VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Machhe, Belagavi, Karnataka 590018

FILE STRUCTURES LABORATORY WITH MINI-PROJECT REPORT

On

FOOD ORDERING SYSTEM USING B+ TREES

Submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Engineering

in

Information Science & Engineering

By

Pooja S (1BG19IS031)

Vidyayāmruthamashnuthe

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Information Science and Engineering

2021 – 2022

B.N.M. Institute of Technology

An Autonomous Institution under VTU

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Vidyayāmruthamashnuthe

CERTIFICATE

Certified that the Mini-project entitled **FOOD ORDERING SYSTEM USING B+ TREES** is carried out by **Ms.Pooja S** USN **1BG19IS031** the bonafide student of **B.N.M Institute of Technology** in partial fulfillment for the award of **Bachelor of Engineering** in **Information Science & Engineering** of the **Visvesvaraya Technological University**, Belagavi during the year 2021-2022. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The mini-project report has been approved as it satisfies the academic requirements in respect of mini-project prescribed for the said Degree.

Ms. Jamuna S Murthy	Dr. Shashikala
Asst. Professor, Dept. of ISE	**Prof & Head, Dept. of ISE**
BNMIT	BNMIT

Name & Signature of the Examiners with date:

1.

2.

Table of Contents

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Objective

File Structure is the combination of representations for data in files and of operations for accessing the data. An improvement in file structure design may make an application hundreds of times faster.

A File Structure allows applications to read, write and modify data. It might also support finding the data that matches some search criteria or reading through the data in some particular order. A good File Structures aims at:

- Fast access to great capacity.
- Reduce the number of disk access by collecting data into buffers, blocks or buckets.
- Manage growth by splitting these collections.

A B+ tree is an N-ary tree with a variable but often large number of children per node. A B+ tree consists of a root, internal nodes and leaves. The root may be either a leaf or a node with two or more children. A B+ tree can be viewedas a B-tree in which each node contains only keys (not key—value pairs), and to which an additional level is added at the bottom with linked leaves.

1.2 Scope

With the development of internet technology, the scope of Online Food Order System has really improved. At present, around 2.1 of business organizations are managing their financial transactions through online payments. In 2022, over 2.14 billion people worldwide are expected to buy goods and services online.

Purchasing goods and services online has become a common practice among many people around the world. Some choose to make online purchases for convenience, others because of the competitive price offered by some e-commerce platforms .Our online food ordering platform allows users to order dishes from their favorite restaurants and have them delivered within a reasonable timeframe.

1.3 Motivation

It is common for everyone to wait in line in front of food trucks or restaurants, and sometimes it will take more than 30 minutes for orders to be taken, since in addition to waiting in line, the cooking process may also be slow.

This is the motivation behind our solution: The Delivery Express! By offering customers smart food delivery website, a problem can be solved. A reduction in the number of people jamming in front of restaurants will also improve the hygiene and security conditions outside the campus.

CHAPTER 2

METHODOLOGY

2.1 Technique

A B+ tree is an advanced form of a self-balancing tree in which all the values are present in the leaf level.An important concept to be understood before B+ tree is multilevel indexing. In multilevel indexing, the index of indices is created as in figure below. It makes accessing the data easier and faster. Fig2.1 shows structure of B+Trees

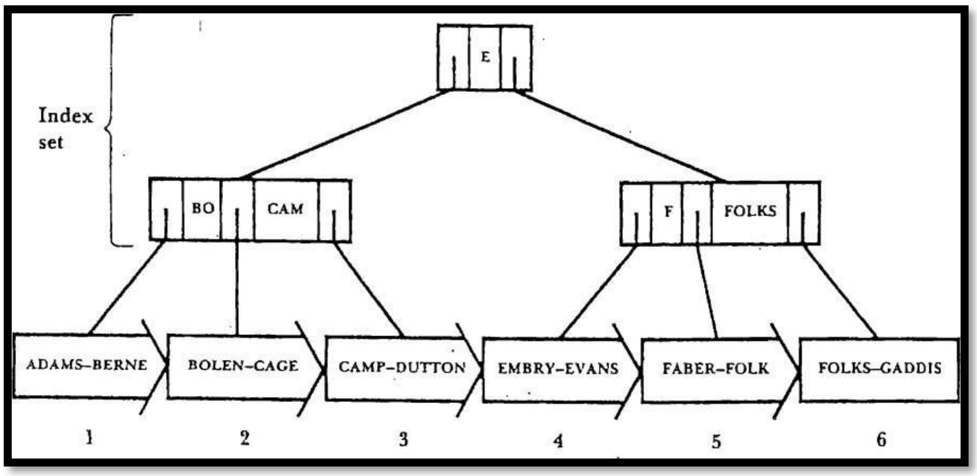


Fig 2.1 B+ Tree

Properties of a B+ Tree

- All leaves are at the same level.
- The root has at least two children.
- Each node except root can have a maximum of m children and at least m/2 children.
- Each node can contain a maximum of m - 1 keys and a minimum of]m/2] - 1 keys.

2.2 Tools

- **VISUAL STUDIO CODE -** It is commonly know as VS CODE is a source code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

It can be used with other programming languages including Java , JavaScript , GO , Node.js, Python ,C++ and Fortran.. It is based on the Electron framework, which is used to develop Node.js Web applications that run on the Blink layout engine.

- **Anaconda Prompt** - Anaconda prompt is similar to a terminal or command prompt (cmd). It refers to a black screen used to type in the commands by the user.

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATION

3.1 User Requirements

User using our delivery express website require minimal requirements.

- **Browser –** To run our website user require browser like Chrome, Internet Explorer etc.
- **Command prompt** – to update the details of B+ trees user will require a terminal to execute user commands

3.2 Software Requirements

- Operating System: - Windows 10 64-bit
- Chrome Browser
- Command prompt

3.3 Hardware Requirements

- Processor – Intel core i5 7400 or i7 i7-7700k clocked CPU @3.40GHz
- RAM – 16 GB DDR3
- Hard Disk – 1TB at 7200 rpm
- GPU – Nvidia GTX 1050 or Intel 6400

3.4 Functional Requirements

In this project the functional requirements of the online order delivery are defined

- Initialization of Order – Addition of items in the menu by the admin .
- Admin Authentication
- Searching of order placed .
- Processing of authorization - transmission of a request for authorization of the order to a producer, request for acceptance of the transaction, authorization of the transaction.
- User interface and information management.

3.5 Non - Functional Requirements

Details of non-functional requirements (NFRs) that describe system attributes such assecurity, reliability, In maintainability, scalability, and usability.

- **Security :**Provider systems shall resist unauthorized, accidental orunintended usage and provide access only to legitimate users.

- **Maintainability :** Provider systems shall be designed to optimize the abilityof maintenance personnel to revise or enhance it.

- **Scalability:** Provider systems shall be designed to accommodate increased volumes, workloads and users.

- **Usability** - Provider and consumer systems should follow the ISO 13407 / ISO9241-210 to explain a user-centered design process.

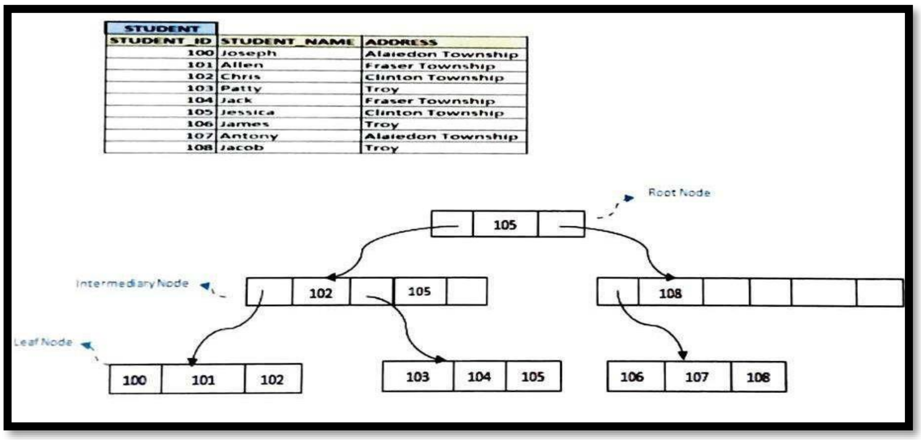
CHAPTER 4

SYSTEM DESIGN AND DEVELOPMENT

4.1 Architectural Design

B+ tree is a storage method in a tree like structure . B+tree has one root, any number of intermediary nodes (usually one) and a leaf node. Here all leaf nodes will have the actual records stored. Intermediary nodes will have only pointers to the leaf nodes; it not has any data. Any node will have only two leaves. This is the basic of any B+ tree.

Fig 4.2 refers to STUDENT table below. This can be stored in B+ tree structure as shown below. We can observe here that it divides the records into two and splits intoleft node and right node. Left node will have all the values less than or equal to root node and the right node will have values greater than root node. The intermediary nodes at level 2 will have only the pointers to the leaf nodes. The values shown in the intermediary nodes are only the pointers to next level. All the leaf nodes will have the actual records in a sorted order.



STUDENT

STUDENT_ID	STUDENT_NAME	ADDRESS
100	Joseph	Alaiedon Township
101	Allen	Fraser Township
102	Chris	Clinton Township
103	Patty	Troy
104	Jack	Fraser Township
105	Jessica	Clinton Township
106	James	Troy
107	Antony	Alaiedon Township
108	Jacob	Troy

Fig 4.2 Architectural design using B+ Tree

When the user wants to search for a record, he can search for it in the leaf nodes only. Hence searching for a particular record takes time since they are all of same size. Also, they are sorted, hence this is a sequential search.

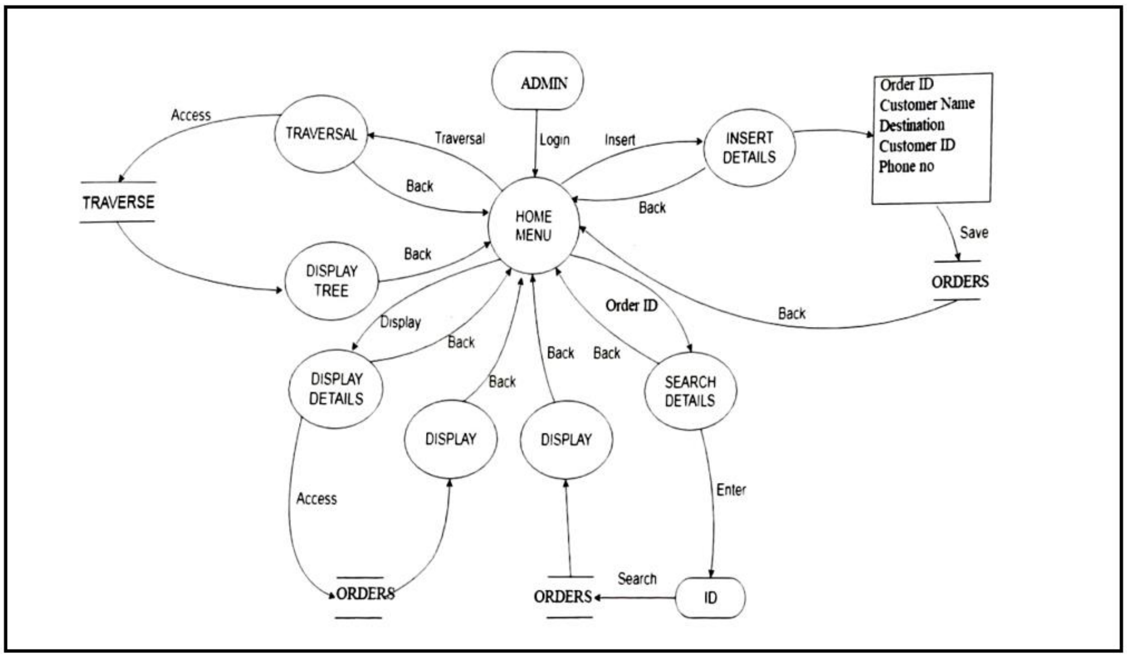


Fig 4.3 Dataflow Diagram of Food delivery

Fig 4.3 shows how the flow of control of execution is done across several functions given by the admin to the user. Initially, the user, i.e., the admin in this case enters the main home menu, in which he can opt for a function out of search, insert, display and traversal. The dataflow diagram consists of mainly four notations, i.e. functions usedto denote what is the action to be executed by the user. File is used to depict a location which stores the record entries. I/O is used to provide interface

between user and the system by providing inputs to retrieve the particular outputs whenever required, like using ID to retrieve the whole record or inputting name and passwordto login to the home page. Lastly the arrow depicts the direction of the flow of control between various functions via the front end.

Initially the admin/user is given a set of functions to select from. If he opts for insert, then the details can be entered Into the insert page and finally saved to thefile where the records are currently being stored. Return to home page by clicking Back button.

Similarly, if he opts for search button, then it opens a search page where theuser is supposed to enter the order ID to be searched. If the system finds it in the filesystem, then it returns the complete record entry corresponding to the Order ID given record.

.

The display function is simply used to display all the records given into the file as inputs in a Table form, where the fields are separated by columns indicating the field header.

Finally traversal function is opted to see how the records have been traversed using the B+ Tree by retrieving and accessing all the records from the rent file and displaying it in a tree structure by representing the Order ID's as the blocks of the tree. It will be displayed line by line to represent the levels of the tree and the depth up to which the records have been entered.

CHAPTER 5

IMPLEMENTATION

B+ Tree is an advanced method of ISAM file organization. It uses the same concept of key-index, but in a tree like structure. B+ tree is similar to binary search tree, but it can have more than two leaf nodes. It stores all the records only at the leaf node. Intermediary nodes will have pointers to the leaf nodes. They do not contain any data/records.

5.1 Modules Implemented

The front end is where the graphical user interface happens. It consists of the following operations supported by B+ Trees which is used to implement order objects, as listed below

- **Insert :** Module inserts the order in an entry sequenced manner that is pre-defined and helps in proper management of all the information, which is records consisting of required order field entries
- **Search :** Module searches the order with the help of the order-id entered by the adminand displays the order record information of the corresponding record.
- **Display :** Module is used to review all the information that is stored in a file at once.This module can be used to see all records that is stored in a file.

5.2 Algorithms

Find Leaf: This function is used to find the leaf node where we can insert a new node or used to search for a particular node. The algorithm for the function is shown below:

Step l : Start

Step 2: Initialize two nodes ,ptr and prevptr and declare an integer variable i

Step 3: Till ptr equals NULL, increment the value of I to find the size of the tree ,after that point the pointer to the next node of the tree.

Step 4: return prevptr

Step5:End

Insertion of a Node:

Step 1: Every element is inserted into a leaf node. So, go to the appropriate leaf node.

Step 2: Insert the key into the leaf node in increasing order only if there is no overflow. If there is an overflow go ahead with the following steps mentionedbelow to deal with overflow while maintaining the B+ Tree properties.

Case 1: Overflow in leaf node

Step1: Split the leaf node into two nodes.
Step2: First node contains **ceil((m-1)/2)** values.
Step3: Second node contains the remaining values.
Step4: Copy the smallest search key value from second node to the parent node.(Right biased)

Case 2: Overflow in non-leaf node

Step1: Split the non leaf node into two nodes.

Step2: First node contains ceil(m/2)-1 values.

Step3: Move the smallest among remaining to the parent.

Step4: Second node contains the remaining keys.

Traverse - This function traverse the tree to find the correct position where the node can be Inserted. The algorithm for the function is shown below

Step 1 : Start

Step 2: If n equals NULL, then return to the previous node.
Step 3: If n.size less than 4, then the insertlntoNode Function is called, which inserts thevalue into the concerned node and then return to the previous node.

Step 4: If n equals root, then create a new root node and make it point to the parent node.

Step 5. Declare a new node that splits the parent node with the help of the split functiontakes the node to be splitted as an argument.

Step 6: If key is less than n.a[0],then declare a object of node class and assign it the value ol' newptr else assign the value of n.

Step 7. Call the InsertlntoNode lilnction with the arguments, n initialized to t, key to keyand address to address to Insert a node into the tree.

Step 8: Call traverse function for the arguments, n initialised to parent node, key to the number of values stored in n and address to n.

Step 9: Repeat the process from Step 6 to execute promote function for the arguments, n initialised to newptr parent node, key to number of values stored in newptr and addressto newptr

Step10: End

Update: This function is used to update the values of a node. The algorithm for the function is given below:

Step l : Start

Step 2: Check if the parent exists or not.

Step 3: If parent.size equals zero, then return the node.

Step 4: Declare an integer variable old key that stores the value of old key present in theparent node currently.

Step 5: Find the node that contains the old key and replace the old key with the new key.

Step 6: End

Split: This function is used to split a node when the node is full and we try to enter onemore value into that node. The algorithm for the function is shown below:

Step l : Start

Step 2: Declare two integer variables midpoint and newsize and initializeit, also declare two objects of node class, i.e., newptr and child.

Step 3: Store the values of the current node to the newptr and change the values of the current node.

Step 4: Initialize the node size to midpoint.

Step 5: Initialize the new pointer Size to new size.

Step 6: For all the elements in the tree, assign the pointer to next node to child node. If child not equals NULL, then child pointer parent node points to the newptr node.

Step7: return newptr

Step 8: End

Insert Module - This module is used to insert the records into the file system field by field by asking the user admin to enter a set of details required, which in this case is the order details. The key value determines a record's placement in a B+ tree. The leaf pages are maintained in sequential order and a doubly linked list connects each leaf page with its sibling page(s).

Step l: Start

Step 2: Input Order record details

Step 3: Check if the tree is empty or not. If root equals NULL, then create a root node, assign it a key value and increment the node size.

Step 4: If the tree is not empty, then find the leaf node with the help of key and level, using the findLeaf function.

Step 5: If leaf node equals key, then the record to be inserted alreadyexists.

Step 6: Promote function is called to search the correct position wherethe key can be inserted

Step 7: The insertIntoNode function is called to insert element into respective available nodes.

Once Inserted, update the key value of parent node accordingly, along with which the parent node is split using split function. The function places the values in the child nodes such that each child node carries values lesser than or equal to its parent node.

Step 8: End

Search Module — The Order ID acts as the key to be searched, where the program compares it with all the records in the file to match that ID. If a match exists then the search is successful otherwise it is unsuccessful.

Step l: Start

Step 2: Input Order ID that the user wants to search

Step 3: Check if the tree exist or not before searching the id. If root equals NULL, then the tree does not exist.

Step 4: If the tree is not empty, then find the leaf node with the help of key and level, using the findLeaf function.

Step 5: Declare a flag variable to keep a track if the node exists and initialize it to zero. Step 6: If leaf matches the key, initializeflag=l, and display Order 'key' with its respective record.

Display Module – All the records stored in file are displayed inTable form

Step1: Start

Step2:Create an array of objects of class records

Step3:Open the text file in input mode and store all the data inrecords.

Step4: Create a Table

Step6:end

Traversal Module — All the Order ID's stored in traversal file are displayed in atree format, accordingly in a sorted manner while divided into levels.

Step l: Start

Step 2: Input the pointer to the root node

Step 3: If pointer doesn't exist, then return to previous node.

Step4 : Open the text file in output mode and store the value of all the nodes in that text file.

Step5:Display the file values in a tree format using the bufferef writer function

Step6:Repeat from step2 with ptr.next[i] as the argument of the function.

Front End:

HTML and CSS are used as front end for our Mini project user has following options.

Menu : It display the list of items available from which a user can place his/ her order

Add Item: Admin can add item into the menu. Items are inserted into the menu in aentry sequenced fashion.

Search Item: Items /order is searched using order id and is displayed to the admin /user whether the order is valid or not.

B+ trees- It displays the added order id in a tree fashion where information is presentat the leaf node and is sorted using Order Id.

Back End: Flask is used as Backend for our project.

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It hasno database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself.The different operations used in the program are also written in java class which helps the program for internal operation of the function. Every function has a call to other function internally and Flask helps in the proper execution of these functions.

CHAPTER 6

RESULTS AND DISCUSSION

6.1 Snapshots of the project and description

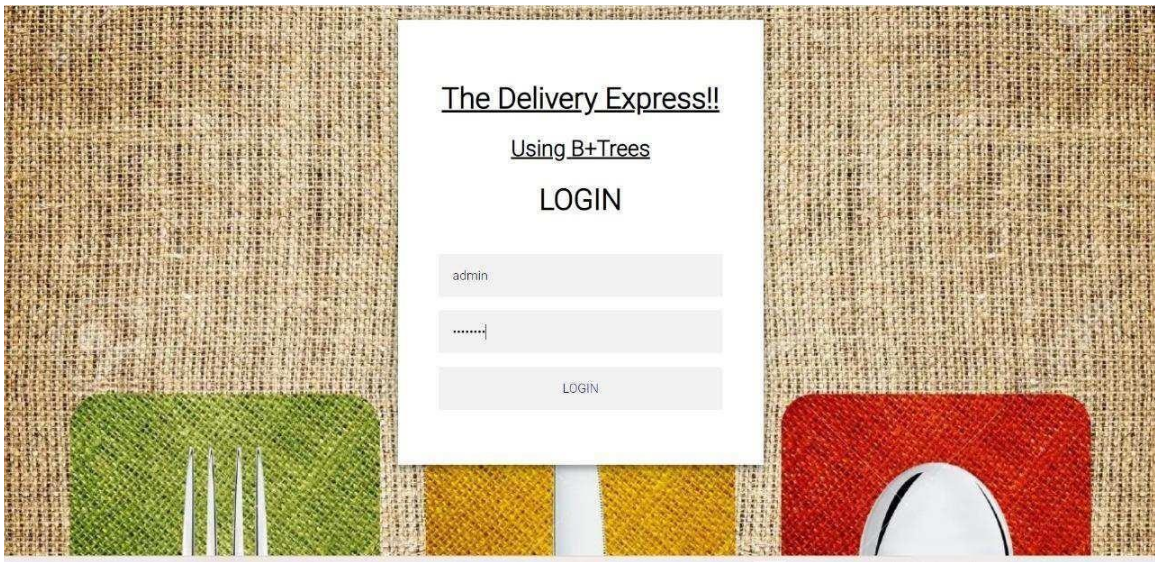
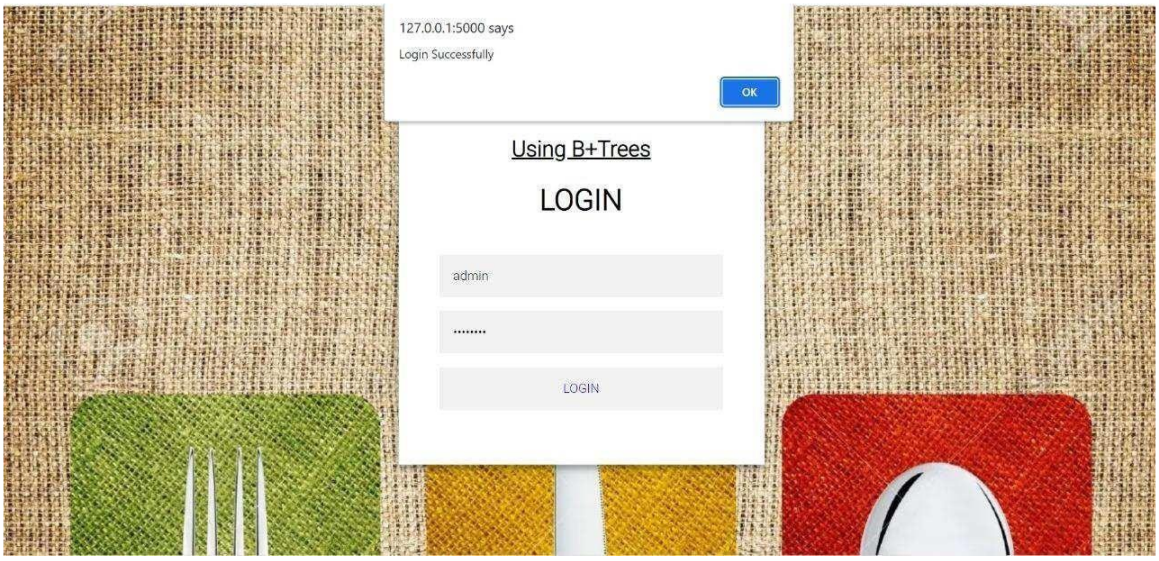


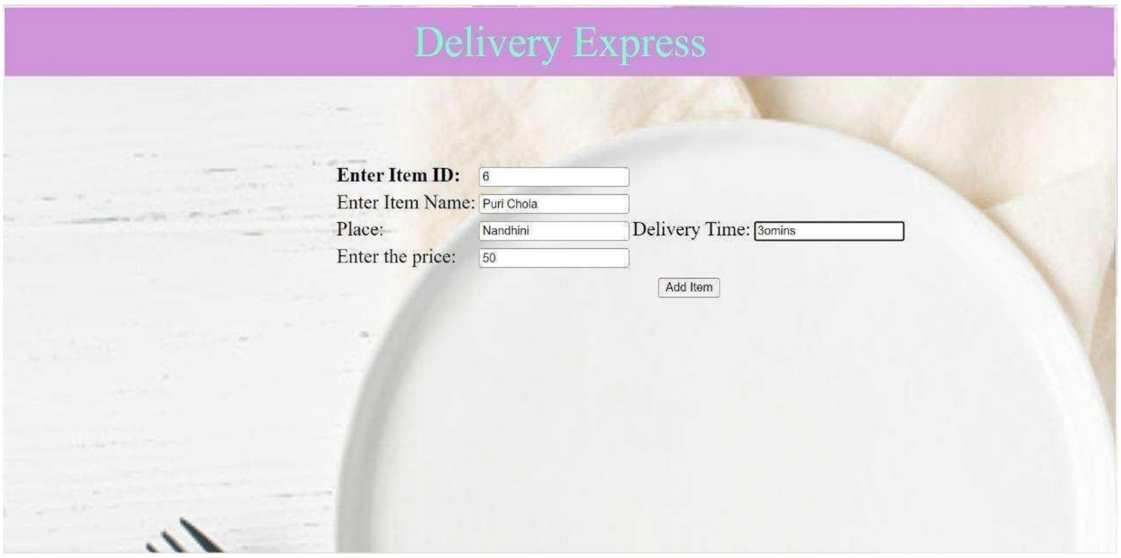
Fig 6.1 Login Page


127.0.0.1:5000 says

Login Successfully

Fig 6.2 On Successful Login

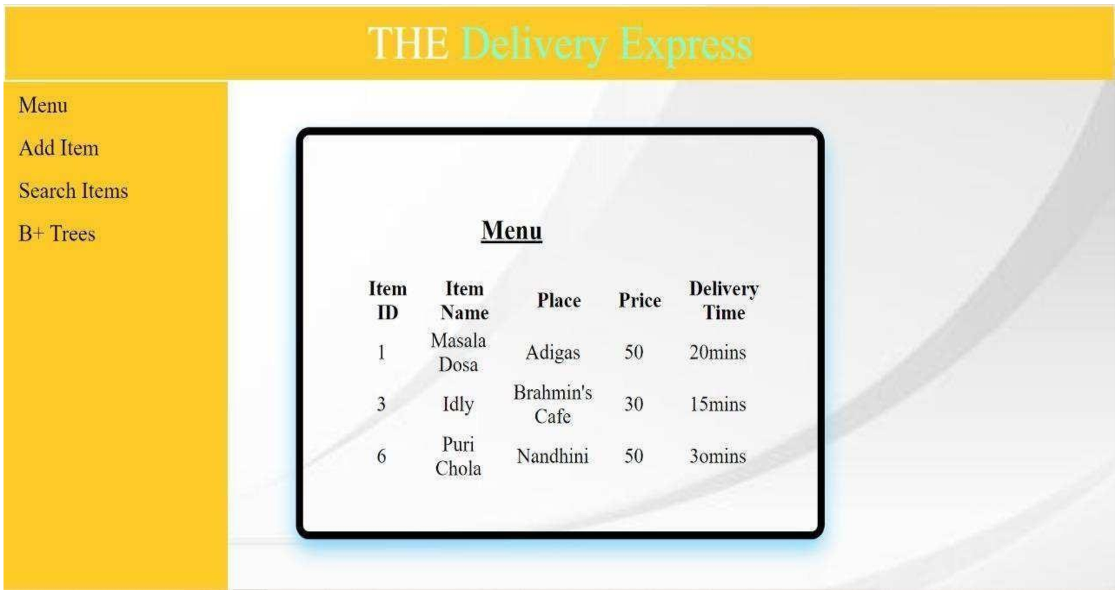
Fig 6.1 refers to Login page which asks the admin to enter correct username and password to login into the website. Fig 6.2 shows successful login by admin and has following options Menu, Search item, Add item ,B+trees.



Delivery Express

Enter Item ID:	6		
Enter Item Name:	Puri Chola		
Place:	Nandhini	Delivery Time:	3omins
Enter the price:	50		

Fig 6.3 Add Item



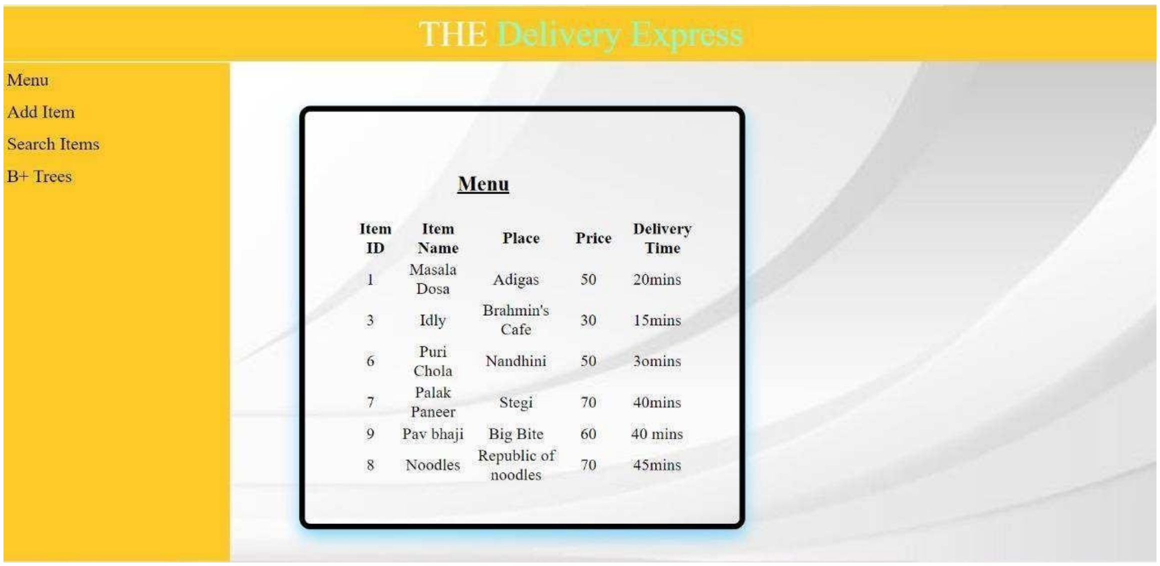
THE Delivery Express

Menu

Item ID	Item Name	Place	Price	Delivery Time
1	Masala Dosa	Adigas	50	20mins
3	Idly	Brahmin's Cafe	30	15mins
6	Puri Chola	Nandhini	50	3omins

Fig 6.4 Menu After Adding Item

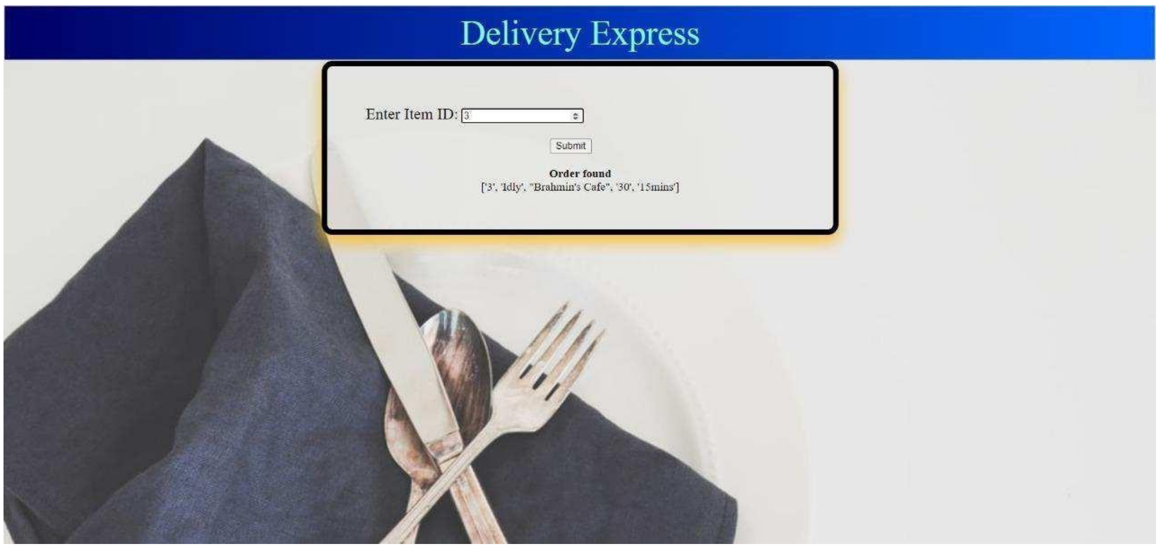
Fig 6.3 shows On clicking Add item following page opens and admin can add new item into the menu by entering the item id , item name , place, price and the delivery time.Fig 6.4 refers to once user enters all the required fields and clicks Add item ,Item will be added to the menu.



Menu

Item ID	Item Name	Place	Price	Delivery Time
1	Masala Dosa	Adigas	50	20mins
3	Idly	Brahmin's Cafe	30	15mins
6	Puri Chola	Nandhini	50	3omins
7	Palak Paneer	Stegi	70	40mins
9	Pav bhaji	Big Bite	60	40 mins
8	Noodles	Republic of noodles	70	45mins

Fig 6.5 Menu

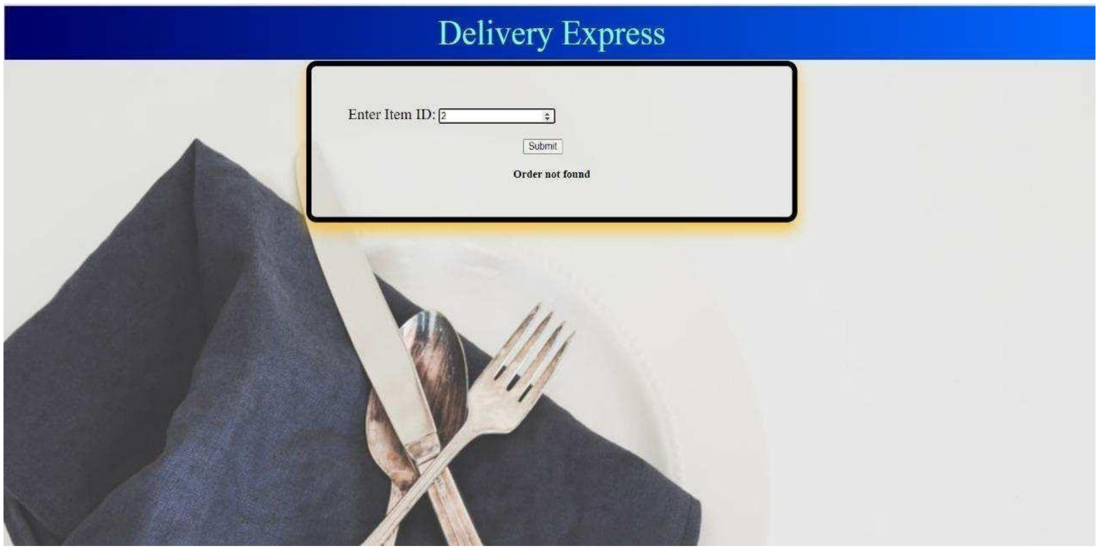


Enter Item ID: 3

Order found
['3', 'Idly', "Brahmin's Cafe", '30', '15mins']

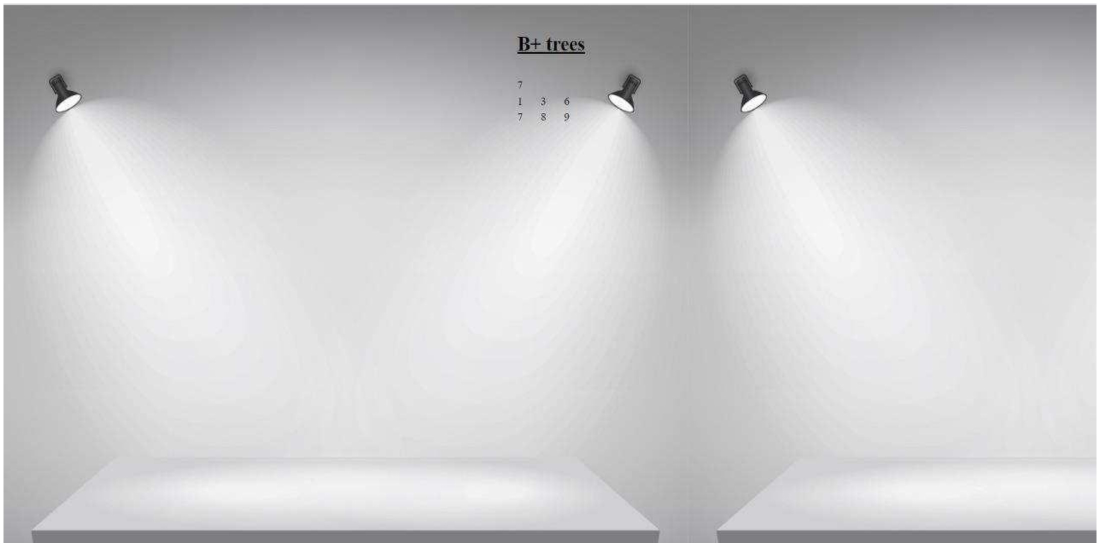
Fig 6.6 On Successful Search

Fig 6.5 refers to our menu which is entry sequenced.Fig 6.6 shows how Admin can find a particular order from the menu by entering the order id. If the order-id is present in the menu it will display the order found message alongwith the details of the order.



Order not found

Fig 6.7 On Unsuccessful Search



```
7
1   3   6
7   8   9
```

Fig 6.8 B+ Tree Structure

Fig6.7 refers to if the entered order id is not present in the menu it will display a message Order not found.Fig 6.8 shows on clicking on B+ trees option, a tree structure where information is stored in the leaf nodes will be displayed in sorted order of item-id

6.2 Observation about the project

Some of the observations of B+ trees are :

- The method is less efficient for static tables.
- Any search will end at the leaf node only.
- Waste of Memory
- Time Complexity for every search result in O(h) where h- height of the B+ tree.

6.3 Advantages of B+ trees

- Records can be fetched in equal number of disk accesses.
- Height of the tree remains balanced and less as compare to B tree.
- We can access the data stored in a B+ tree sequentially as well as directly.
- Keys are used for indexing.
- Faster search queries as the data is stored only on the leaf nodes.

6.4 Disadvantages of technique

- There is no rearrangement of nodes while insertion or deletion, then it would be an overhead.
- It takes little effort ,time and space
- But this disadvantage can be ignored compared to speed of traversal

CHAPTER 7

CONCLUSION

This project has incorporated Flask python language and Html and CSS with B+ Tree technique to provide smooth transition and connectivity between the front end which the user has access to, and the back-end which hold the data in the form of set of fields. Once a record has been added, the admin can utilize the backend as storage medium from which records can be accessed and modified by the admin as per the user's requirements.

REFERENCES

1. Michael J.Folk, Bill Zoellick, File Structure, 2nd edition
2. A.V. Oppenheim and R.W. Schafer, Digital Signal Processing, Englewood, N.J.,Prentice Hall, 3 Edition, 1975.
3. https://www.w3schools.com
4. https://code.visualstudio.com