CODING & PROGRAMMING

FEYNMAN LIBRARY MANAGER

VARUN SHENOY

CALIFORNIA
BAY SECTION
CUPERTINO HIGH SCHOOL

CONTENTS OF THIS FILE

FILE STRUCTURE

- Bay_Cupertino
  - README.pdf
  - bin
  - DOCUMENTATION.pdf
    - The documentation file elaborates on this README file with more information.
  - src
    - This folder contains all source code files for the application, including binaries for npm frameworks and semantic UI.
    - The key file that I wrote code in is 'windows.js'. It has comments explaining the logic and structure behind the code.
    - 'mainWindow.html' is the HTML file that contains the user interface structure of the application.
  - screenshots

- This folder has various screenshots/images from the application.

SOFTWARE USED

The following tools were used during this project:

Atom Text Editor (atom.io)
Git & Github (github.com)
Languages:
    HTML
    CSS
    Javascript ES6
npm modules:
    Electron Web Development Framework (electronjs.org)
    jQuery (jquery.com)
    node-json-db (github.com/Belphemur/node-json-db)
    moment.js (momentjs.com)
    SweetAlert2 (sweetalert2.github.io)
    jsPDF and jsPDF-Autotable (github.com/MrRio/jsPDF, github.com/simonbengtsson/jsPDF-AutoTable)
Design Frameworks:
    Semantic UI (semantic-ui.com)
    Sketch (sketchapp.com)

SOURCE OF INFORMATION

* Stack Overflow (stackoverflow.com)
* Electron Documentation (electronjs.org/docs)
* Semantic UI Documentation (semantic-ui.com/introduction/getting-started.html)
* YouTube Workshops on Electron
    Traversy Media:
      https://www.youtube.com/watch?v=kN1Czs0m1SU
      https://www.youtube.com/watch?v=mr9Mtm_TRpw
    Github Universe Workshop:
      https://www.youtube.com/watch?v=FNHBfN8c32U

COPYRIGHT NOTATIONS

The following Node Package Manager (npm) modules and design frameworks were used in the application.

npm modules:

Electron Web Development Framework (electronjs.org)

jQuery (jquery.com)

node-json-db (github.com/Belphemur/node-json-db)

moment.js (momentjs.com)

SweetAlert2 (sweetalert2.github.io)

jsPDF and jsPDF-Autotable (github.com/MrRio/jsPDF, github.com/simonbengtsson/jsPDF-AutoTable)

Design Frameworks:

Semantic UI (semantic-ui.com)

Icons were sourced from the Semantic UI library.

INSTRUCTIONS FOR RUNNING PROJECT

*Note: you may need to disable anti-virus/anti-malware software (i.e. Norton, McAfee) to run the application. Unfortunately, I don't have credentials (or resources) to sign the application.

Windows:

1. Navigate to the "Windows" folder in the "bin" folder
2. Extract files from the .zip file if needed.
2. Double-click the "Feynman Library Manager.exe" file to run the program

*Note: there are many files in this folder and they are necessary for the application to run. Do not remove the "Feynman Library Manager.exe" file or any other files from this folder.

Mac:

1. Navigate to the "Mac" folder in the "bin" folder
2. Double-click the "Feynman Library Manager" program to run it

Source Code:

All source code is in the "src" folder.

The main code lies in the 'window.js' file.

The user interface for the app is defined in the 'mainWindow.html' file.

All Javascript code that I wrote has been documented using comments within the file.

TEMPLATES USED

No templates were used in building the Feynman Library Manager app, just open-source frameworks to create a immersive and intuitive user interface.

FEATURES

Minimum Features for FBLA Coding & Programming:

- Track student and teacher names with ability to enter/view/edit names
- Track the issuance of books for a student or teacher
- Manage different limits for the number of books that can be issued to a student or teacher
  - Teachers can hold up to 15 books for 42 days each
  - Students can hold up to 5 books for 21 days each
- Unique generated IDs for every book
- Track student and teacher names with ability to enter/view/edit names
- Generate/Print overdue reports for individual patrons
- Generate/Print weekly reports to show books issued to whom and number of days leading to the due date return

Custom Features:

- Simple and clean user interface with vibrant colors
- Support for both Windows and Mac operating systems
- Designed with the best user experience for a librarian in mind
- Secure local datastore with daily dynamic backups
- Load backups effortlessly from previous days
  - Test this out by adding new patrons and books and then revert to a previous database backup
- Filter books/patrons via the interactive search bar
- Add new patrons and books through a simple form interface
- Interactive Help Menu