**Vaccine and Patient Management Web Application**

*A Full-Stack ASP.NET Core MVC Web Application for Efficient Hospital Vaccine and Patient Management*

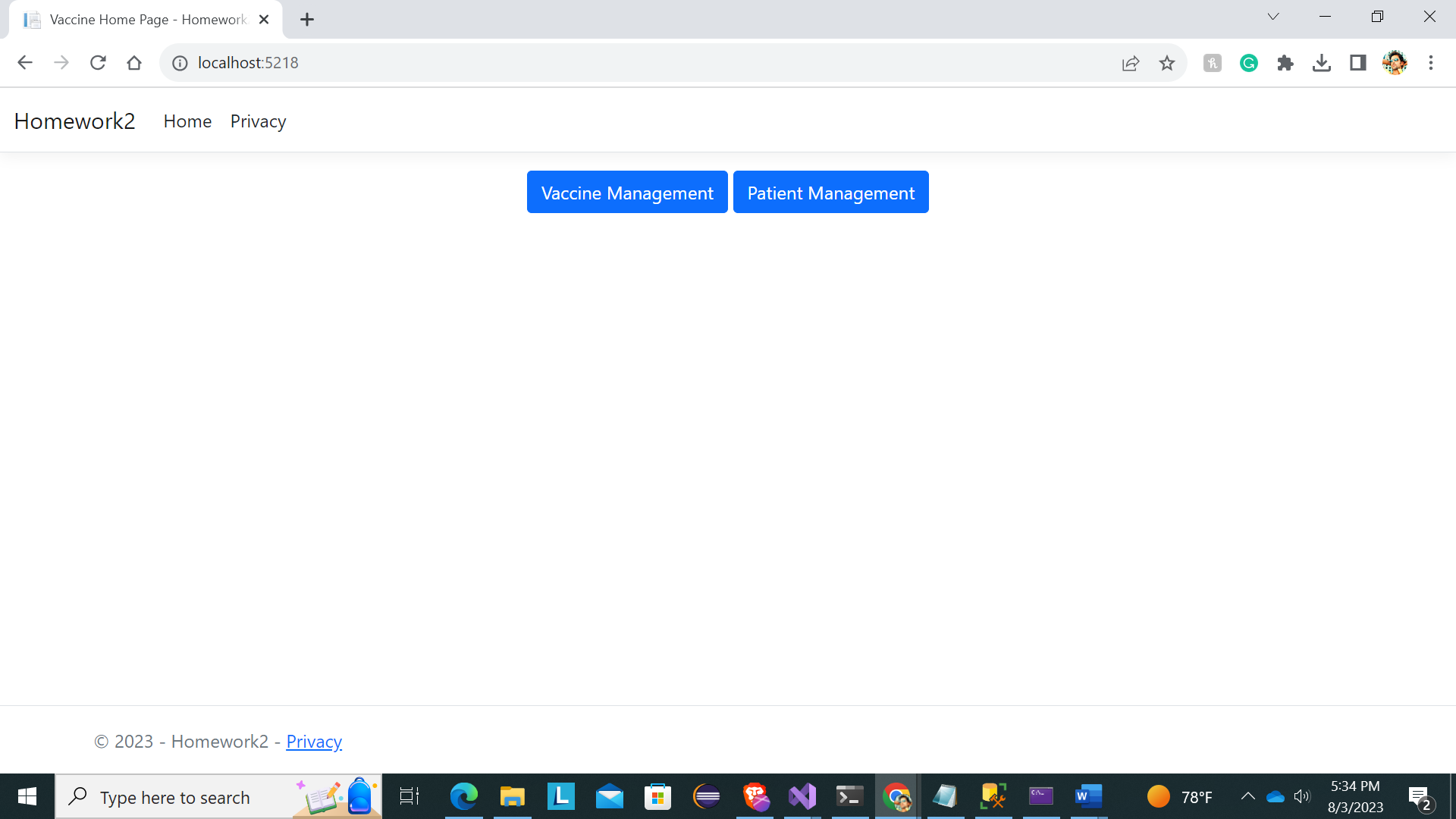
**Introduction**

This repository contains the source code and files for the Vaccine and Patient Management System, an ASP.NET Core MVC web application. The system is designed to manage vaccines and patients in a hospital setting. It allows hospital staff to track available vaccines, administer doses, and keep a record of patients and their vaccination status.

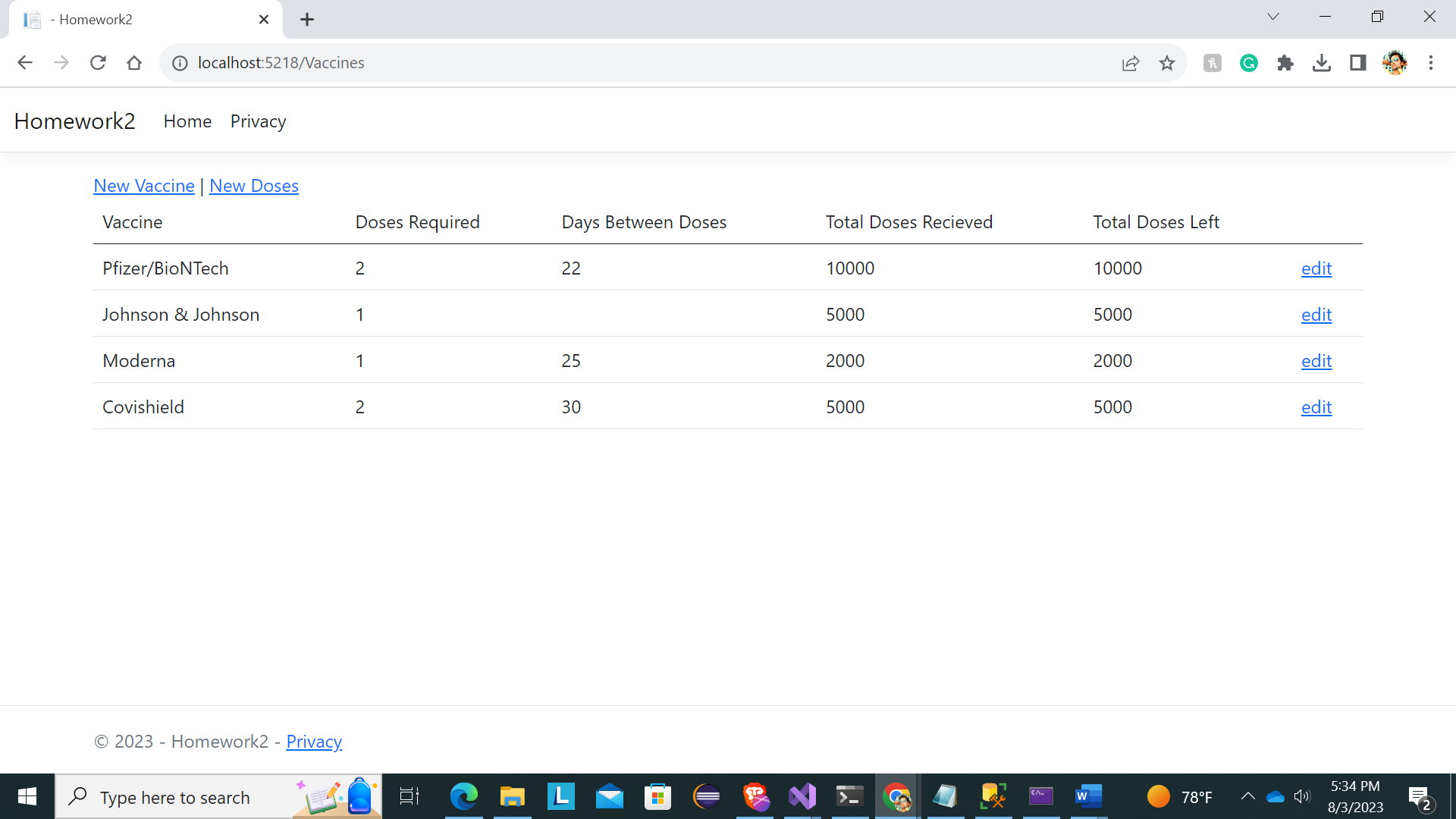
**Features**

The Vaccine and Patient Management System has the following key features:

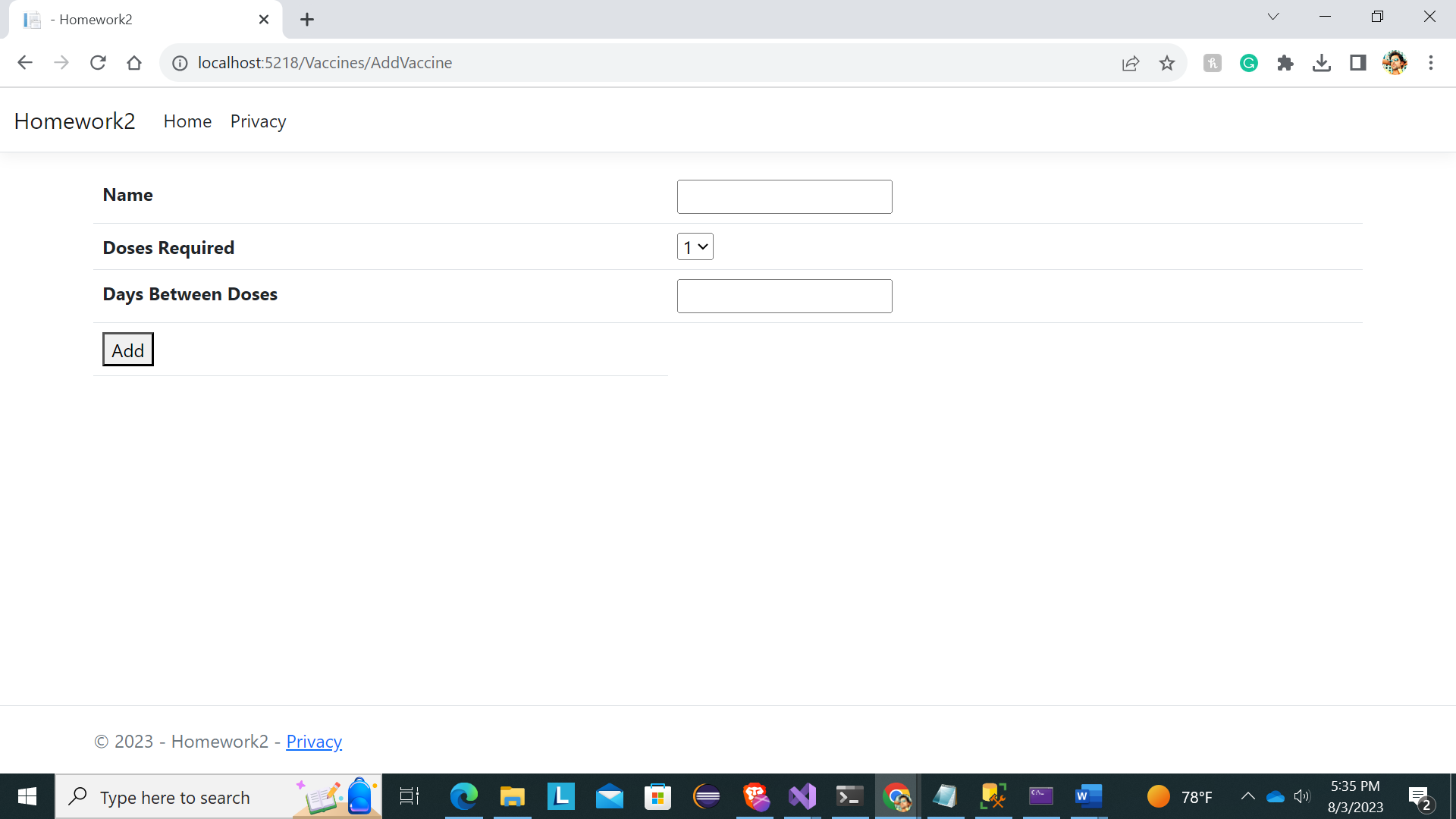
**Front Page**: The application's front page presents the user with two options: "Vaccine Management" and "Patient Management." Clicking on "Vaccine Management" takes the user to the vaccine listing page, while "Patient Management" takes the user to the patient listing page.



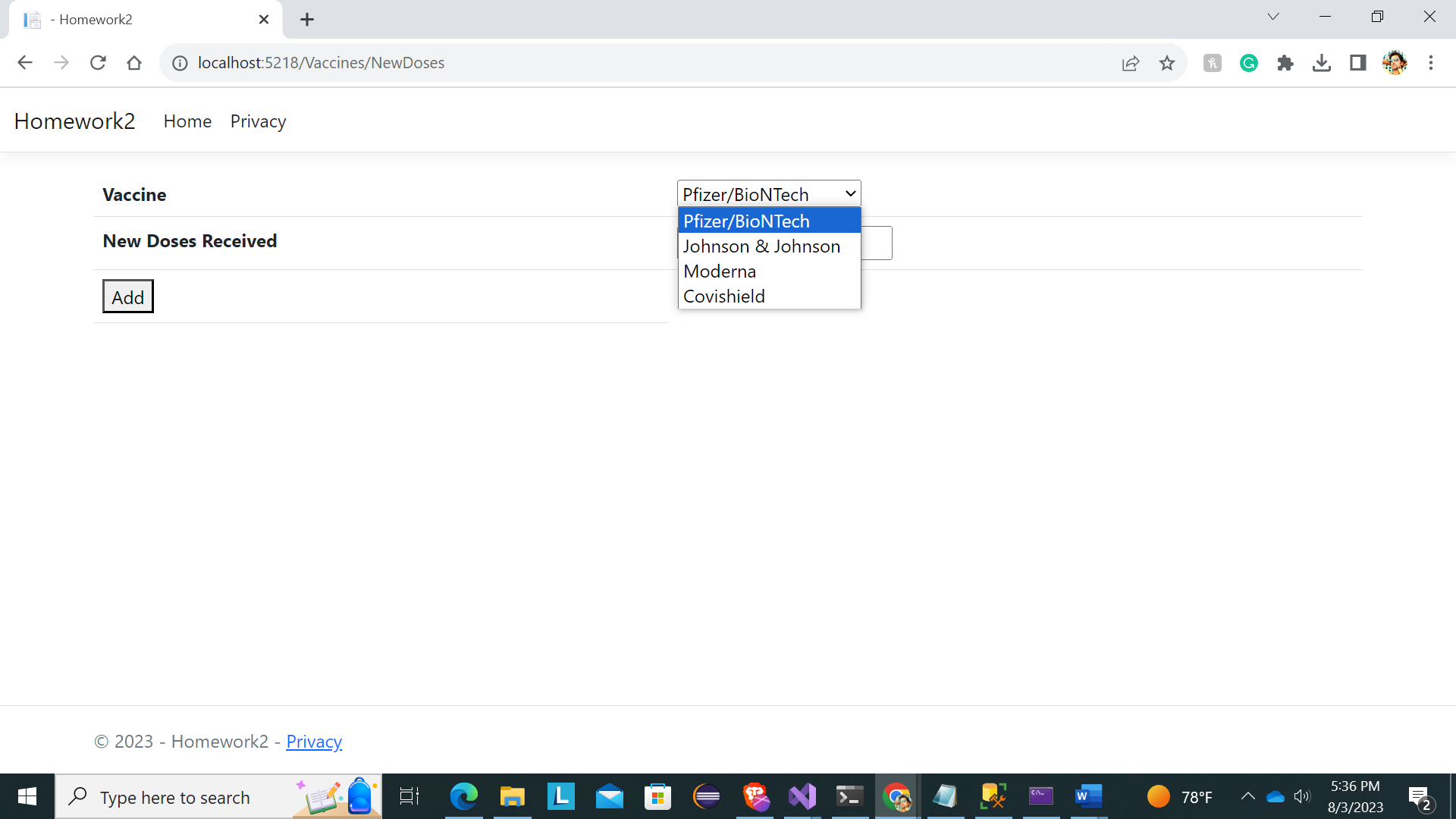
**Vaccine Management**



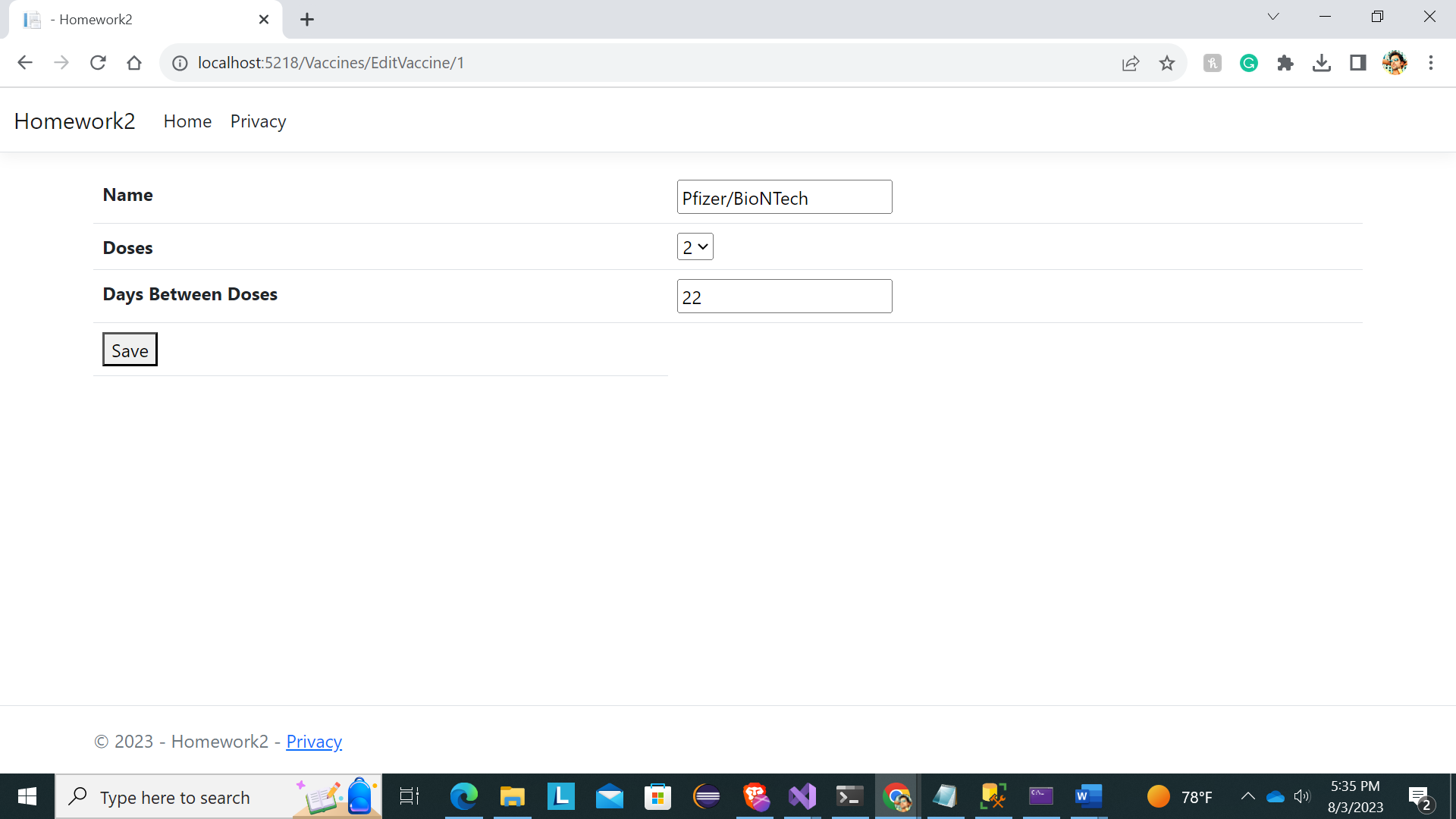
1. **List Vaccines**: The vaccine listing page displays a table with the vaccines available at the hospital. For each vaccine, the following information is shown: vaccine name, doses required, days between doses (if applicable), total doses received, total doses left, and an "Edit" link to edit vaccine details.
2. **New Vaccine**: Hospital staff can add a new type of vaccine to the system by clicking on "New Vaccine." They will be redirected to a page where they can enter the new vaccine's name, required doses, and days between doses (if applicable). After adding a new vaccine, it will appear on the vaccine listing page with a total of 0 doses received and left until new doses are added.



1. **New Doses**: If the hospital receives a new shipment of vaccine doses, staff can add them to the system by clicking on "New Doses." They will be taken to a page where they can select the vaccine and enter the number of new doses received. The system dynamically generates the list of available vaccines based on what is currently in the system. After adding new doses, the vaccine listing page will be updated with the new total doses received and left.

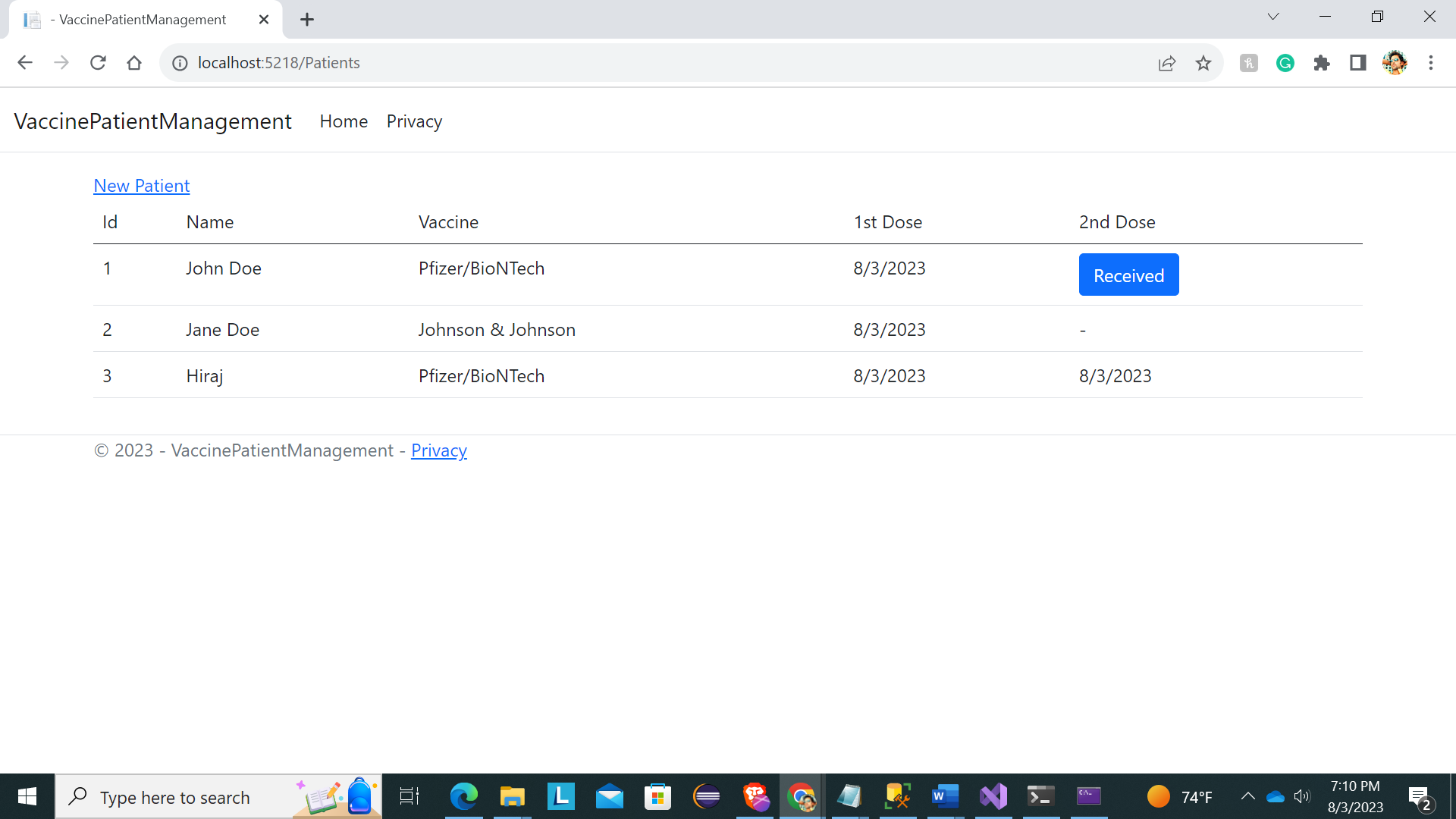


1. **Edit Vaccine**: Clicking on the "Edit" link of a vaccine allows users to update vaccine details, such as the name and required doses. After editing a vaccine, the user is redirected to the vaccine listing page.

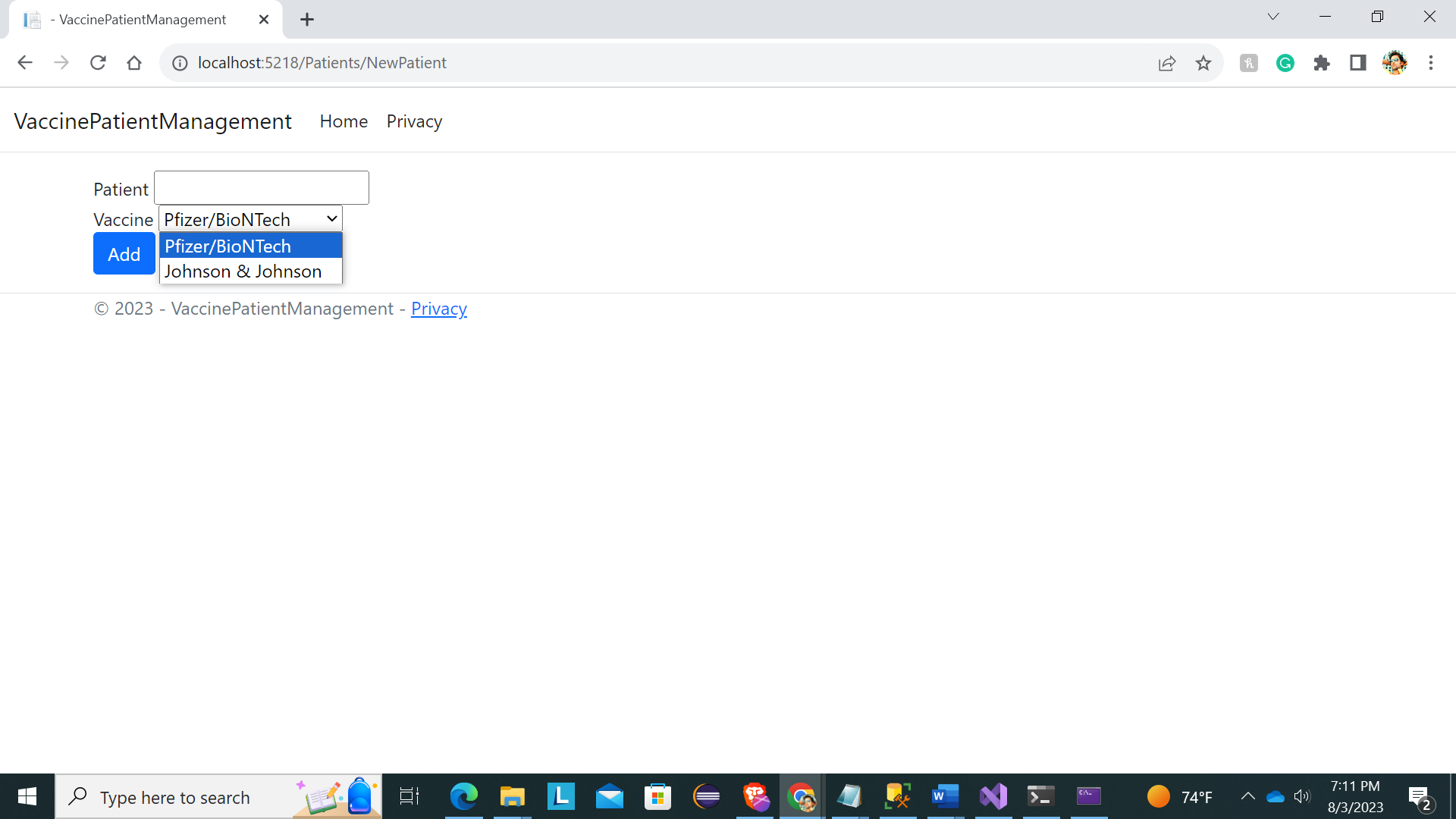


**Patient Management**

1. **List Patients**: The patient listing page displays a table with information about patients who received vaccinations. Each patient is identified by a unique ID and is associated with the vaccine they received, the date of their first dose, and the date of their second dose (if applicable). The "2nd Dose" column displays either the date of the second dose or "Out of Stock" if the vaccine requires two doses and no more doses are available.



1. **Receive 2nd Dose**: Clicking on the "Received" link (or button) for a patient indicates that they have received their second dose on that day. The system will decrement the stock of the vaccine by 1 and set the date of the second dose for the patient to the current date.
2. **Add Patient (for 1st Dose)**: Clicking on "New Patient" takes the user to a page where they can add a new patient to the system. The form asks for the patient's name and the vaccine to be given to the patient. The dropdown list for the vaccine is dynamically generated based on the available vaccines. After adding a new patient, the system generates a unique ID, decrements the stock of the selected vaccine by 1, and sets the date of the first dose for the patient to the current date.



**Technology Used and Database Access**

The Vaccine and Patient Management System is built using the following technologies:

* **ASP.NET Core MVC**: The application is developed using ASP.NET Core MVC framework, which allows for creating scalable and maintainable web applications.
* **C#**: The primary programming language used for building the application's backend logic.
* **Entity Framework Core (EF Core)**: EF Core is utilized for database access and management. It provides an object-relational mapping (ORM) framework, allowing us to work with the MS SQL Server database using C# objects and LINQ queries.
* **MS SQL Server**: The application's data is stored in an MS SQL Server database. EF Core handles the communication between the application and the database, abstracting away the low-level SQL queries.
* **LINQ (Language Integrated Query)**: LINQ is used in conjunction with EF Core to perform queries against the database using C# code, providing a strongly-typed and convenient way to interact with the data.

**Database Access with Entity Framework Core**

Entity Framework Core simplifies data access by automatically generating database tables based on the model classes. In this project, we have two main model classes: **Patient** and **Vaccine**, representing patients' data and vaccine information, respectively.

To interact with the database, we use EF Core's **DbContext**, which is responsible for defining the entities and their relationships, as well as configuring the connection to the database. The **DbContext** class, named **AppDbContext**, is located in the **Services** folder within the project.

The **DbSet** properties, **Patients** and **Vaccines**, represent the collections of **Patient** and **Vaccine** entities, respectively. With these DbSet properties, we can perform CRUD operations on the corresponding database tables.

Entity Framework Core allows us to perform various operations on the database using LINQ queries, making it easy to insert, update, delete, and retrieve data.

By leveraging the power of Entity Framework Core, we can focus on building application logic and let EF Core handle the underlying database interactions. This significantly reduces boilerplate code and accelerates development.

**How to Use**

1. Clone this repository to your local machine.
2. Open the project in Visual Studio or any preferred .NET Core development environment.
3. Set up the MS SQL Server database connection by providing your connection string in the appsetings.json file and link it to the AppDbContext.cs file.
4. Run the application and access it through your web browser.

**Future Improvements**

While the current version of the Vaccine and Patient Management System provides essential functionalities, there are several areas where improvements can be made:

* Enhance the user interface and user experience with more interactive and visually appealing elements.
* Add authentication and authorization features to restrict access to certain functionalities based on user roles.
* Implement data validation and error handling to ensure data integrity and enhance user feedback.
* Provide real-time updates for vaccine stock and patient status using modern web technologies like SignalR