1. Prerequisites

Before you begin, ensure you have the following:

- A company-provided laptop or an approved personal device.
- Administrative rights to install software.
- A stable internet connection.

---

2. Install Required Software

2.1 .NET SDK

We utilize the latest Long-Term Support (LTS) version of the .NET SDK for our projects.

1. **Download the .NET SDK:**
    - Visit the official .NET download page: https://dotnet.microsoft.com/download/dotnet
    - Select and download the latest LTS version suitable for your operating system.
2. **Install the SDK:**
    - Follow the installation prompts specific to your OS.
3. **Verify the Installation:**
    - Open a terminal or command prompt.

Run:
bash
CopyEdit

```
dotnet --version
```

    -
    - Ensure the version displayed matches the LTS version you installed.

2.2 Integrated Development Environment (IDE)

We recommend using **Visual Studio 2022** for ASP.NET development.

1. **Download Visual Studio 2022:**
    - Access the installer from: https://visualstudio.microsoft.com/vs/
2. **Install Visual Studio:**
    - During installation, select the following workloads:

        - **ASP.NET and web development**
        - **Azure development** (if applicable)
        - **.NET Core cross-platform development**
3. **Configure the IDE:**
    - Set the default environment settings to "Web Development."
    - Sign in with your company-provided Microsoft account to sync settings and access additional resources.

2.3 Version Control System

We use **Git** for version control and **Azure Repos** for repository management.

1. **Install Git:**
    - Download and install from: https://git-scm.com/
2. **Configure Git:**

Set your user name and email:
bash
CopyEdit

```
git config --global user.name "Your Name"
git config --global user.email "your.email@sunlife.com"
```

    -
3. **Authenticate with Azure Repos:**
    - Use your company credentials to access repositories.
    - Follow internal documentation for setting up authentication methods, such as SSH keys or personal access tokens.

2.4 Database Management Tool

For database interactions, we use **SQL Server Management Studio (SSMS)**.

1. **Download SSMS:**
    - Obtain the installer from: https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms
2. **Install SSMS:**
    - Follow the installation prompts to complete the setup.
3. **Configure Database Connections:**
    - Connect to the development database using credentials provided by your team lead.

2.5 Containerization Platform

We employ **Docker** for containerization to ensure consistent development and deployment environments.

1. **Install Docker Desktop:**
    - Download from: https://www.docker.com/products/docker-desktop
2. **Configure Docker:**
    - Ensure Docker is set to start with your system.

Verify the installation by running:
bash
CopyEdit
```
docker --version
```

    -

2.6 Package Manager

We use **NuGet** for managing project dependencies.

1. **Access NuGet:**
    - NuGet is integrated into Visual Studio. You can manage packages via the NuGet Package Manager within the IDE.
2. **Configure Package Sources:**
    - Add any internal package sources as specified in the internal documentation.

---

3. Project-Specific Setup

3.1 Clone the Repository

1. **Access the Repository:**
    - Navigate to the project's repository in Azure Repos.
2. **Clone the Repository:**

Use the following command:
bash

CopyEdit

```
git clone <repository-url>
```

-
- Replace `<repository-url>` with the actual URL of the repository.

Navigate to the Project Directory:
bash
CopyEdit

```
cd <project-directory>
```

3.

3.2 Restore Dependencies

1. **Restore NuGet Packages:**
   - In Visual Studio, right-click on the solution in the Solution Explorer.
   - Select "Restore NuGet Packages."
2. **Build the Solution:**
   - Press `Ctrl + Shift + B` to build the entire solution and ensure all dependencies are correctly installed.

3.3 Environment Configuration

1. **Environment Variables:**
   - Set the following environment variables as per the project's requirements:
     - `ASPNETCORE_ENVIRONMENT`: Set to `Development`.
     - `ConnectionStrings__DefaultConnection`: The connection string for the development database.
2. **User Secrets (Optional):**

If the project uses User Secrets, initialize and configure them:
bash
CopyEdit

```
dotnet user-secrets init
dotnet user-secrets set "Key" "Value"
```

-

---

4. Additional Tools

Depending on your project's requirements, you might need the following tools:

- **Postman:** For API testing.
  - Download from: https://www.postman.com/downloads/
- **Azure CLI:** For managing Azure resources.
  - Install from: https://docs.microsoft.com/en-us/cli/azure/install-azure-cli
- **Node.js and npm:** If your project includes frontend components.
  - Download from: https://nodejs.org/

---

5. Documentation and Resources

- **Sun Life Internal Documentation:** Access internal wikis and Confluence pages for detailed project documentation and best practices.
- **Microsoft Documentation:**
  - .NET Documentation
  - ASP.NET Core Documentation
- **Team Communication:** Join the relevant Slack channels or Microsoft Teams groups as directed by your mentor.

---

6. Support

If you encounter any issues during the setup process:

1. **Consult the Internal Wiki:** Many common issues and their solutions are documented.
2. **Contact Your Mentor:** They are your first point of contact for any onboarding-related queries.
3. **IT Support:** For hardware or network-related issues, reach out to the IT support team.

Once your environment is set up, please inform your mentor to proceed with the next steps in your onboarding process. Welcome aboard, and we look forward to your contributions to the team!