



CS261-Data Structure and Algorithms Mid Project Proposal(Fall 2022)



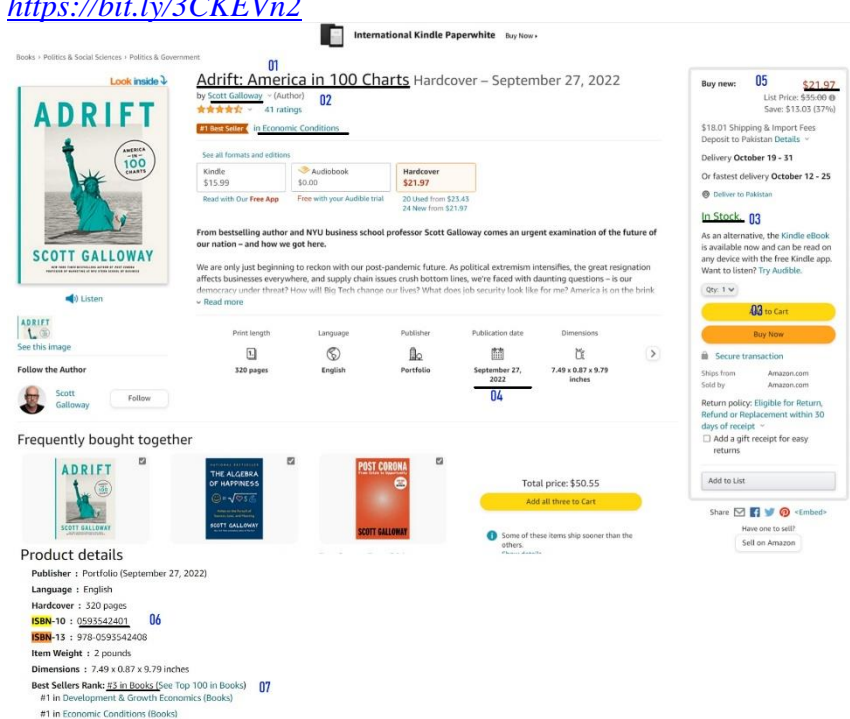
Proposer Details

Group Number	Group 03
Registration Number of Group Members	2021-CS-02 2021-CS-11 2021-CS-35

Proposal Details

<i>Project</i>	
Proposed Project Title	LIBRARY MANAGEMENT SYSTEM
Executive Summary	<p>Libraries are important meeting places in communities and are critical for providing public access to knowledge and information. Our solutions help libraries serve their communities and run more efficiently.</p> <p>From small school libraries to country libraries, pop-up libraries to big city libraries and special libraries, we help our customers serve their patrons, giving librarians more time to pursue their important work of promoting reading, learning and creativity, supporting researchers and information seekers, and creating a welcoming space for library users.</p> <p>Library Management System eliminates the hassle of keeping track of all the books and manually entering the data which makes it easier to organize the books and to ultimately run your business in an efficient manner. Using the software results in lower costs which is good for business. All the data is stored in a database and the users can easily access and categorize the books by utilizing the searching and sorting algorithms</p> <p>The library management system is a software that is designed to manage all the functions of a library. It helps the librarian to maintain the database of new books and the books that are borrowed by members along with their due dates.</p> <p>This system completely automates all your library's activities. The best way to maintain, organize, and handle countless books systematically is to implement a library management system software.</p> <p>A library management system is used to maintain library records. It tracks the records of the number of books in the library, how</p>

	<p>many books are issued, or how many books have been returned or renewed or late fine charges, etc.</p> <p>You can find books in an instant, issue/reissue books quickly, and manage all the data efficiently and orderly using this system. Thus, the purpose of a library management system is to provide instant and accurate data regarding any type of book, thereby saving a lot of time and effort.</p>
Business Case	
Outline the business need for the project	<p>What happens when someone makes a mistake in manually entering the data?</p> <p>Could the time spent in manually entering and indexing the books be utilized in a more efficient manner? Like making plans for the staff or making the library more reader friendly.</p> <p>What happens if there is a fire or a hazardous chemical on the premises of the library? It would have to close for a few days which could result in losing our customers.</p> <p>If any of the aforementioned things do happen, it could prove to be very damaging for business and may ruin the reputation of the library. Thus, implementing a Library Management System would solve 90% of these problems. The data will be secured and easy to handle in case of a wrong entry, this will save time which could then be used to maintain the physical conditions of the library or to go through the business plans.</p> <p>It will offer 24-7 connectivity so that readers can access their books in a soft copy form in case of an On-sight emergency.</p> <p>Thus, even a medium to small business would profit by implementing a library management system.</p>
End user of the product	This system can be utilized by all libraries for example, Academic or National. Students and avid readers will benefit greatly from this.
Motivation for Project	To find out how data flows in a management system and to create something we're proud of which could benefit others.

State the level of impact expected should the project proceed and implications of not proceeding	We will make sure if the implementation does have an impact on the surface level, we had made sure that the user is aware of it and all the necessary information will be provided in the user-manual.																													
Technical Details																														
Name of Entity	Book																													
Attributes of Entity (Minimum seven attributes/rows can be increased)	<table><tr><th>Name</th><th>Data Type</th><th>Description</th></tr><tr><td>Title</td><td>String</td><td>Name of the book</td></tr><tr><td>Author</td><td>String</td><td>Book Publisher</td></tr><tr><td>Price</td><td>Float</td><td>Price of the book</td></tr><tr><td>Publish Year</td><td>String</td><td>Date the book was released on</td></tr><tr><td>ISBN</td><td>Int</td><td>Book serial number</td></tr><tr><td>Ratings</td><td>Float</td><td>Reviews of the books in terms of digits</td></tr><tr><td>In Stock</td><td>Bool</td><td>Availability of the book</td></tr><tr><td>Rating</td><td>Int</td><td>Rating of Book based on reviews.</td></tr></table>			Name	Data Type	Description	Title	String	Name of the book	Author	String	Book Publisher	Price	Float	Price of the book	Publish Year	String	Date the book was released on	ISBN	Int	Book serial number	Ratings	Float	Reviews of the books in terms of digits	In Stock	Bool	Availability of the book	Rating	Int	Rating of Book based on reviews.
Name	Data Type	Description																												
Title	String	Name of the book																												
Author	String	Book Publisher																												
Price	Float	Price of the book																												
Publish Year	String	Date the book was released on																												
ISBN	Int	Book serial number																												
Ratings	Float	Reviews of the books in terms of digits																												
In Stock	Bool	Availability of the book																												
Rating	Int	Rating of Book based on reviews.																												
Sample of Scrapping Source	<div><div>https://bit.ly/3CKEVn2</div><div></div></div>																													

Gitlab Repository Link	https://bit.ly/3CKdWYM
Sorting Algorithms	<i>Insertion Sort, Bubble Sort, Merge Sort, Selection Sort, Hybrid Merge Sort, Combo sort, Shell Sort, Strand Sort, Quick Sort, Counting Sort, Radix Sort, Bucket Sort, Postman Sort, Pigeon-hole Sort.</i>
Algorithm Name	Description(Each algorithm in 2-3 lines)
Insertion Sort	Comparison-based stable sorting algorithm. Starts iteration from the 2 nd element of array to end, and move each element to left where it fits if necessary
Bubble Sort	Comparison-based stable sorting algorithm. It keeps comparing adjacent elements and swapping till no swapping is done in one complete iteration of main loop.
Merge Sort	Comparison-based stable sorting algorithm. Recursively divides array into two parts till sub array cannot be further divided then merges them together in sorted order
Selection Sort	Comparison-based unstable sorting algorithm. It finds the minimum element of array and swaps it with the starting value of loop till it reaches loop end.
Hybrid Merge Sort	Comparison-based stable sorting algorithm. Combines merge sort with insertion in such a way that it switches between them depending on the efficiency of algorithms with accordance to elements.
Comb Sort	Comparison-based unstable sorting algorithm. It's an improved version of bubble sort. It first handles the small values on end of list (turtles) and big values on start of list (rabbits) and then apply bubble sort.
Shell Sort	Comparison-based unstable sorting algorithm. It is similar to insertion. It starts by sorting elements far from each other and reduces the gap of elements to be compared after every main loop iteration.
Strand Sort	Comparison-based stable sorting algorithm. It is recursive algorithm that creates sorted sub-arrays and merges them
Quick Sort	Comparison-based unstable sorting algorithm. It divides array into two using a pivot such that values smaller to pivot on left and larger on right and we get sorted array after the recursive calls.
Counting Sort	Linear stable sorting algorithm. Finds max and min of array, creates an array of size max – min and place each element at index (which is equal to its value) of new array.
Radix Sort	Linear stable sorting algorithm. It creates ten buckets from 0 to 9. Then starts from least significant digit, places element according in bucket and progressively moves to more significant digits
Bucket Sort	Linear stable sorting algorithm. It groups element into 10 buckets Based on the place value and sorts them by other algorithms or by recursive calls
Postman Sort	Linear stable sorting algorithm. It sorts from most significant digit to the least. Length of the largest value is used in process
Pigeonhole Sort	Linear stable sorting algorithm. Same as Counting Sort, but it moves items twice, one on input array and other in bucket array

Searching Algorithms	<i>Linear, Binary, Jump, Fibonacci</i>
Algorithm Name	Description(Each algorithm in 2-3 lines)
Linear Search	Starts from beginning of array to end, searches each index of array . it stops at soon as it finds the data or reach end of array
Binary Search	Divides a sorted array into half . Take the part that has the required value and then keeps on dividing it by comparing till it is left with only one element(Answer)
Jump Search	Takes a sorted array and makes jump equal to square root of array length till it jump to an index having greater value then the required one then performs linear search on that array.
Fibonacci Search	It also takes sorted array but divides into unequal part based on Fibonacci series to narrow down possible locations.
Searching Filters for each data type	<p>A column can be searched by giving at least one input string and using filters:</p> <p>Is equal to : Required data is exactly equal</p> <p>Is not equal : Required data not exactly same as provided string</p> <p>Contains : Required data is sub string of elements</p> <p>Does not contain : Required data is not sub string of elements</p> <p>Starts with : Required data is present at beginning of element</p> <p>Ends with : Required data is at end of element</p> <p>There will be another input string having same filters and results of both inputs can be compared by AND or OR operator to get desired result.</p>
Multi-Level Sorting	Levels are added where array is sorted normally in required order in first level but it can happen that that column has multiple same values so at next level those recurring values are sorted based on another column and so on.
Any other features	<p>A graph between time and number of entries scrapped per unit tie shall be displayed on the Scrapping window.</p> <p>Bonus Task <i>Any URL Scrapping.</i></p>

LIBRARY MANAGMENT SYSTEM

LMS

Group 03

✕ SCRAP-DATA

✕ DASHBOARD

✕ EXIT

SCRAPPING

SCRAP FROM URL

URL for Scrapping

WWW.....COM

Start

SCRAP FROM PRE-CODED WEBSITE

Start

ATTRIBUTES FALIED TO SCRAPP

STATISTICS

308 x 198

Number of Books Scrapped: 10,450

Pause

End

UI Component Name	Type of UI component	Purpose of UI Component/Other details
Frames	Containers	To group widgets of same semantics.
Graph	Image	To display <i>Scrapp-time</i> graph.
Icons	Image	Icons for Menu.
Labels	Display Widgets	To display headings and outputs.
List	Item Widget	To display <i>failed attributes in scrapping</i> .
Progress Bar	Display Widgets	To display progress of scrapping.
Push Buttons	Buttons	To Start, Pause & Play and End Scrapping.
Text Box	Input Widget	To Enter URL for Scrapping(<i>Bonus Task</i>).

LIBRARY MANAGEMENT SYSTEM																																																																																											
<p>LMS</p> <p>Group 03</p> <hr/> <div style="margin-bottom: 10px;"><input type="checkbox"/> SCRAP-DATA</div> <div style="margin-bottom: 10px;"><input checked="" type="checkbox"/> DASHBOARD</div> <div><input type="checkbox"/> EXIT</div>	<h1 style="text-align: center; margin: 0;">BOOK DASHBOARD</h1> <div style="display: flex; justify-content: space-between; margin-top: 20px;"> <div style="width: 45%;"> <div style="background-color: #333; color: white; padding: 5px; text-align: center; margin-bottom: 10px;">MULTI LEVEL SORTING</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">SELECT COLUMN ▼</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">SELECT ALGORITHM ▼</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">SELECT ORDER ▼</div> <div style="text-align: center; background-color: #eee; padding: 5px; margin-bottom: 10px;">ADD LEVEL</div> <div style="border: 1px solid #ccc; padding: 5px;"> LIST OF ALL LEVELS <div style="border-bottom: 1px solid black; height: 15px; width: 100%;"></div> <div style="border-bottom: 1px solid black; height: 15px; width: 100%;"></div> <div style="border-bottom: 1px solid black; height: 15px; width: 100%;"></div> <div style="border-bottom: 1px solid black; height: 15px; width: 100%;"></div> </div> <div style="margin-top: 20px;"> TIME ANALYSIS OF ALL LEVELS <div style="border-bottom: 1px solid blue; height: 15px; width: 100%;"></div> <div style="border-bottom: 1px solid blue; height: 15px; width: 100%;"></div> <div style="border-bottom: 1px solid blue; height: 15px; width: 100%;"></div> <div style="border-bottom: 1px solid blue; height: 15px; width: 100%;"></div> </div> </div> <div style="width: 50%;"> <div style="background-color: #333; color: white; padding: 5px; text-align: center; margin-bottom: 10px;">MULTI LEVEL SEARCHING</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">SEARCH STRING</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">OPERATOR(AND , OR NOT) ▼</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">SEARCH STRING</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">OPERATOR(IS EQUAL etc) ▼</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">COLUMNS CHECK BOXES ▼</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">OPERATOR(CONTAINS etc) ▼</div> <div style="text-align: right; margin-right: 10px;"> <div style="background-color: #eee; padding: 5px; margin-bottom: 10px; border: 1px solid #ccc;">CLEAR FILTER</div> <div style="background-color: #eee; padding: 5px; border: 1px solid #ccc;">SEARCH</div> </div> </div> </div> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 20px;"> <thead> <tr> <th>UPDATE</th> <th>DELETE</th> <th>TITLE</th> <th>AUTHOR</th> <th>YEAR</th> <th>CALL NO</th> <th>STOCK</th> <th>PRICE</th> <th>RATING</th> </tr> </thead> <tbody> <tr><td>EDIT</td><td>DEL</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td></tr> <tr><td>EDIT</td><td>DEL</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td></tr> <tr><td>EDIT</td><td>DEL</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td></tr> <tr><td>EDIT</td><td>DEL</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td></tr> <tr><td>EDIT</td><td>DEL</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td></tr> <tr><td>EDIT</td><td>DEL</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td></tr> <tr><td>EDIT</td><td>DEL</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td></tr> <tr><td>EDIT</td><td>DEL</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td></tr> <tr><td>EDIT</td><td>DEL</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td><td>Content</td></tr> </tbody> </table>	UPDATE	DELETE	TITLE	AUTHOR	YEAR	CALL NO	STOCK	PRICE	RATING	EDIT	DEL	Content	Content	Content	Content	Content	Content	Content	EDIT	DEL	Content	Content	Content	Content	Content	Content	Content	EDIT	DEL	Content	Content	Content	Content	Content	Content	Content	EDIT	DEL	Content	Content	Content	Content	Content	Content	Content	EDIT	DEL	Content	Content	Content	Content	Content	Content	Content	EDIT	DEL	Content	Content	Content	Content	Content	Content	Content	EDIT	DEL	Content	Content	Content	Content	Content	Content	Content	EDIT	DEL	Content	Content	Content	Content	Content	Content	Content	EDIT	DEL	Content	Content	Content	Content	Content	Content	Content
UPDATE	DELETE	TITLE	AUTHOR	YEAR	CALL NO	STOCK	PRICE	RATING																																																																																			
EDIT	DEL	Content	Content	Content	Content	Content	Content	Content																																																																																			
EDIT	DEL	Content	Content	Content	Content	Content	Content	Content																																																																																			
EDIT	DEL	Content	Content	Content	Content	Content	Content	Content																																																																																			
EDIT	DEL	Content	Content	Content	Content	Content	Content	Content																																																																																			
EDIT	DEL	Content	Content	Content	Content	Content	Content	Content																																																																																			
EDIT	DEL	Content	Content	Content	Content	Content	Content	Content																																																																																			
EDIT	DEL	Content	Content	Content	Content	Content	Content	Content																																																																																			
EDIT	DEL	Content	Content	Content	Content	Content	Content	Content																																																																																			
EDIT	DEL	Content	Content	Content	Content	Content	Content	Content																																																																																			

UI Component Name	Type of UI component	Purpose of UI Component/Other details
Combo Box	Input Widget	Multiple Combo Boxes are used to make users select column, algorithm and level of <i>Multi-Level Sorting</i> and operators & columns in <i>Multi-Level Searching</i> .
Icons	Image	Icons for Menu.
Labels	Display Widget	To display headings and outputs.
List	Item Widget	To display levels in <i>Multi-Level Sorting</i> .
Push Buttons	Button	To <i>Add level</i> , <i>Clear Filter</i> and <i>Search</i> push buttons.
Table	Item Widgets	To display list of selected entities (Book and its attribute).
Text Box	Input Widget	To take input for search strings.

UPDATE BOOK

TITLE

Title

AUTHOR NAME

AUTHOR NAME

CALL NO/ISBN

CALL NO/ISBN

PUBLISH YEAR

PUBLISH YEAR

PRICE

PRICE

RATING

Rating

STOCK STATUS

STOCK STATUS

UPDATE