorsen

Table of Contents

Step by Step

1. Scanning RFID Tags into a Textbox
2. **Retrieve Information based on the scanned Tag**
3. Retrieve Information from Database based on the scanned Tag
4. Get List of available Tools in the Database
5. Add New Tools into the Database
6. Edit Existing Tool in the Database
7. Delete Existing Tool in the Database

ASP.NET Application



Step 3

Retrieve Information from **Database** based on the scanned Tag

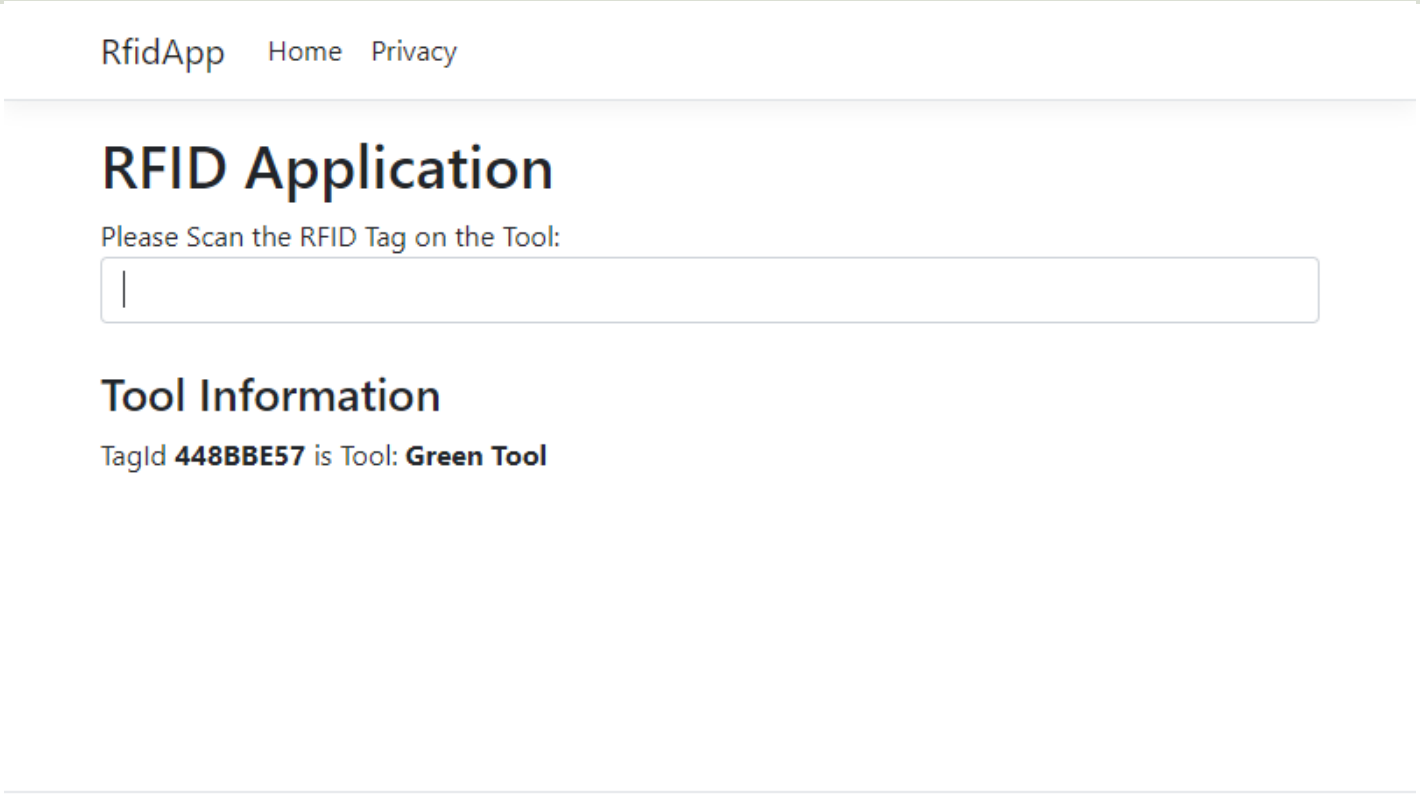
Hans-Petter Halvorsen

Table of Contents

Step by Step

1. Scanning RFID Tags into a Textbox
2. Retrieve Information based on the scanned Tag
3. **Retrieve Information from Database based on the scanned Tag**
4. Get List of available Tools in the Database
5. Add New Tools into the Database
6. Edit Existing Tool in the Database
7. Delete Existing Tool in the Database

ASP.NET Application



Database

```
CREATE TABLE [TOOL]
(
[ToolId]      int  NOT NULL  IDENTITY (1,1) Primary Key,
[ToolName]    varchar(100)  NOT NULL UNIQUE,
[TagId]      varchar(10)  NULL UNIQUE
)
go
```

Step 4

Get List of available Tools in the Database

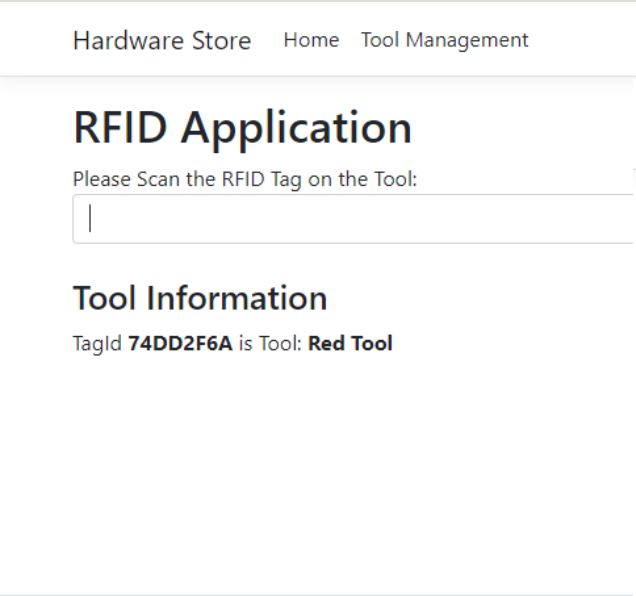
Hans-Petter Halvorsen

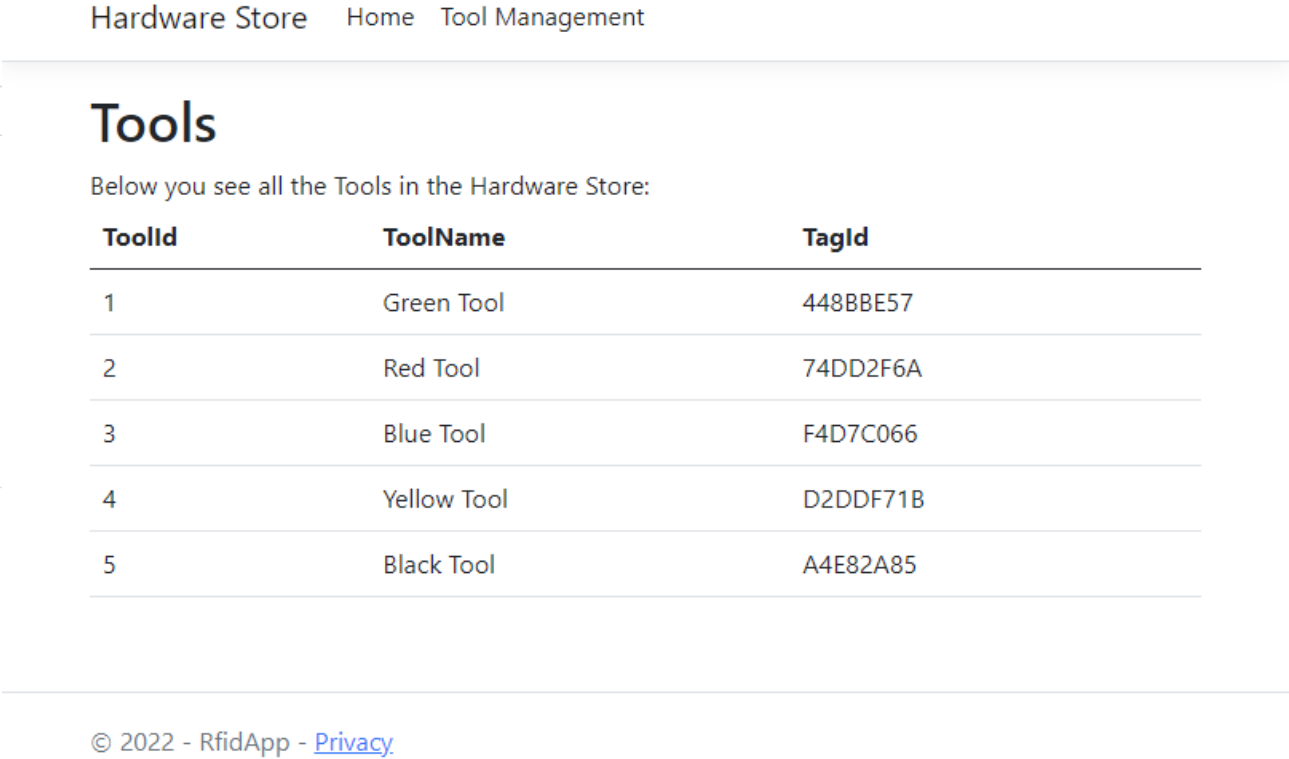
Table of Contents

Step by Step

1. Scanning RFID Tags into a Textbox
2. Retrieve Information based on the scanned Tag
3. Retrieve Information from Database based on the scanned Tag
4. **Get List of available Tools in the Database**
5. Add New Tools into the Database
6. Edit Existing Tool in the Database
7. Delete Existing Tool in the Database

ASP.NET Application





Step 5

Add New Tools into the Database

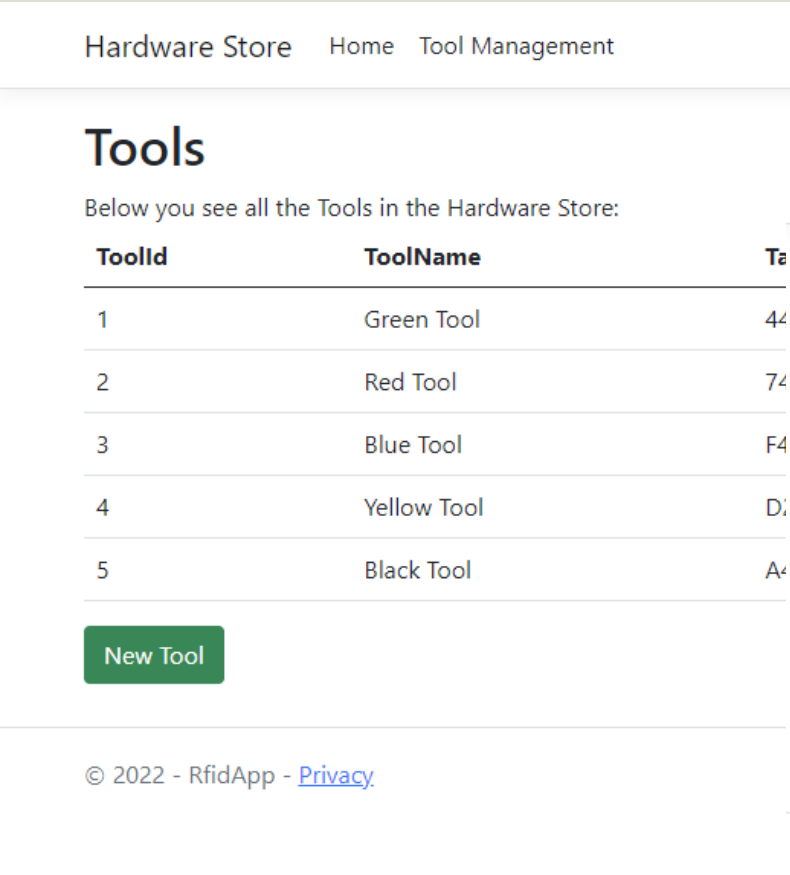
Hans-Petter Halvorsen

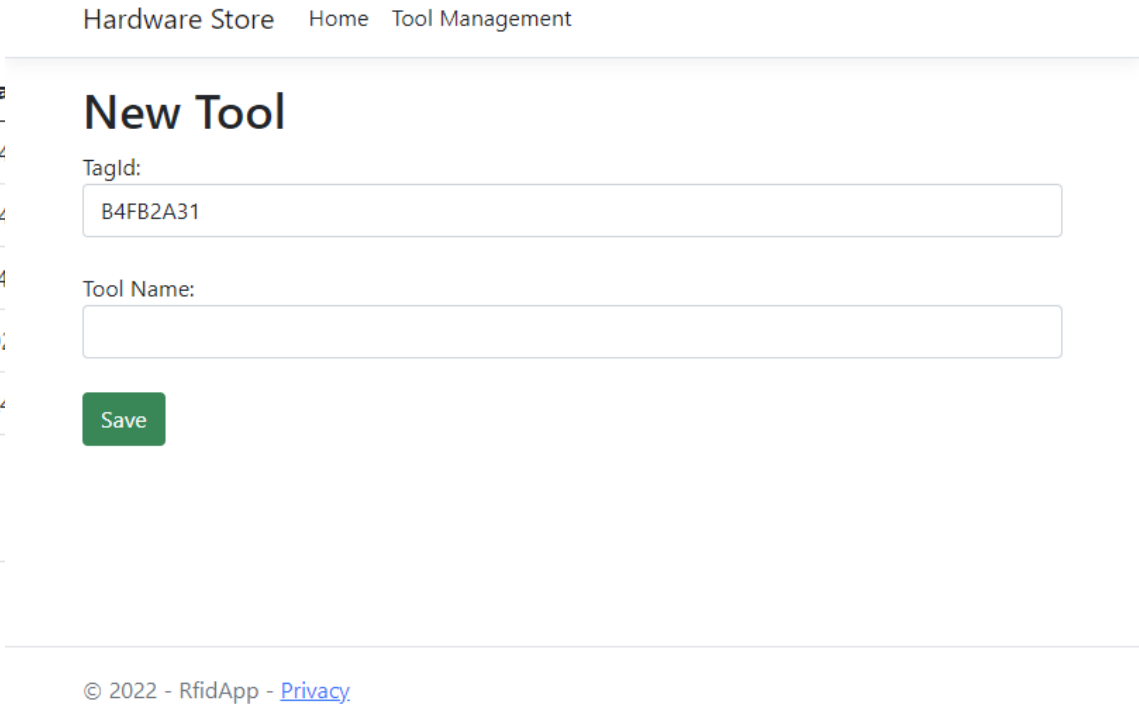
Table of Contents

Step by Step

1. Scanning RFID Tags into a Textbox
2. Retrieve Information based on the scanned Tag
3. Retrieve Information from Database based on the scanned Tag
4. Get List of available Tools in the Database
5. **Add New Tools into the Database**
6. Edit Existing Tool in the Database
7. Delete Existing Tool in the Database

ASP.NET Application





Database

```
CREATE PROCEDURE CreateTool
@ToolName varchar(100),
@TagId varchar(10)
AS

if not exists (select * from TOOL where TagId = @TagId)
   INSERT INTO TOOL (ToolName, TagId) VALUES (@ToolName,@TagId)

GO
```

Step 6

Edit Existing Tool in the Database

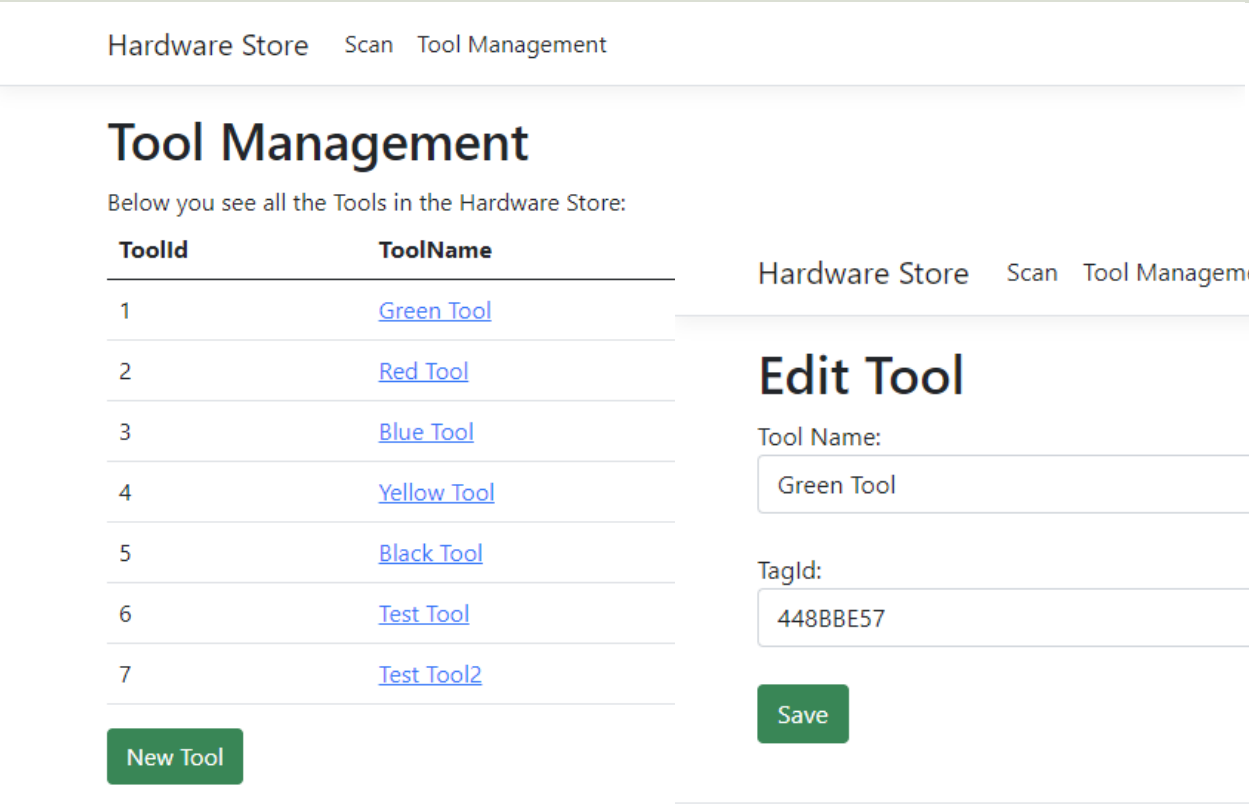
Hans-Petter Halvorsen

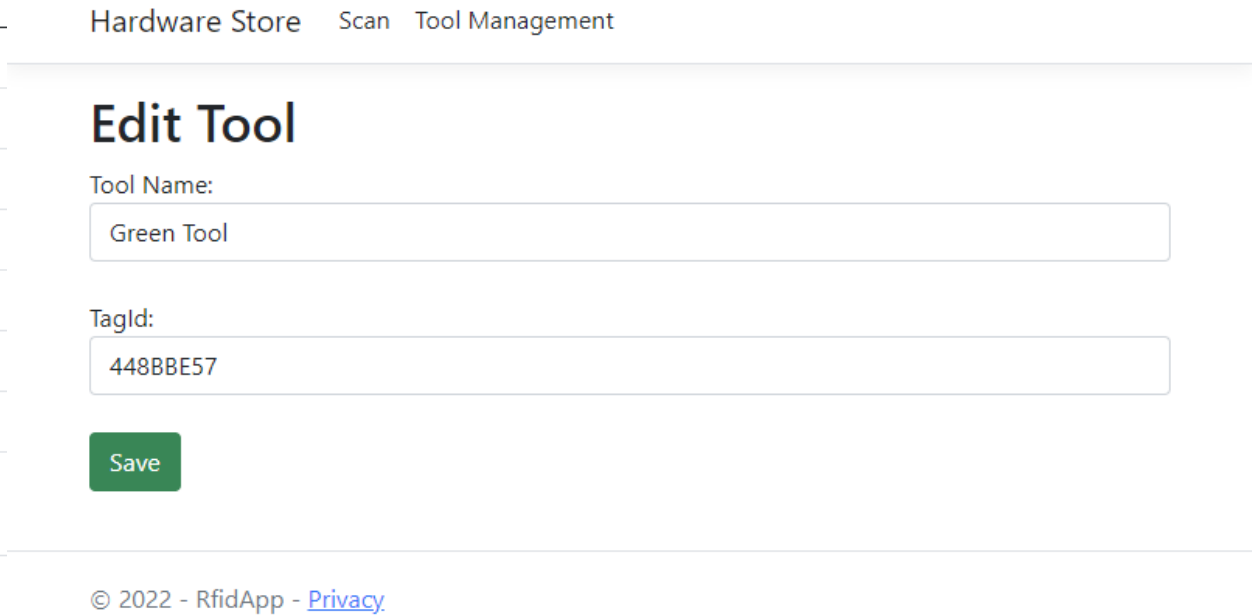
Table of Contents

Step by Step

1. Scanning RFID Tags into a Textbox
2. Retrieve Information based on the scanned Tag
3. Retrieve Information from Database based on the scanned Tag
4. Get List of available Tools in the Database
5. Add New Tools into the Database
6. **Edit Existing Tool in the Database**
7. Delete Existing Tool in the Database

ASP.NET Application





Database

```
CREATE PROCEDURE EditTool
@ToolId int,
@ToolName varchar(100),
@TagId varchar(10)
AS

UPDATE TOOL SET ToolName=@ToolName, TagId=@TagId WHERE ToolId=@ToolId

GO
```

Step 7

Delete Existing Tool in the Database

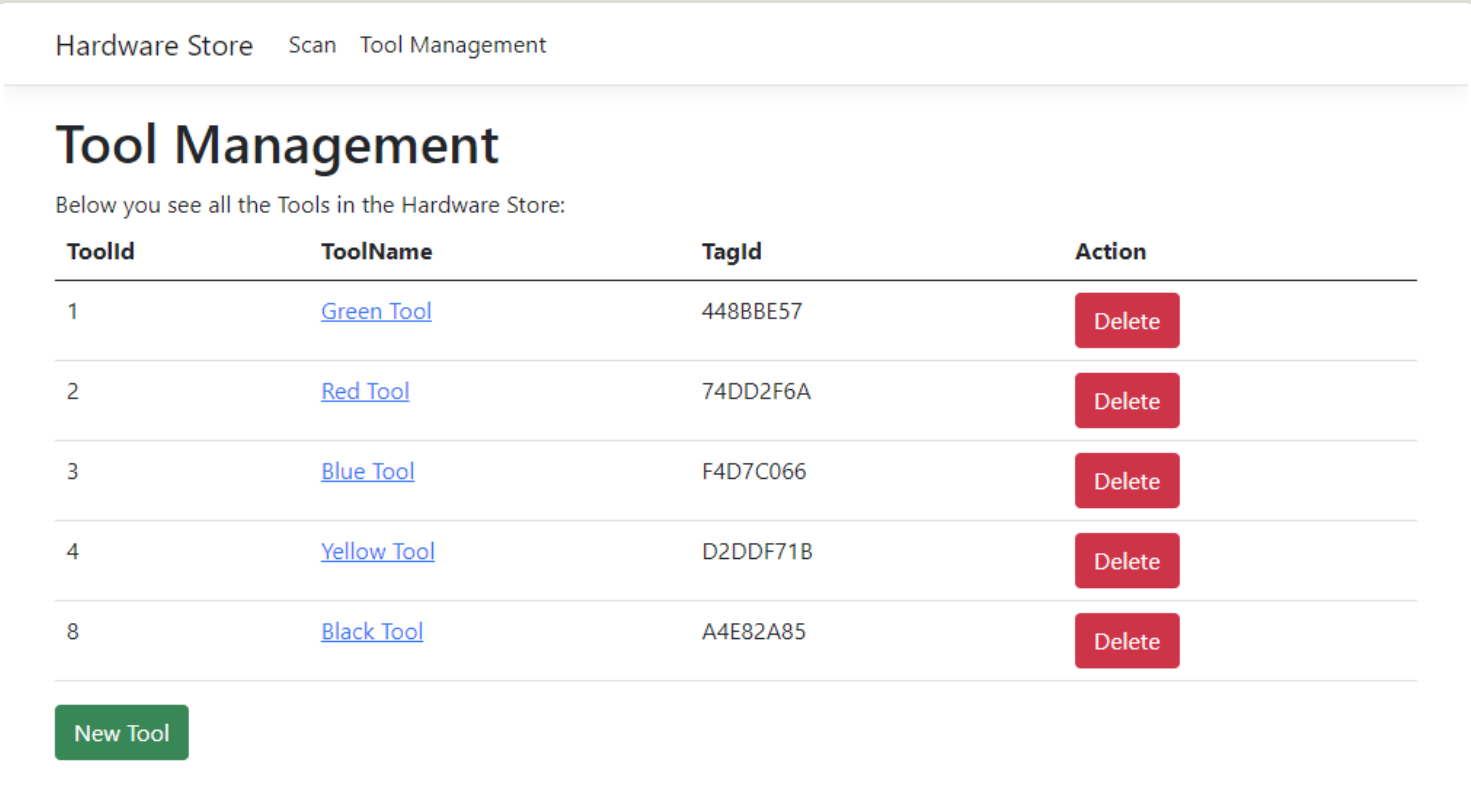
Hans-Petter Halvorsen

Table of Contents

Step by Step

1. Scanning RFID Tags into a Textbox
2. Retrieve Information based on the scanned Tag
3. Retrieve Information from Database based on the scanned Tag
4. Get List of available Tools in the Database
5. Add New Tools into the Database
6. Edit Existing Tool in the Database
7. **Delete Existing Tool in the Database**

ASP.NET Application



Database

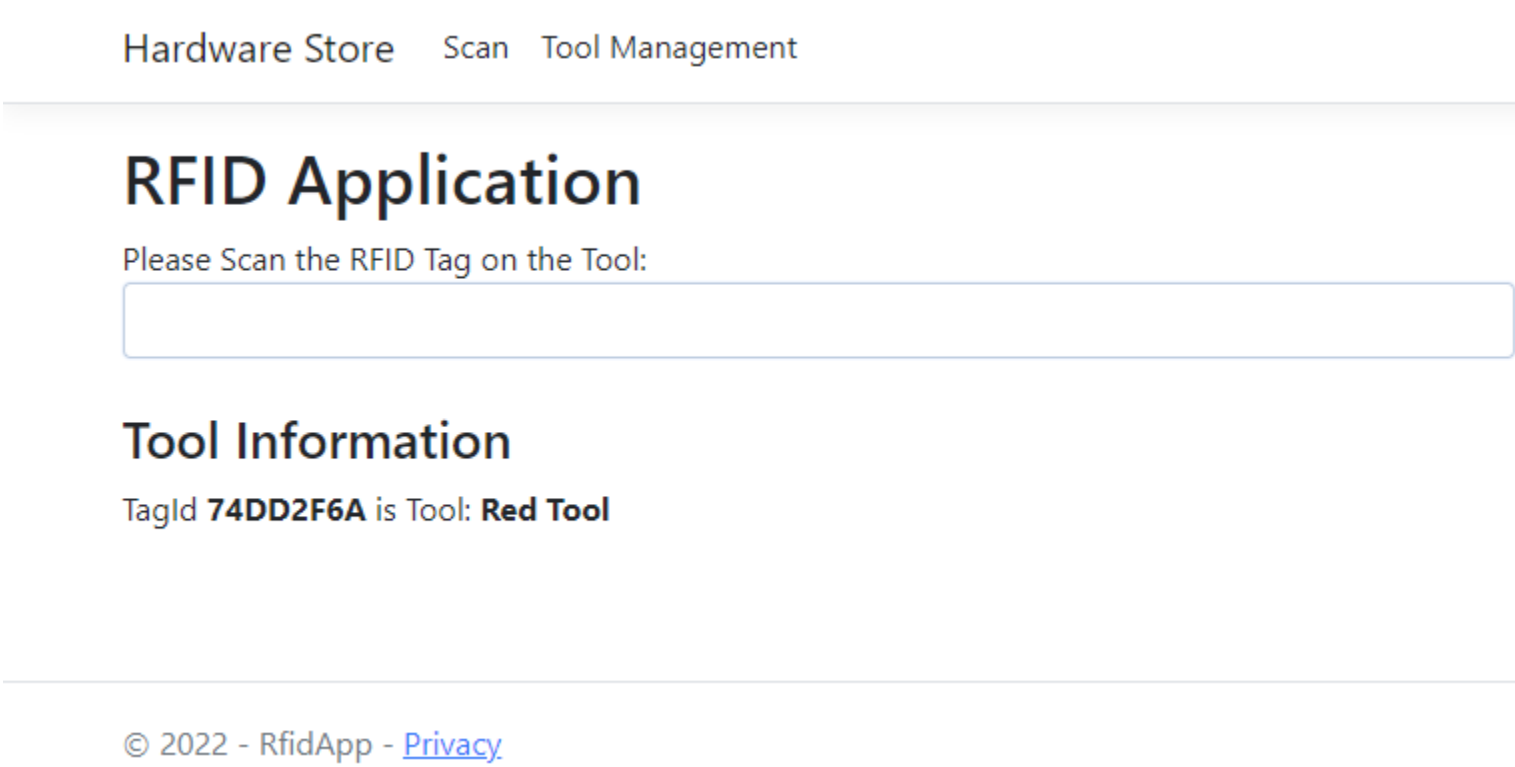
```
CREATE PROCEDURE DeleteTool
@ToolId int
AS

DELETE FROM TOOL WHERE ToolId=@ToolId

GO
```

Final Application

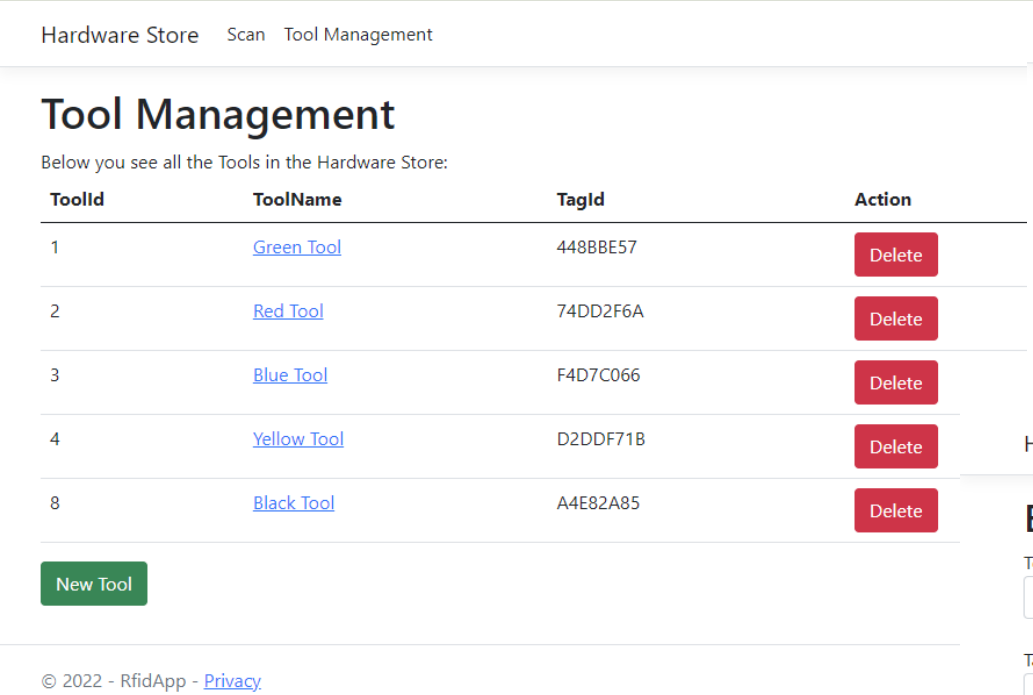
Hans-Petter Halvorsen

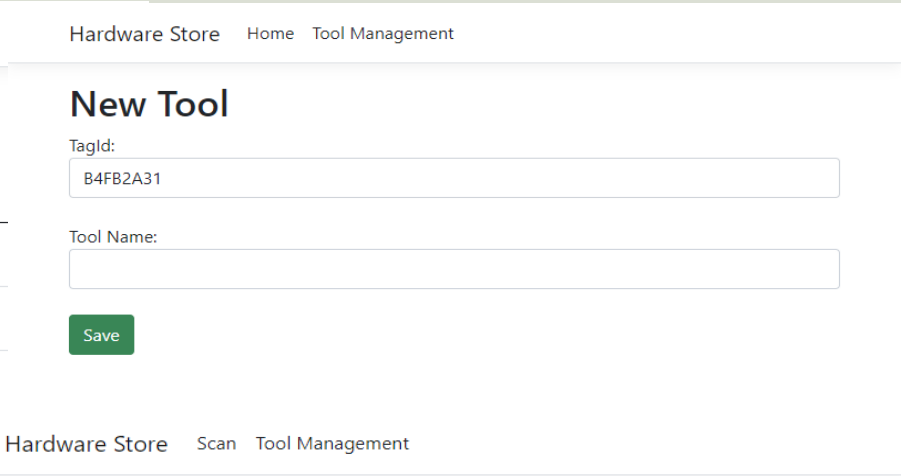
Table of Contents

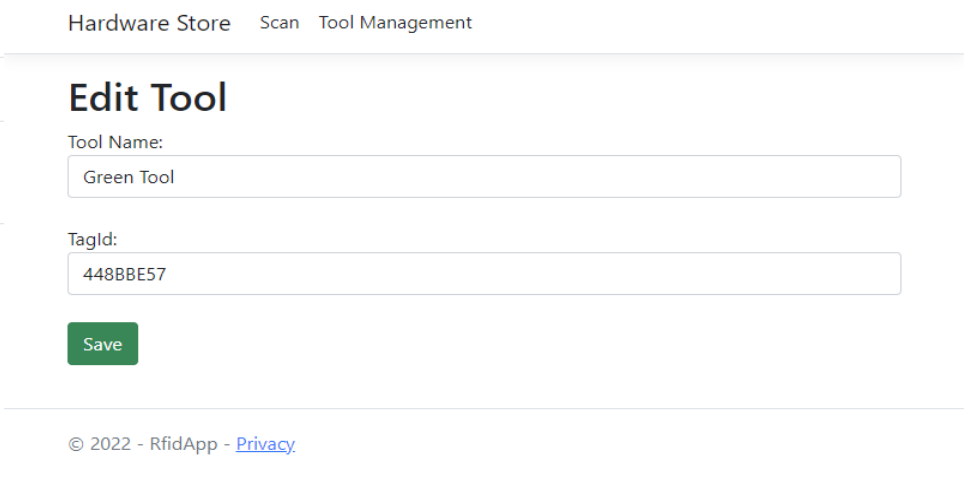
ASP.NET Application



ASP.NET Application







Summary

- We have made a basic RFID ASP.NET Core Application.
- It is important to take it step by step, i.e., add more and more features slowly step by step.
- Start simple then add more and more features.
- In that way it is so much easier to make it work and test it before you take the next step.
- Make sure that you always have a working Application before you take the next step.
- If you run out of time before, you can finish all the planned steps. Then you can always use the Application as it is.

Steps in the Development of this Application:

1. Scanning RFID Tags into a Textbox
2. Retrieve Information based on the scanned Tag
3. Retrieve Information from Database based on the scanned Tag
4. Get List of available Tools in the Database
5. Add New Tools into the Database
6. Edit Existing Tool in the Database
7. Delete Existing Tool in the Database

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog