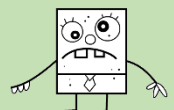
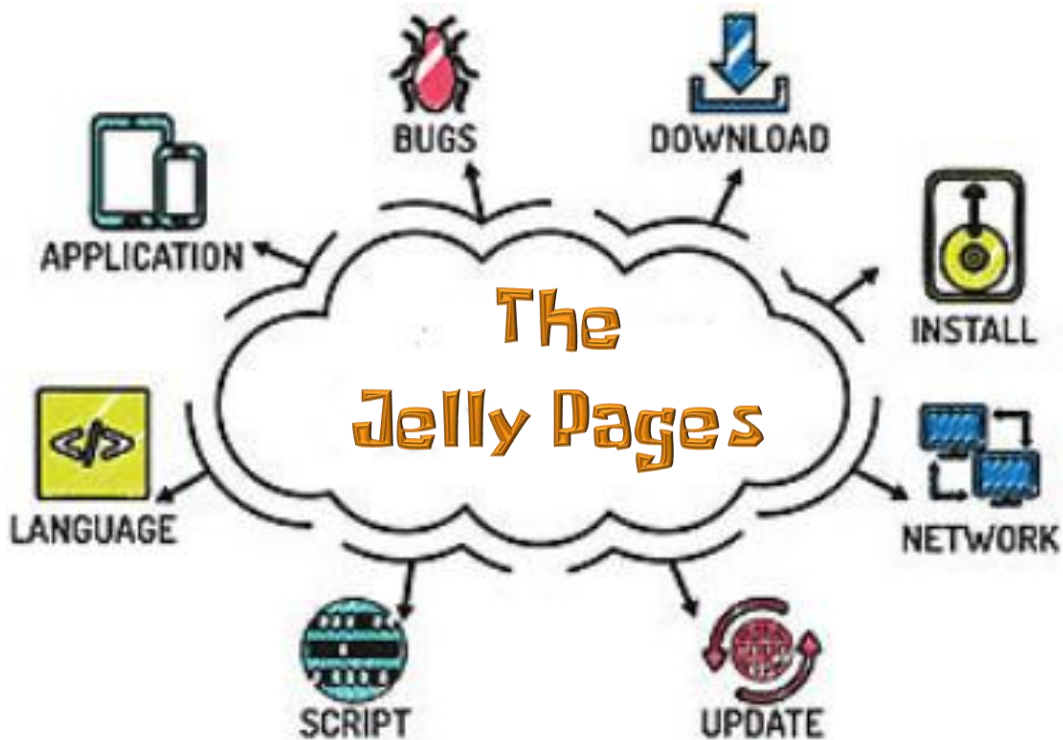




# TECHNICAL MANUAL



## TABLE OF CONTENTS

---

INTRODUCTION .....	2
SOFTWARE REQUIREMENTS.....	2
KNOWLEDGE REQUIREMENTS .....	2
PROCESS 1 < GIT /GITHUB > .....	3
1. CLONE THE REPOSITORY ONTO YOUR LOCAL COMPUTER. ....	3
PROCESS 2 < VISUAL STUDIO > .....	4
1. OPEN THE PROJECT ON YOUR LOCAL COMPUTER .....	4
2. CREATE YOUR OWN PAGE .....	4
3. ADD SCAFFOLDING FOR YOUR UI .....	4
4. ADDING NAVIGATION BUTTON TO YOUR PAGE .....	4
PROCESS 3 <PUSH TO GITHUB> .....	5
1. PULL ALL BRANCHES .....	5
2. COMPARE & PULL ON GITHUB .....	5
LAST REVISION DATE/VERSION .....	6

## INTRODUCTION

Welcome to the user manual for the Jelly Pages! This document is intended to provide you with all the information you need to get started with and use the software effectively.

The Jelly Pages has been designed to be intuitive, easy to use, and assist you in quickly getting in contact with other residents of Bikini Bottom.

## SOFTWARE REQUIREMENTS

- ✚ Visual Studio 2022
  - a. <https://visualstudio.microsoft.com/downloads/>
- ✚ GitHub Jelly Pages Repository
  - a. <https://github.com/mel348/JellyPages> (Jelly Pages App Repository)
- ✚ Discord Account
  - a. <https://discord.com/channels/1146910668440752148/1164021066948169840> (Jelly Pages App Thread)
- ✚ Trello Account
  - a. <https://trello.com/b/M9OW0IQi/sprint-4>
- ✚ Zoom Account
  - a. <https://zoom.us/signup#/signup> (Create an Account)
- ✚ Git
  - a. <https://gitforwindows.org/> (Git for Windows)
- ✚ Windows 10 or later

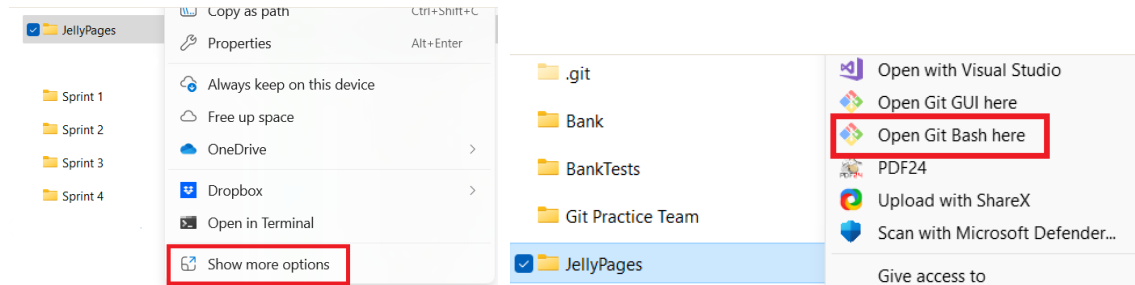
## KNOWLEDGE REQUIREMENTS

- ✚ C#
- ✚ ASP.NET Core RazorPages
- ✚ BootStrap 4
- ✚ MVVM

## PROCESS 1 < GIT /GITHUB >

### 1. CLONE THE REPOSITORY ONTO YOUR LOCAL COMPUTER.

- Create a folder on your local computer named JellyPages.
- Right click on the folder and select "show more options" and then click on "Open Git Bash here."



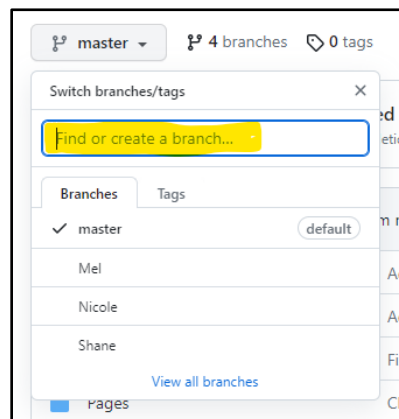
### c. Enter the following Git commands:

- "git init" – initializes the folder as a Git repository
- "git clone https://github.com/mel348/JellyPages"

```
Lisa@Lisa-B-Laptop MINGW64 ~/Desktop/Git Practice Team
$ git clone https://github.com/balbach1/git_practice
Cloning into 'git_practice'...
remote: Enumerating objects: 16, done.
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 16 (delta 3), reused 14 (delta 1), pack-reused 0
Receiving objects: 100% (16/16), done.
Resolving deltas: 100% (3/3), done.

Lisa@Lisa-B-Laptop MINGW64 ~/Desktop/Git Practice Team
$ cd git_practice
```

- Create a branch for yourself on GitHub in the repository.



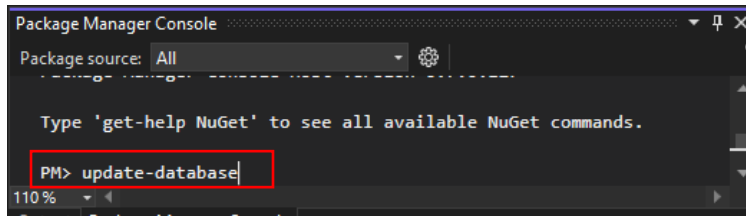
- Fetch all remote branches with Git Bash.
  - "git fetch origin"
  - "git checkout <your branch name>"

- "git pull" – You should now have a current version of the Jelly Pages app on your local computer.

## PROCESS 2 < VISUAL STUDIO >

### 1. OPEN THE PROJECT ON YOUR LOCAL COMPUTER

- a. Use "update-database" command with the package manager console to download the database.

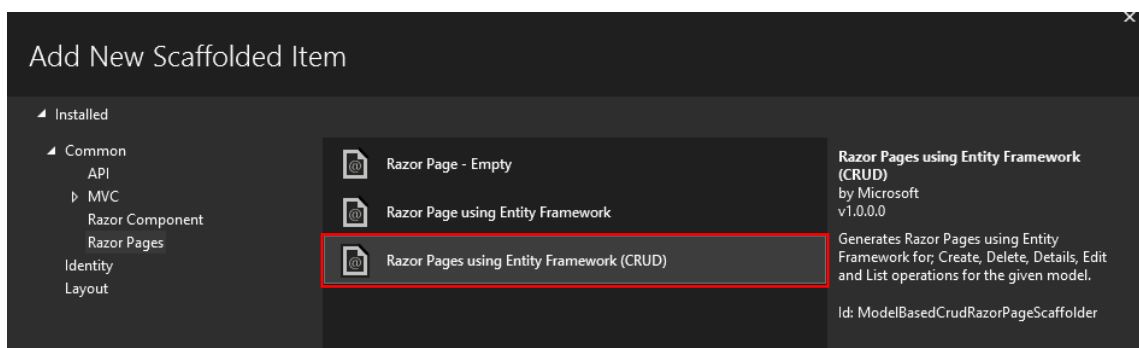


### 2. CREATE YOUR OWN PAGE

- a. Go to the "pages" folder, located in your solution explorer, and right click.
- b. Select "add" > "New Folder" (name the folder with your first name).

### 3. ADD SCAFFOLDING FOR YOUR UI

- a. Right click on the folder with your name and select "add new scaffolded item."
- b. Make sure the "Razor Pages using Entity Framework (CRUD)" is selected and click "add" at the bottom of the window.



- c. Select the model class, Data context, and click "add."
  - Model Class: Phonebook (JellyPages.Models)
  - DbContext class: JellyPagesContext(JellyPages.Data)

### 4. ADDING NAVIGATION BUTTON TO YOUR PAGE

- a. Go to "Shared" folder and open the \_Layout.cshtml
- b. Enter your navigation here like such:

```
<li class="nav-item">  
  <a class="btn btn-primary" asp-area="" asp-page="/yourName/Index">yourName</a>  
</li>
```

- c. "Save All" then go to "build" in the navigation bar > "build solution" > Run App to check for any errors.
- d. Your front end will start out looking like this:

<a href="#">Home</a> <a href="#">Mel</a> <a href="#">Nicole</a> <a href="#">Shane</a> <a href="#">yourName</a> <a href="#">Privacy</a>					
<h2>Index</h2> <a href="#">Create New</a>					
First Name	Last Name	Address	Phone	Email	
SpongeBob	SquarePants	124 Conch St.	776-4377	<a href="mailto:ssquarepants@bikinibottom.com">ssquarepants@bikinibottom.com</a>	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Squidward	Tentacles	122 Conch St.	778-4392	<a href="mailto:stentacles@bikinibottom.com">stentacles@bikinibottom.com</a>	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Sheldon	Plankton	1490 Bottom Feeder Lane	723-5366	<a href="mailto:splankton@bikinibottom.com">splankton@bikinibottom.com</a>	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Sandy	Cheeks	123 Treedome St.	726-3977	<a href="mailto:scheeks@bikinibottom.com">scheeks@bikinibottom.com</a>	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
Larry	Lobster	1 Muscle Beach	581-1956	<a href="mailto:llobster@bikinibottom.com">llobster@bikinibottom.com</a>	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>

- e. Once you've worked a little bit on your UI you can push your updates to your branch on GitHub and commit a merge with the main branch.

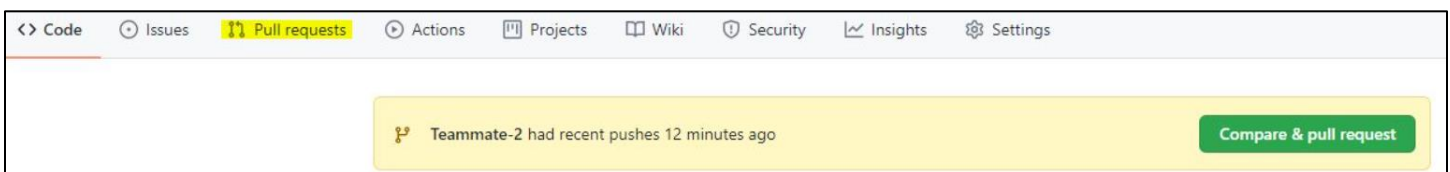
## PROCESS 3 <PUSH TO GITHUB>

### 1. PULL ALL BRANCHES

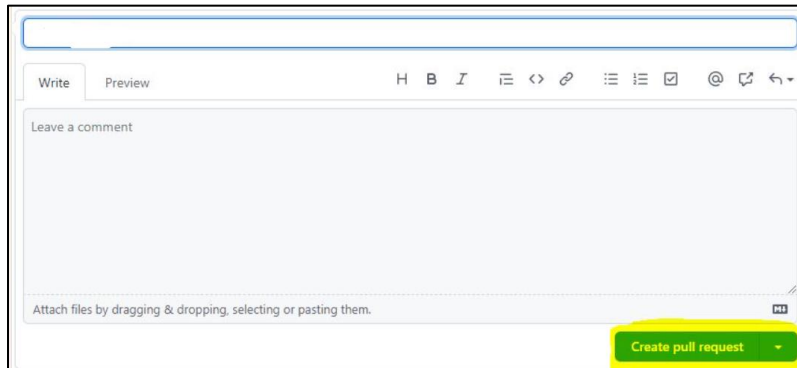
- a. In Git Bash type "git fetch origin" (pulls in all of the repository branches)
- b. "git branch -a" (to confirm all branches were pulled onto your local computer)
- c. "git checkout <your branch name>" (puts you on your branch)
- d. "git add ." (prepares your updated files to be staged)
- e. "git status" (should show "nothing to commit, working tree clean" letting you know the files were properly staged)
- f. "git commit -m "type a message" (this way teammates know what you changed or added)
- g. "git push" (pushes your changes to the repository on GitHub)

### 2. COMPARE & PULL ON GITHUB

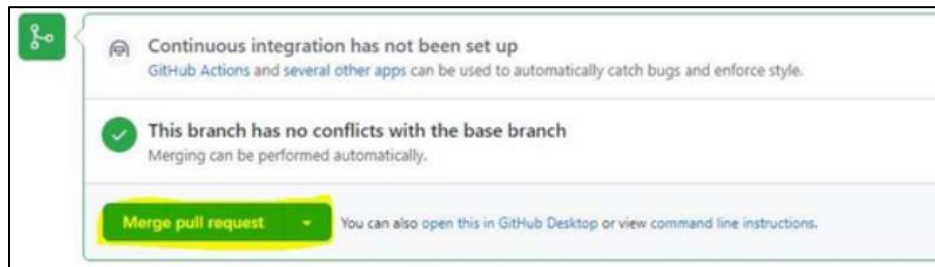
- a. Go to the GitHub repository <https://github.com/mel348/JellyPages>
- b. Go to "Pull Requests" (It will show as shown below)



- c. You will see the "Open Pull Request" window. Click on "Create Pull Request"



- d. Select "Merge Pull Request" and "confirm"



- e. Go back to Git Bash to pull the Master Branch.
1. "git checkout Master"
  2. "git pull"
- f. You should now be up-to-date on the project files.

## LAST REVISION DATE/VERSION

10/17/2023 – Version 1.0 (A release to the public has not yet been issued.)