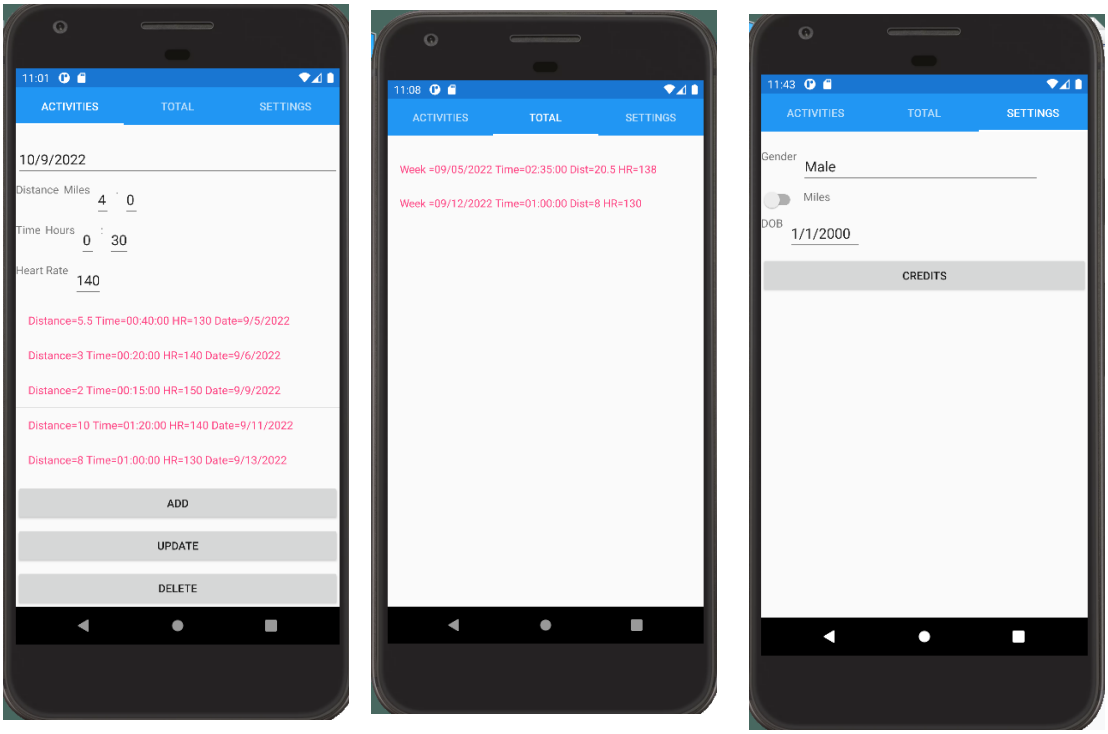
CSE 382
Project #4 - Running Log App
Fall 2022

Create a project named `RunningApp` in your Git repository. This app is meant to be used by runners. It will be a multi-tab application with tabs that will contain the information shown in the mock-up (which is not a polished, final product) shown below.

Activities Totals Settings



A Description of the tabs:

- **Activities** - This page allows the user to enter/update/delete an activity that they have performed. The user is able to enter the following information:
    - Date of run
    - Distance of run
    - Time taken for the run
    - Average heart rate during the run

  The user can perform one of the following actions:
    - Add a new run to the database
    - Update an existing run in the database (the selected one)
    - Delete an existing run (the selected one)
- **Totals** - Provide a list of total weekly mileages, total time run, and average heart rate over the time run. Consider two runs:
    - 1:00 and heart rate = 150

    - 2:00 and heart rate = 130

  Yields an average heart rate of 136.67.
  Use Monday as the starting day of the week.
- **Settings** - This will allow the user to configure information about themselves: gender and date of birth. In addition, the user can select their preferred units - miles or kilometers. Whichever is selected, the GUI must display and accept values in the desired unit. The user configuration must have persistence across invocations of the app. You should use Preferences to accomplish this.

  The credits button must open up a browser (e.g., Safari) to the app's website (use https://www.miamioh.edu). Do not use a WebView, instead have the device open up the webpage in the default browser. Do some research to learn how to do this.

Notes.

- UWP developers should assume that the window will be shaped in the smallest size that appears like a phone held in portrait mode.
- Your app should perform a reasonable action in error conditions; your program must not crash.
- An SQLite database should be used to store the running activities.
- Preferences should be used to store user preferences.
- You can demo your app using Android, iOS, or UWP. However, your approach must be runnable with the other platforms.
- Follow the process for submission used for the previous project.

Script for Video.

Like the previous project, you must submit a video demonstrating your project. Try to make the video at most 4 minutes long. Here is the script:

- Introduction
    - Announce yourself
    - State what works and what does not
    - Do a quick walk through of your 3 main XAML pages
    - Do a quick walk through of your 3 C# code behind.
- Set user configuration to:
    - Female, miles, DOB=3/1/1969
    - Close app, reopen, click on config tab to ensure show the values are persistent
- Click the credits button to show that a website is brought up in a browser. Dismiss the browser and go back to the app.
- Click on Activities tab and add the following activities (which are in miles):
    - 10/24/2022 4.0 0:35 130
    - 10/26/2022 6.0 0:45 130
    - 10/28/2022 6.0 0:46 130
    - 10/31/2022 10.0 1:20 200
    - 11/1/2022 3.1 0:35 130
- Click on the Totals tab and read off the weekly totals

- Click on the Activities tab, delete the run on 10/24/2022, click on Totals tab and show the weekly totals are correct.
- Click on the Activities tab, update the run on 10/26/2022 to be 16 miles and 3:00, click on the totals tab. Point out if the totals are correct or incorrect.
- Close app. Restart. Show the individual runs are still present and the totals are correct.
- Click on the Settings tab, select metric, click on Totals tab. Show the values are displayed in kilometers.
- Click on the Activities tab. Add a 25:00 5k run on 10/17/2022. Click on the Totals tab.
- Click on the Settings tab. Select miles. Click on the Totals tab.
- Show the app behaves under the following error conditions:
    - That negative and zero distances are not possible to enter (or handled via code)
    - That negative and zero times are not possible to enter (or handled via code)
    - Only legal dates are possible to enter (or handled via code)

Scoring.

- **(80) Operation/Correctness**. Your application must operate correctly, not crash, use a database to store the running activities, and store user configuration using Preferences. Here are the approximate point breakdowns.
    - **(50)** Using a database to store running activities and correctly using them within the application. Compute the weekly totals correctly.
        - **(20)** Add
        - **(10)** Modify
        - **(10)** Delete
        - **(10)** Totals
    - **(15)** Using preferences to store user data and correctly using them within the application.
    - **(10)** The user's unit preference is correctly dealt with. Note, the user can change this at any time and all displays/labels will reflect the current usage.
    - **(5)** Browser is opened up to show credits.
    - The GUI must be functional. If not, deductions will be taken.
- **(5) Robustness.** Your program should not crash and should prevent the user from entering illegal values.
- **(15) Appearance and functionality**.
    - **(6) Overall appearance and functionality**. Your application will be subjectively evaluated for visual and operational appeal. Doing well on this criteria will require good use of colors, font sizes, contrasting colors, and effective organization/display of information.
    - **(9) ListViews.** Two of the tabs include a ListView and you should ensure that the displayed values have a good size, color, contrast, and format. The ones shown above would be considered poor. For example: it is not clear what information is being shown, the different pieces of information flow into one another, the primary key is shown but is irrelevant to the user.
- Appropriate application of techniques. It is expected that students use good software development practices and also apply learned techniques to mobile apps when appropriate.