myEHR

Code Maintenance Documentation

Mid-state IT SOFTWARE DEVELOPER capstone

2024

Table of Contents

Description of Documentation 2

Resources 2

Source Control 2

Hosting 4

Development Environment Setup 6

Other [Important Credentials](#_Toc1338920470) 8

# Description of Documentation

The code maintenance documentation explains code snippets and provides guidance on how to maintain the web application and database. This will be made available to those who are responsible for maintaining the project upon completion.

# Resources

## Source Control

1. Introduction

2. GitHub Collaboration

* 1. Create a GitHub Repository
  2. Pull Requests
  3. Peer Reviews

3. Best Practices

3.1 Commit Messages

3.2 Git Ignore

**1. Introduction**

This section provides detailed information regarding version control of the project content on GitHub.

**2. GitHub Collaboration**

**2.1 Create a GitHub Repository**

Retrieve the source code from the remote repository and save it to the main branch of your new repository.

Source control for myEHR can be found in a GitHub repository at the following address:

<https://github.com/kaitlinrwheeler/myEHR>

**2.2 Pull Requests**

Create a pull request to submit changes to the dev branch of the remote repository:

* Create a new branch before making any changes. The branch name should indicate the task being worked on using the task’s name, description, or other unique identifier.
* Commit your changes to the new branch as you go.
* When your task is complete and has been tested, merge the dev branch into the task’s working branch and review any merge conflicts that may arise.
* Create a pull request on GitHub for another team member to peer review.

**2.3 Peer Reviews**

Team members will thoroughly review each other’s pull requests to ensure that the code works as intended without any unexpected detriments.

* Checkout the branch that has been submitted for review.
* Pull the latest commit changes on that branch.
* Run the application to ensure that the solution builds properly.
* Thoroughly test the results of the task at hand.
* Review the code that was altered or added.
  + Commits can be reviewed in GitHub when the pull request has been selected.
  + When you select a commit, GitHub will display the original code and the updated code side by side, highlighting the changes. This will be a useful tool when conducting peer reviews.
* Create a list to document any errors, typos, etc. that are encountered during the review process. The list should be detailed enough to explain exactly what the problems are and, if possible, where the problems can be seen.
* Return the list of errors to the team member who created the pull request so they can fix things as needed.
* This process will be repeated until the branch is ready to be merged with the rest of the project.
* Ensure that the pull request is set to merge the working branch with the dev branch – not the main branch.

**3. Best Practices**

**3.1 Commit Messages**

Commit messages should explain what was changed or worked on in the code being submitted. These messages should be clear and concise.

**3.2 Git Ignore**

Maintain a .gitignore file to exclude unnecessary files and directories. This will make each commit easier to manage for reviewers.

## Hosting

**Azure Hosting Documentation**

1. Introduction

2. Azure Account Credentials

2.1 Azure Subscription Account Credentials

2.2 myEHR Azure Database Credentials

3. Publishing myEHR Updates

3.1 Visual Studio Account

3.2 Publish Updates

4. Troubleshooting Deployed Code

4.1 Visual Studio Production Mode

4.2 Azure Troubleshooting Guide

**1. Introduction**

This section provides guidance on maintaining the application which is hosted on the Azure cloud platform. It includes credentials for accessing the myEHR Azure account and the database.

**2. Azure Account Credentials**

**2.1 Azure Subscription Account Credentials**

Username: MSTCmyEHR@outlook.com

Password: myEHR@123!

These credentials will also grant you access to this outlook email account.

**2.2 myEHR Azure Database Credentials**

Username: myEHRAdmin

Password: myEHR@123!

These credentials will grant you access to the database administrator role on Azure. This will be needed whenever the database needs updating.

**3. Publishing myEHR Updates**

**3.1 Visual Studio Account**

Log in to Visual Studio using the Azure account credentials. This will allow you to link your local code to the Azure account.

**3.2 Publish Updates**

* Check out the project’s main branch from the remote repository.
* Pull the most recent commit.
* Run the application to ensure that the solution builds and operates as expected.
* In Visual Studio’s Project tab, select Overview.
* From the Overview view, select Publish from the left side navigation menu.
* Click the Publish button at the top of the Publish view to deploy the current code from the project’s main branch.
  + This part of the process may take a few minutes to complete.
  + Once the updated code has been published, the Azure site may take an additional few minutes to update as well.

**4. Troubleshooting Deployed Code**

**4.1 Visual Studio Production Mode**

If errors are encountered in the deployed project, following these steps and recreating the actions that lead to the error can aid in troubleshooting efficiently.

* Checkout the main branch from the remote repository. Do not pull changes as you want to ensure you are testing the same code that has been deployed.
* Click the Project tab in the top menu and select Project Properties.
* From the left side navigation menu, select Debug.
* Open the debug launch profiles UI and locate Environment variables.
* Find ASPNETCORE\_ENVIRONMENT and change the value from ‘Development’ to ‘Production’.
  + If you do not see the name ASPNETCORE\_ENVIRONMENT, add it to the next empty field under Environment variables in the UI.
* Close the UI and run the application in debug mode.
* Recreate the actions that lead to the error. The application should respond in the same way that it did on the deployed site while offering insight into what caused the problem. If this does not work, see the troubleshooting guide in 4.2.

**4.2 Azure Troubleshooting Guide**

Refer to the [Troubleshooting Guide](https://learn.microsoft.com/en-us/troubleshoot/azure/) for common issues and their solutions.

## Development Environment Setup

**Development Environment Setup Guide**

1. Introduction

2. Requirements

3. Visual Studio Version Control

3.1 Git

3.2 Git Repository

4. Entity Framework Core Database Migrations

**1. Introduction**

This section provides guidelines for setting up an appropriate environment and the necessary tools to access and contribute to myEHR.

**2. Requirements**

Before you begin, ensure that you have the following tools installed on your machine:

* Operating System: Windows
* Visual Studio 2022
* SQL Server Management Studio (SSMS)

**3. Version Control**

**3.1 Git Changes**

Git is used for version control. The Git Changes tool in Visual Studio can be accessed from the View tab in the top menu or by using CTRL+0, CTRL+G on your keyboard. This feature will allow you to use all the necessary Git functionality. Git Changes shows you the branch you’re currently on and lists any documents that have been changed, added, or deleted since checkout. There is a commit message field where you can outline the changes made before submitting the commit. It also provides the ability to fetch, push, and pull code changes. Git Changes also offers the option to create a pull request within Visual Studio or in your web browser when a commit has been pushed.

* 1. **Git Repository**

The Git Repository view in Visual Studio can be accessed from the View tab in the top menu or by using CTRL+0, CTRL+R on your keyboard. In Git Repository, you can see a list of both local and remote branches as well as a list of commits and commit messages for the current branch. Right clicking on branches in the comprehensive list of local and remote branches reveals a menu with options for the following:

* Checkout the selected branch.
* Create a new branch.
* Merge or rebase a branch.
* Reset a branch.
* Delete a branch.
* Compare two branches.
* Fetch, push, and pull changes to a branch.
* Create a pull request.

**4. Entity Framework Core Database Migrations**

Update the database using Entity Framework Core migrations:

* Open the Package Manager Console in Visual Studio.
* When changes have been made to your models that need to be reflected in the database:
  + Run the command: Add-Migration migration-name
    - The migration name should be clear and concise, indicating the changes being made.
  + Review the migration (located in the auto-generated Migrations folder) to ensure that the changes outlined reflect the intended database changes and that additional changes are not made unintentionally.
* When the migration accurately details the changes to be made to the database:
  + Run the command: Update-Database
  + This may take a moment to complete. If the update is rejected, review the error message, and make any necessary changes to the migration file.
* If a migration has been added in error:
  + Run the command: Remove-Migration migration-name
  + Follow that command with the Update-Database command.

## Other Important Credentials

The following credentials grant admin access to myEHR.

**myEHR Admin Access**

**Login: EHRAdmin@mstc.edu**

**Password: !EHRAdmin12!**

Please be sure to review and update this documentation as configurations or best practices evolve.