**Avcol Forms**

Technical Details and Write Up

**Avcol Forms App**

[GitHub Page](https://github.com/AvcolDJPownall/AvcolForms)

**Introduction**

My desktop application intends to provide an easier method to fill out school-related paperwork. Both students and teachers will be able to access various forms using the school computers.

**Client**

Avondale College primarily uses paper forms for both students and teachers. This presents an inconvenience to many users, as finding and printing the appropriate paperwork can require a large amount of time and work. The solution I am presenting gathers many common forms and displays them in one digital repository. Since Windows 10 desktop computer are exclusively utilized across the school, this creates an interesting opportunity for a Windows Forms application.

**Specifications**

- Windows Forms application with GUI  
- Utilizes .NET Framework 4.7.2

- Automatically draft emails containing the form’s results

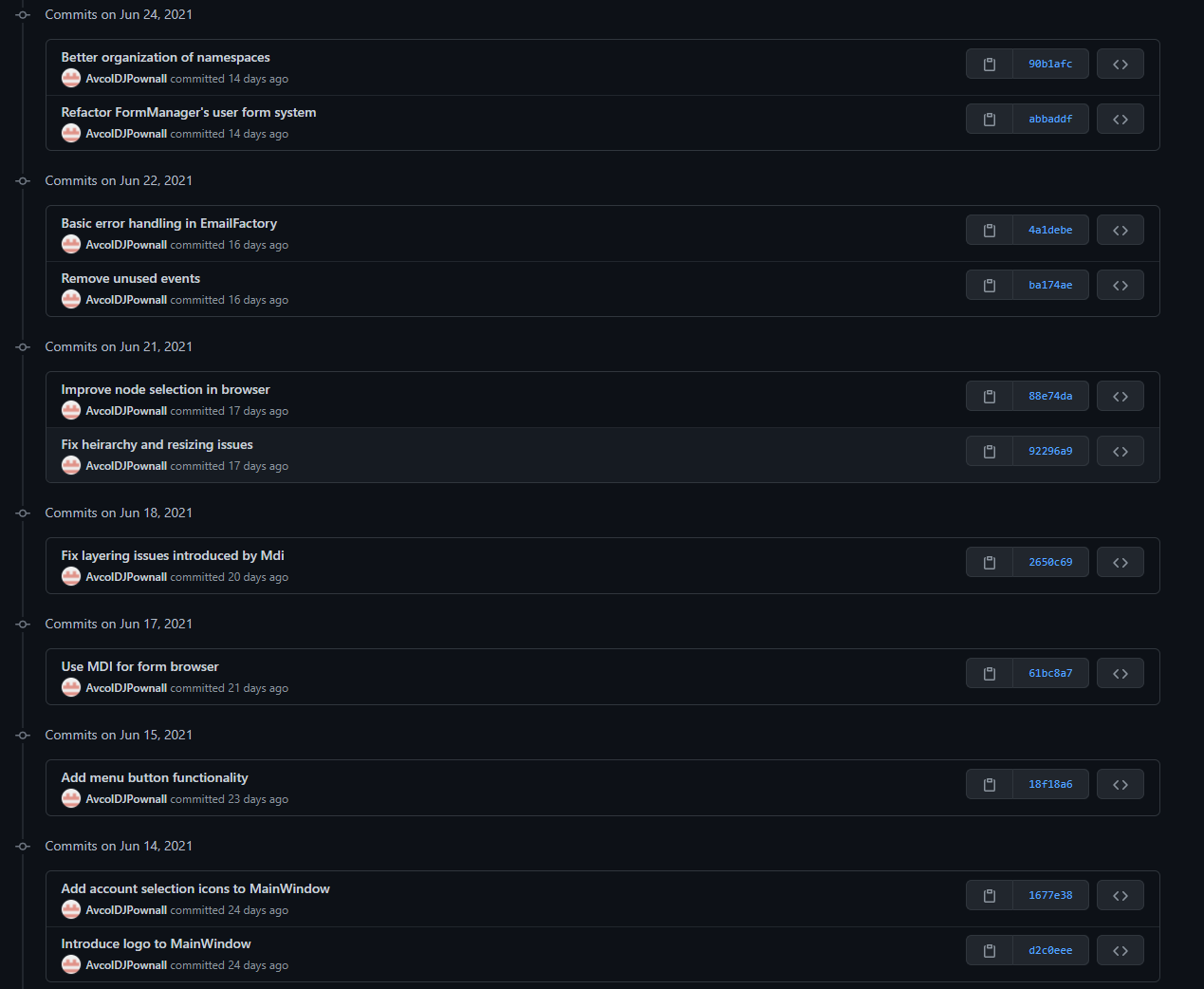
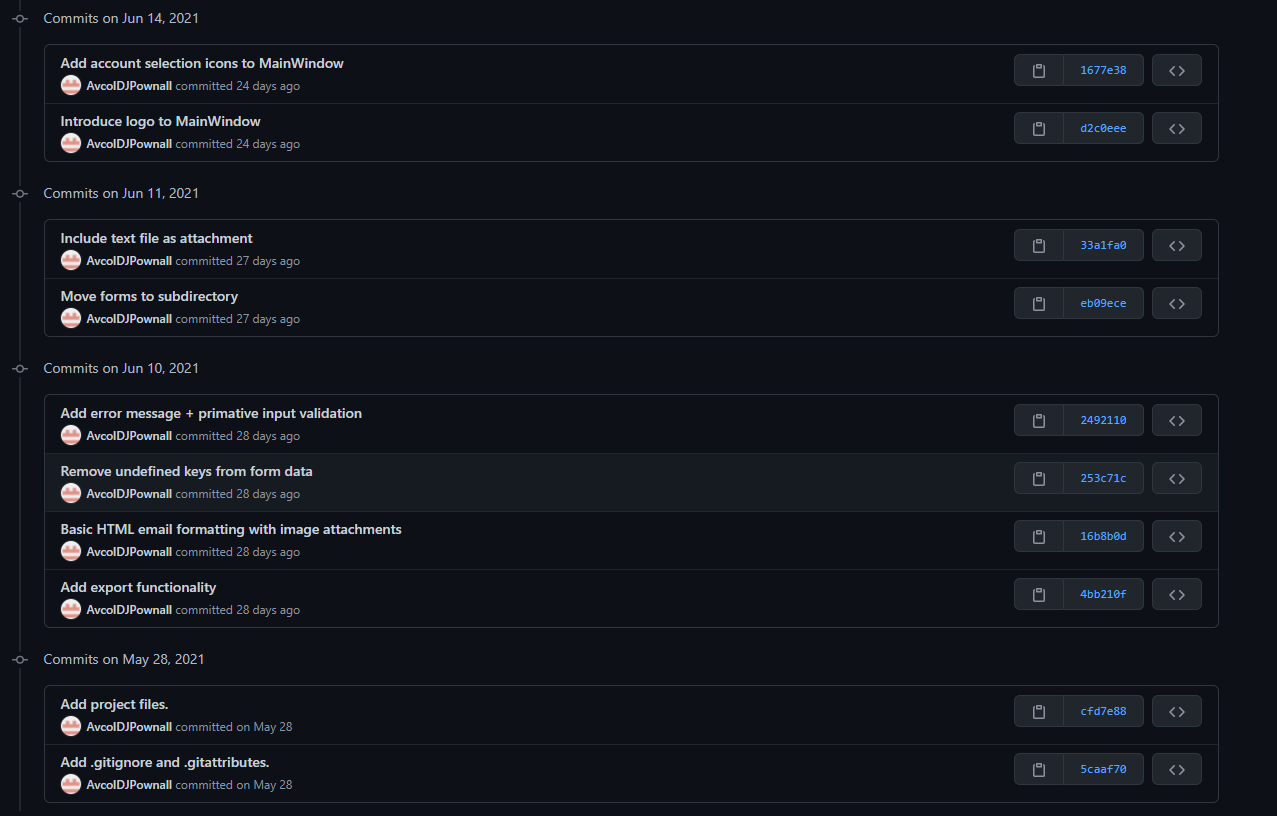
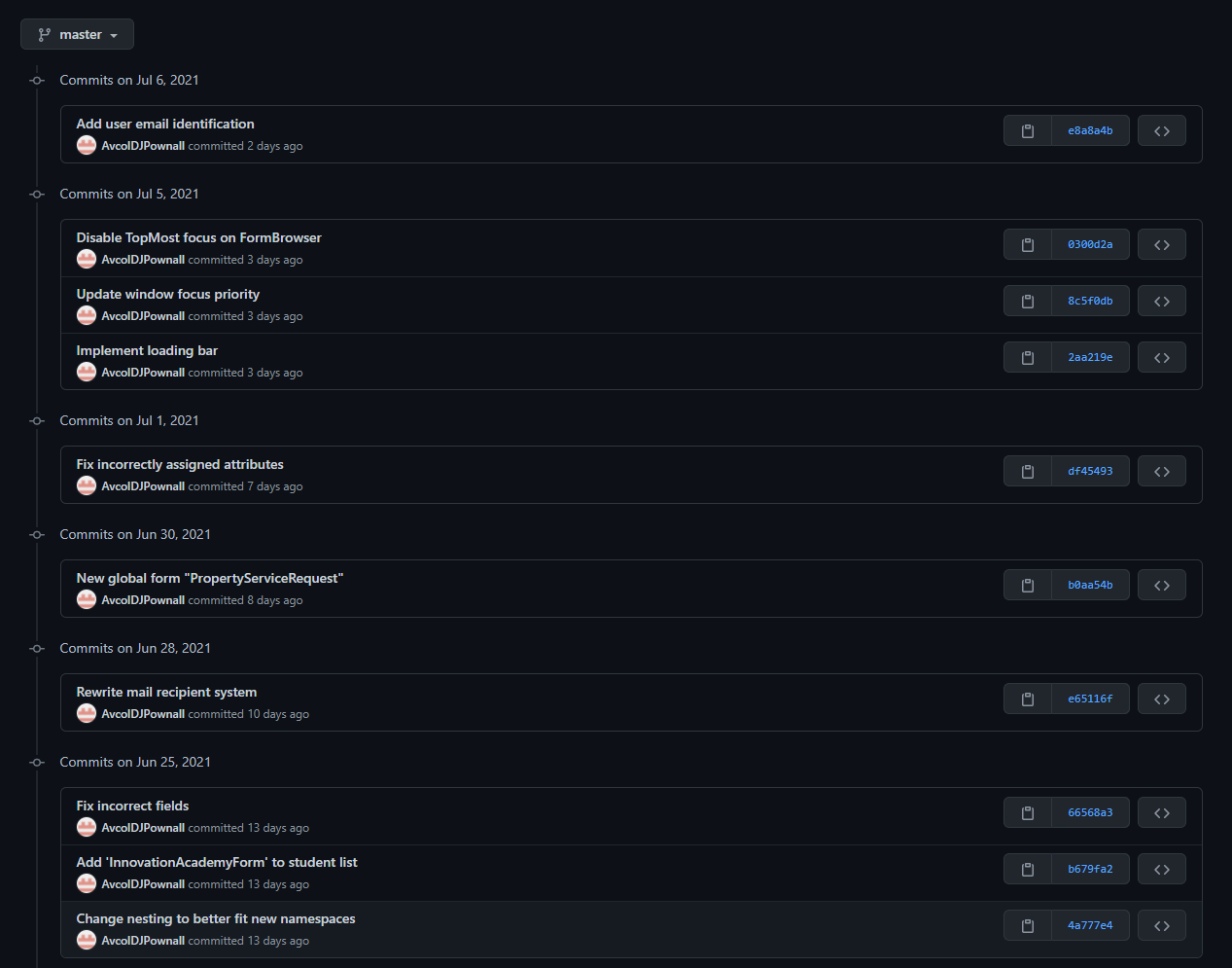
**Development**

Working with Windows Forms, there were a few limitations I ran into.

For instance, many attributes cannot be modified without halting the main thread. This posed issues when working with events, as some elements needed to be modified when the form changed states. To remedy this, I extensively use the method Task.Run() to run basic operations on a new thread, thus allowing actions such as submitting the form to feature multi-threading.

Another issue I encountered was authorization inside the SMTP client. Storing the mail client’s password inside the program would result in security issues, especially considering how easy it is to decompile .NET Framework binaries using third-party tools. In an official implementation, emails should be sent from a server-side program, perhaps from a TCP connection or POST request. This would drastically improve security, as there would be no way to reverse-engineer the program to find the password string. While the program does not feature these measures, I’ve taken steps to ensure the password is not stored in plain-text. If the school’s IT department decide to use this program, they would need to implement a server-side solution, rather than obfuscating the client.

**Progress**

All version control and historic commits can be viewed on GitHub via [this link](https://github.com/AvcolDJPownall/AvcolForms/commits/master).

**Maintenance**

This program will need to be maintained following its initial implementation, due to various factors.

New form entries currently must be made using Visual Studio’s designer tool. As such, implementing more forms would require recompiling the code, then pushing the updated binaries across the school. Improvements could be made in future to make adding new entries easier, such as an update system or automatically generating elements by fetching a json file from a webserver.

The program would also benefit from a more robust email system, where a server-side application could handle form submissions. A future update could move authentication from the client application to an external server, greatly improving security. This would practically be a requirement for continued use in Avondale College.