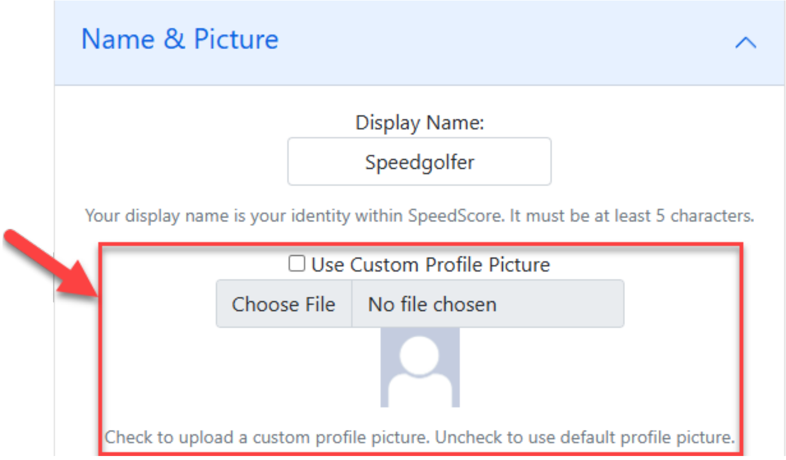
# Implement Feature: Add “Profile Picture” Selector

(Please take **up to 4 minutes** to read the following instructions)

## Introduction

The following pages describe a series of three programming tasks that work toward the implementation of a new “Profile Picture” selector in the “Account & Profile” dialog box used to update a user account:



Display Name: Speedgolfer

Your display name is your identity within SpeedScore. It must be at least 5 characters.

☐ Use Custom Profile Picture

Check to upload a custom profile picture. Uncheck to use default profile picture.

When the “Use Custom Profile Picture” checkbox is checked, users can upload a custom profile picture. When the checkbox is unchecked, the default profile picture is used.

To get started with this task, first check out the `addPicture` branch like so:

```
git checkout addPicture
```

### Relevant Parts of the Code Base

In this task, you’ll be working with three code files:

- `index.html`— This file contains the HTML code for the application. The form shown above is enclosed in a `<div>` element whose `id` is `profileSettingsDialog`.
- `editProfile.js`—This file contains the JavaScript code that defines the event handlers for the form shown above. For example, the submit handler for the form shown above is called `GlobalEditProfileForm.addEventListener`.
- `main.js`—This file includes declarations of global variables (prefaced by `Global`) for several key HTML elements defined in `index.html`. Functions and event handlers in `editProfile.js` frequently use these global variables to refer to HTML elements in the “Account & Profile” form. Global variables for the form shown above are located under a comment block labeled “ACCOUNT & SETTINGS DIALOG FORM.”

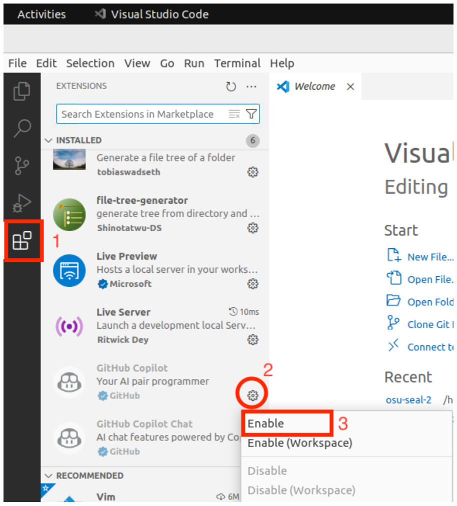
## Data Storage

SpeedScore does *not* interact with a back-end database. SpeedScore's user data are instead stored in the global variable `GlobalUserData` (declared in `main.js`) and in `localStorage`.

## Prepare to Get Started

Before starting these tasks, please note the following:

- You will complete the programming tasks **with the use of CoPilot**.
- You are free to **use the web to search for help or solutions**.
- Read all task instructions aloud *completely* before starting the task.
- Keep your focus on meeting the requirements of the current task. Do not work ahead.
- To enable Copilot, follow these steps, which are shown in the annotated screenshot below:
    - Click on to "Extensions" icon on side panel,
    - Click on the "Settings" icon for GitHub Copilot to open the pop-up menu.
    - Select "Enable" from the pop-up menu.
    - Restart VS Code and verify that Copilot is enabled.

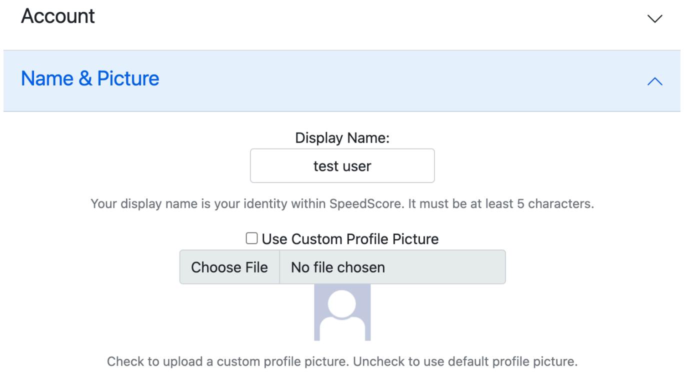


**Do you have any questions before you begin?**

# Task 1: Add Profile Picture Selector to Form

**(Please take up to 2 minutes to read these task instructions aloud. Let us know when you are ready to begin the task.)**

*Add a profile picture selector and label to the “Name & Picture” panel of “Account & Profile” form, as shown in this screenshot:*



Account

Name & Picture

Display Name:

test user

Your display name is your identity within SpeedScore. It must be at least 5 characters.

☐ Use Custom Profile Picture

Choose File No file chosen

Check to upload a custom profile picture. Uncheck to use default profile picture.

### Solution Requirements

1. The profile picture selector must include the following elements, as shown in the screenshot:
   a. A checkbox labeled “Use Custom Profile Picture”
   b. A file picker
   c. A 70-pixel high preview of the default profile picture image, which can be found in the project file `images/DefaultProfilePic.jpg`
   d. A label “Check to upload a custom profile picture. Uncheck to use default profile picture.”
2. The elements listed in the previous requirement must be centered.
3. The “Use Custom Profile Picture” checkbox is unchecked.
4. The file picker is *disabled* (i.e., it cannot be interacted with).

### Testing Your Solution

You may determine the correctness of your solution at any point by running a test suite like so:

```
npm run test task1
```

To practice this, run the test suite **right now**.

### Completing the Task

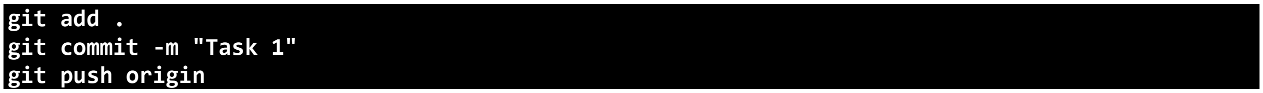
You are done with the task when all tests in the test suite pass.

**Please let us know when you are ready to begin the task.**

**Please await further instruction before proceeding to the next page.**

### Committing and Pushing Your Solution

Commit and push your code solution for Task 1 like so:



```
git add .
git commit -m "Task 1"
git push origin
```

**Please await further instruction before proceeding to the next page.**

## Task 2: Update Profile Picture Selector based on Checked Status

**(Please take up to 2 minutes to read these task instructions aloud. Let us know when you are ready to begin the task.)**

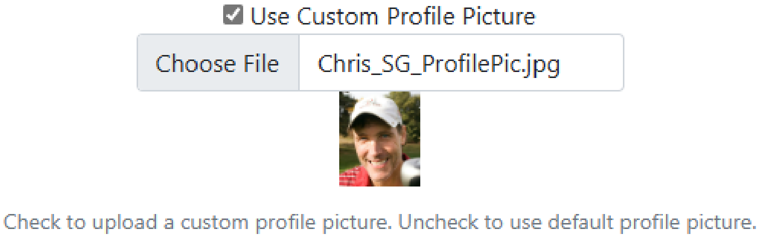
*When the user checks the "Use Custom Profile Picture" checkbox, enable the file picker and allow the user to choose a custom profile image. When the user unchecks the checkbox, clear the file picker, replace the custom profile image with the default image, and disable the file picker.*

### Solution Requirements

1. When the user checks on the "Use Custom Profile Picture" checkbox, the file picker is enabled.

2. When the user uses the file picker to select a custom profile image file from their computer,
   a. The name of the selected file appears in the file picker.
   b. A preview of the selected image appears below the file picker, with a height of 70 pixels and an auto width.

   This screenshot shows an example of the correct appearance:



☑ Use Custom Profile Picture

Chris_SG_ProfilePic.jpg

Check to upload a custom profile picture. Uncheck to use default profile picture.

   3. When the user unchecks the "Use Custom Profile Picture" checkbox,
      a. The preview of the default profile picture appears (with a height of 70 pixels and an auto width)
      b. The file name in the file picker is cleared and the file picker is disabled.

### Testing Your Solution

You may determine the correctness of your solution at any time by running a test suite like so:

```
npm run test task2
```

To practice this, run the test suite **right now**.

### Completing the Task

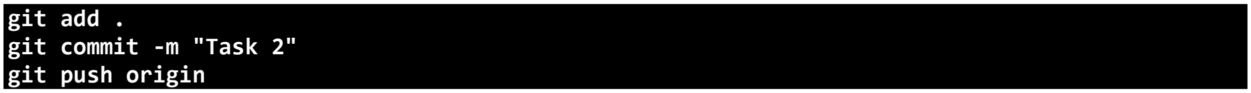
You are done with the task when all tests in the test suite pass.

**Please let us know when you are ready to begin the task.**

**Please await further instruction before proceeding to the next page.**

### Committing and Pushing Your Solution

Commit and push your code solution for Task 2 like so:



```
git add .
git commit -m "Task 2"
git push origin
```

**Please await further instruction before proceeding to the next page.**

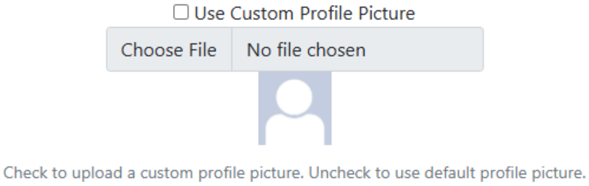
# Task 3: Save Profile Picture in Local Storage

**(Please take 1 minute to this task and let us know when you are ready)**

*When the user clicks the "Update" button for the "Account & Profile" form, the selected profile picture should be saved to local storage.*

### Solution Requirements

1. When the user clicks the "Update" button, save the user's profile picture (whether it be the default profile picture or a custom profile picture) to the `identityInfo.profilePic` property of the user object in Local Storage.
2. When the user opens the "Account & Profile" form, the profile picture selector should correctly display the previously saved profile picture.
   a. If the profile picture is the default profile picture, the "Use Default Profile Picture" radio should be selected and a preview of the default profile picture should be displayed, i.e.:

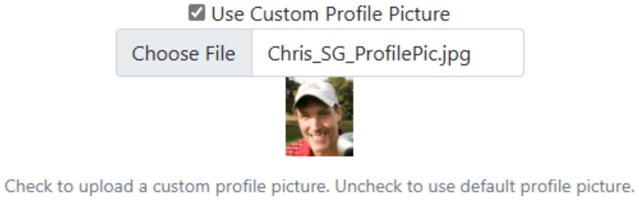


☐ Use Custom Profile Picture

No file chosen

Check to upload a custom profile picture. Uncheck to use default profile picture.

   b. If the profile picture is a custom profile picture, the "Use Custom Profile Picture" checkbox should be selected and a preview of the custom profile picture should be displayed, e.g.:



☑ Use Custom Profile Picture

Chris_SG_ProfilePic.jpg

Check to upload a custom profile picture. Uncheck to use default profile picture.

### Testing Your Solution

You may determine the correctness of your solution at any point by running a test suite like so:

```
npm run test task3
```

To practice this, run the test suite **right now**.

### Completing the Task

You are done with the task when all tests in the test suite pass.

**Please let us know when you are ready to begin the task.**

**Please await further instruction before proceeding to the next page.**

### Committing and Pushing Your Solution

Once all tests pass, commit and push your code like so:

```
git add .
git commit -m "Task 3"
git push origin
```

**Congratulations! You are done with all tasks for this feature!**