Technical Manual

1. Project Implementation
   1. The application accepts standard Lindenwood University course catalogs in CSV format
   2. The course information on the main page of the application is dynamically fetched from your MySQL database (this means your MySQL database must be populated with at least one course catalog for the main page to run as intended)
   3. The application will only generate schedules matching the selected filters that contain all the selected courses
2. Project algorithm
   1. The algorithm first creates an array each section of the selected courses and stores their information
   2. Then, any course in the generated array that interferes with the selected personal schedule is removed
   3. Next, the generated array is converted to a non-binary tree where each entry of the array is turned into a node. A node is added to the tree if and only if all its ancestors (i.e., previous courses) are compatible (i.e., don’t have conflicting schedules)
   4. Finally, the non-binary tree is searched to find each path to a leaf node at the maximum possible depth for the non-binary tree (this is not necessarily the actual maximum depth of the non-binary tree)
   5. Any path found this way is displayed in a table as a schedule at the bottom of the application’s main web page
3. Known Issues
   1. Minor stylistic bugs with certain devices and browsers (mostly desktop devices running Google Chrome as their browser)
   2. When installing the application, administrator privileges are often required and privileges for certain folders may need to be altered
   3. Online classes are considered when selecting courses, but (since they never interfere with an in-class session) are not considered when generating a final schedule

A screenshot of a cell phone

Description automatically generated

A close up of a piece of paper

Description automatically generated

