Given that you're a beginner with a month to complete this project using ASP.NET Core MVC and the Identity library, here's a realistic timeline broken down into weeks. This timeline assumes you can dedicate significant time each week to learning and coding. The tasks are organized into a logical sequence, focusing on core features first and gradually building upon them.

**Week 1: Setup and Core Functionality (Identity Integration & Home Page)**

**Day 1-2: Initial Setup**

* **Install .NET Core SDK and Visual Studio Community** if not already done.
* **Create the project** in Visual Studio using ASP.NET Core MVC.
* **Install Identity NuGet Packages** for user authentication (e.g., Microsoft.AspNetCore.Identity.EntityFrameworkCore).
* **Set up the database connection** and configure Entity Framework Core.
* **Generate Initial Identity Models**: User and Role (you’ll use Identity for login functionality).

**Day 3-5: Home Page and Registration**

* **Create Home Page (US\_001)**: Add the layout with links/buttons for **Log In**, **Register**, **About Us**, and **Contact**.
* **Implement Registration (US\_001)**: Create the registration page using Identity, ensuring new users can register (save to database).

**Day 6-7: Log In and Log Out**

* **Implement Log In (US\_002)**: Create the log-in page using Identity and ensure users can log in with the credentials stored in the database.
* **Implement Log Out (US\_003)**: Ensure users can log out from the dashboard and return to the home page.

**Week 2: Voyager Dashboard and Profile Management**

**Day 8-10: Voyager Dashboard (US\_004)**

* **Create the Voyager Dashboard**: Implement a dashboard where users can access options like **Profile Management**, **Facility Booking**, **Room Booking**, and **Booking History**.
* **Ensure Only Logged-in Users Access Dashboard**: Use [Authorize] attribute to restrict access to authenticated users only.

**Day 11-13: Profile Management (US\_005)**

* **Create Profile Management Page**: Implement a page where users can view and update their profile (name, email, phone number).
* **Test Profile Updates**: Ensure the updates are reflected in the database and display the updated information after submission.

**Day 14: Basic Testing**

* **Test Registration and Login Flows**: Ensure the user can register, log in, log out, and manage their profile.
* **Fix bugs** related to user management (e.g., incorrect redirects or form submission issues).

**Week 3: Facility and Room Booking**

**Day 15-17: Facility Booking (US\_006)**

* **Create Facility Booking Page**: Implement the facility booking page where users can choose a facility, date, and time.
* **Add Booking Confirmation**: Once the user books a facility, show a confirmation message and store the booking details in the database.

**Day 18-20: Room Booking (US\_008)**

* **Create Room Booking Page**: Implement a similar booking page for rooms, where users can select room type, dates, and preferences.
* **Confirm Room Booking**: Once the user confirms a room booking, store the booking details and show a confirmation message.

**Day 21: Bill Generation (US\_011)**

* **Create Bill Generation Page**: Based on the bookings, generate a bill for the user. Display all charges related to facilities and rooms booked.
* **Allow Bill Download**: Allow the user to download the bill (e.g., in PDF format).

**Week 4: Admin Functionality and Final Testing**

**Day 22-24: Admin Functionality (Add/Edit/Remove Facilities)**

* **Add Facilities (US\_012)**: Create a page where admins can add new facilities with name, type, availability, and fees.
* **Edit Facilities (US\_013)**: Implement the ability for admins to edit existing facilities.
* **Remove Facilities (US\_014)**: Allow admins to delete facilities from the system.

**Day 25-27: Admin Viewing (Facility Booking History & Bill History)**

* **View Facility Booking History (US\_016)**: Implement a page where admins can view all bookings made by voyagers.
* **View Bill History (US\_017)**: Create a page where admins can view all generated bills for voyagers.

**Day 28: Final Testing & Debugging**

* **Test All User Stories**: Perform end-to-end testing for all user stories, ensuring that each functionality (e.g., registration, login, booking, profile management) works as expected.
* **Fix Bugs**: Debug and fix any issues that arise during testing (e.g., validation issues, incorrect database entries, missing functionality).

**Day 29-30: Deployment and Documentation**

* **Deploy the Application**: Deploy the project to a cloud platform (e.g., Azure or Heroku).
* **Prepare Documentation**: Write simple documentation for how to run the project, key features, and setup instructions.

**Key Considerations for Success:**

* **Use Templates and Libraries**: Since you're a beginner, use templates or libraries like **Bootstrap** for UI components and styling. Identity provides much of the login functionality out-of-the-box, which will save you time.
* **Break Down Tasks**: If you get stuck, break down complex tasks into smaller chunks and focus on one thing at a time.
* **Seek Help and Learn**: Don’t hesitate to consult documentation, online tutorials, or community forums (e.g., StackOverflow) when needed.

By following this timeline and breaking the project into manageable chunks, you can complete it within a month, even as a beginner. Be sure to keep the project simple and focus on the core functionalities first, adding more complexity as you go.