Code rundown:

My repository name: Sri_Vaishnavi_Ghantasala. Same as last semester. This is where I will upload all of my code, and resources, including all the notes I will take.

Current repository name: aac-ios
Old repositories: TextToSpeech-Ft, PhotoUpload_Ft

Disclaimer: This is **my** understanding of the code base and the code functionality
All of the code is in Swift so far.

The TextToSpeech-Ft repository: The main goal of this repository is to have all the text to speech related stuff. Text to speech is basically a feature where you type in words and the application reads it outloud. Apple has very similar features available already. (Link: https://github.gatech.edu/VIP-BTAP/TextToSpeech-Ft)

TextToSpeech-Ft/TextToSpeech:

- *TextToSpeech*
    - Preview Content/Preview Assets.xcassets: A preview of what the input is supposed to look like in the form of a json file is available under this path: **TextToSpeech-Ft/TextToSpeech/TextToSpeech/Preview Content/Preview Assets.xcassets/Contents.json**
    - ContentView.swift: This is an outline with a VStack (which basically places the objects in rows -> vertically aligning them). The two objects here are the word "Text" and a button that has "Speak" written on that. Clicking this button reads out loud the input. This is under the path: **TextToSpeech-Ft/TextToSpeech/TextToSpeech/ContentView.swift**
    - TextToSpeechApp.swift: This is merely an area where you call on the ContentView. This is under the path: **TextToSpeech-Ft/TextToSpeech/TextToSpeech/TextToSpeechApp.swift**
    - Assets.xcassets:
        - AccentColor.colorset: Contents.json: A json file similar to Preview Content/Preview Assets.xcassets
        - AppIcon.appiconset: Contents.json: A json file similar to Preview Content/Preview Assets.xcassets
        - Contents.json: Same exact contents as the file Preview Content/Preview Assets.xcassets

- *TextToSpeech.xcodeproj:*
    - project.xcworkspace:
        - xcshareddata: IDEWorkspaceChecks.plist: .plist files usually contain information on the configuration of iOS mobile applications. This file

includes version information and a link linking to the different data types in use. The main function of this page is that the App Store uses this information when deploying the app to the app store. Moe information can be learned at: https://docs.flexera.com/adminstudio2019/Content/helplibrary/ISCSTaskImportiOSplist.htm#:~:text=plist%20file%20contains%20critical%20information,the%20mobile%20app%20is%20compiled.
Path to this file: **TextToSpeech-Ft/TextToSpeech/TextToSpeech.xcodeproj/project.xcworkspace/xcshareddata/IDEWorkspaceChecks.plist**

- xcuserdata/sydneydefelice.xcuserdatad: UserInterfaceState.xcuserstate: This is a file with raw data. No code involved. Path to this file: TextToSpeech-Ft/TextToSpeech/TextToSpeech.xcodeproj/project.xcworkspace/xcuserdata/sydneydefelice.xcuserdatad/

- xcuserdata/sydneydefelice.xcuserdatad/xcschemes: xcschememanagement.plist. Another .plist file with information similar to the information in xcshareddata: IDEWorkspaceChecks.plist. Path to this file: **TextToSpeech-Ft/TextToSpeech/TextToSpeech.xcodeproj/xcuserdata/sydneydefelice.xcuserdatad/xcschemes/xcschememanagement.plist**
- project.pbxproj: This is some developer documentation that is all commented out. It is also pretty hard to decipher the actual functionality of this file. Path to this file: **TextToSpeech-Ft/TextToSpeech/TextToSpeech.xcodeproj/project.pbxproj**

TextToSpeech-Ft/README.md: The README is empty. Adding some more information would be helpful. I will discuss with Sydney the possibility of adding this information in the readme so that it is helpful for any new members to understand what is happening in this repository.

PhotoUpload_Ft repository: This repository is mostly for the photo upload stuff our team will work on. Link: https://github.gatech.edu/VIP-BTAP/PhotoUpload_Ft

- AAC:
  - AAC.xcodeproj:
    - project.xcworkspace: IDEWorkspaceChecks.plist: .plist files usually contain information on the configuration of iOS mobile applications. This file includes version information and a link linking to the different data types in use. The main function of this page is that the App Store uses this information when deploying the app to the app store. Moe information can be learned at: https://docs.flexera.com/adminstudio2019/Content/helplibrary/ISCSTaskImportiOSplist.htm#:~:text=plist%20file%20contains%20critical%20information,the%20mobile%20app%20is%20compiled.

Path to this file: **PhotoUpload_Ft/AAC/AAC.xcodeproj/project.xcworkspace/xcshareddata/IDEWorkspaceChecks.plist**

- xcuserdata:
  - Name.xcuserdatad:
    - xschemes: xcschememanagement.plist: Another .plist file with configuration info for the app store to be able to use. Path to file: PhotoUpload_Ft/AAC/AAC.xcodeproj/xcuserdata/Name.xcuserdatad/xcschemes/
    - harshithakotlure.xcuserdatad/xcschemes: xcschememanagement.plist: Another .plist file with configuration info for the app store to be able to use. Path to file: **PhotoUpload_Ft/AAC/AAC.xcodeproj/xcuserdata/harshithakotlure.xcuserdatad/xcschemes/xcschememanagement.plist**
- project.pbxproj: This is some developer documentation that is all commented out. It is also pretty hard to decipher the actual functionality of this file. Path to this file: **PhotoUpload_Ft/AAC/AAC.xcodeproj/project.pbxproj**

- AAC:
  - Assets.xcassets:
    - AccentColor.colorset: Contents.json: Very similar to Contents.json below. A preview of what the input is supposed to look like in the form of a json file is available under this path: **PhotoUpload_Ft/AAC/AAC/Assets.xcassets/AccentColor.colorset/Contents.json**
    - AppIcon.appiconet: Contents.json: Icon information such as size and whether it's on an iphone, ipad, or ios-marketing, as well as scale listed in a json file. Path for this file: **PhotoUpload_Ft/AAC/AAC/Assets.xcassets/AppIcon.appiconset/Contents.json**
    - Contents.json: Contents.json: A preview of what the input is supposed to look like in the form of a json file is available under this path: **PhotoUpload_Ft/AAC/AAC/Assets.xcassets/Contents.json**
  - Preview Content/Preview Assets.xcassets: Contents.json: Contents.json: A preview of what the input is supposed to look like in the form of a json file is available under this path: **PhotoUpload_Ft/AAC/AAC/Preview Content/Preview Assets.xcassets/Contents.json**

        - AACApp.swift: Calls on ContentView (listed view). This is where you have centralized running capability (to run the whole app). If we add functionality in future, we would consolidate all the features and run it from here. Path to this file: **PhotoUpload_Ft/AAC/AAC/AACApp.swift**
        - ContentView.swift: This is the file with the most content in terms of what appears on the page. The main things that are done here is loading the image and processing it. There are two functions that do this and the contentview function that calls on these. Path for this file: **PhotoUpload_Ft/AAC/AAC/ContentView.swift**
        - ImagePicker.swift: This is where the UI appearance is put together. This is where the app decides on whether or not the upload is an image and has error cases for if the input is invalid (not an image, no election, etc.). Path for this file: **PhotoUpload_Ft/AAC/AAC/ImagePicker.swift**
        - ImageSaver.swift: This is where the edits are written and saved. Path to file: **PhotoUpload_Ft/AAC/AAC/ImageSaver.swift**

    - AACTests: This is where the skeleton for the test cases is. There are detailed comments on what each test has to do, but no test cases written out yet. TBD. Path of the file: **PhotoUpload_Ft/AAC/AACTests/AACTests.swift**
    - AACUITests: This is where the skeleton and basic implementation for the UI test cases is. There are detailed comments on what each test has to do and most of the implementation too. Path of the file: **PhotoUpload_Ft/AAC/AACUITests/AACUITestsLaunchTests.swift**
    - .DS_Store: Raw data file
- .DS_Store: This is a file with raw data. No code involved. Path to this file: **PhotoUpload_Ft/.DS_Store**
- README.md: Does not include any information. I will discuss with Sydney the possibility of adding this information in the readme so that it is helpful for any new members to understand what is happening in this repository.

Reflection on learnings about codebase:
The codebase is pretty extensive and will be a great starting point to build off of this semester. I believe we will be able to start using the photo and audio features we have the basic code for here. I think this doc can be very helpful for s documentation purposes for future reference and will bring this up to Sydney to ask if we can update the Readme files to include this information as the documentation.