**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**BELAGAVI**

**FILE STRUCTURES MINI PROJECT REPORT ON**

**“COACHING INSTITUTE MANAGEMENT”**

Submitted in partial fulfilment for the requirements for the SIXTH semester

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION SCIENCE AND ENGINEERING**

For the Academic Year 2022-2023

Submitted By:

|  |  |
| --- | --- |
| SUSHANTH S RAO | USN: 1MV19IS059 |
| PRATHAM PAI B M | USN: 1MV19IS044 |
| DEEPANSHU BACHLOO | USN: 1MV19IS018 |
|  |  |

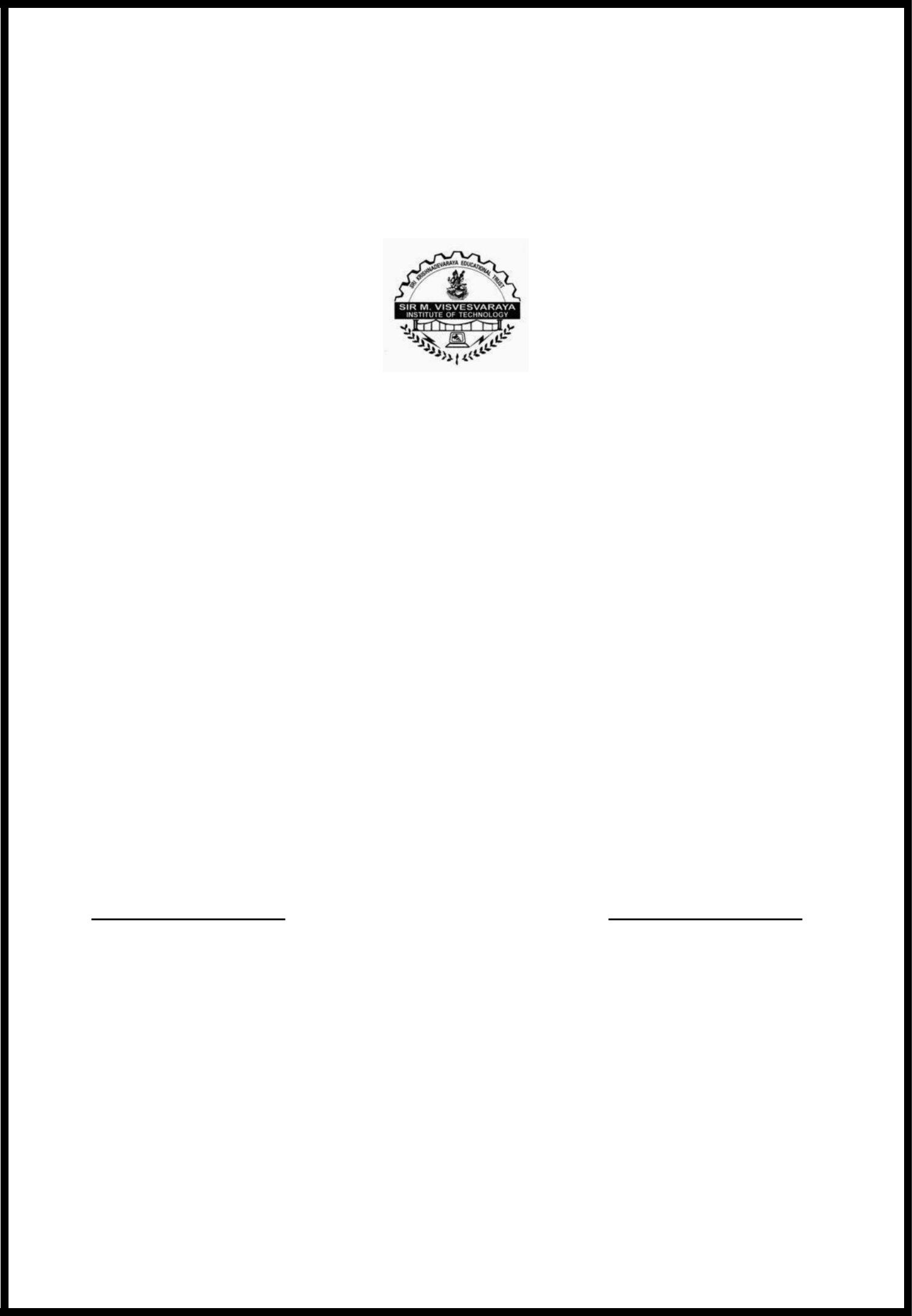
**Under the guidance of**

**Mr. Manohar R**

**Assistant Professor, Department of ISE, Sir MVIT**

**Department of Information Science and Engineering**

**Sir M Visvesvaraya Institute of Technology, Bengaluru**

**SIR M. VISVESVARAYA INSTITUTE OF TECHNOLOGY**

Krishnadevaraya Nagar, International Airport Road,

Hunasmaranahalli, Bengaluru – 562157

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

It is certified that the FILE STRUCTURES Mini Project work entitled “COACHING INSTITUTE MANAGEMENT" is carried out by SUSHANTH S RAO (1MV19IS059), PRATHAM PAI B M (1MV19IS044), DEEPANSHU BACHLOO (1MV19IS018) bonafide students of Sir M Visvesvaraya Institute of Technology in partial fulfilment for the 6th semester for the award of the Degree of Bachelor of Engineering in Information Science and Engineering of the Visvesvaraya Technological University, Belagavi during the academic year 2022. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the course of Bachelor of Engineering.

Name & Signature of Guide Name & Signature of HOD

Mr. Manohar R

Asst. Professor, Dept Of ISE,

Sir MVIT Bengaluru

Dr. P. Vijay Karthik

HOD, Dept Of ISE,

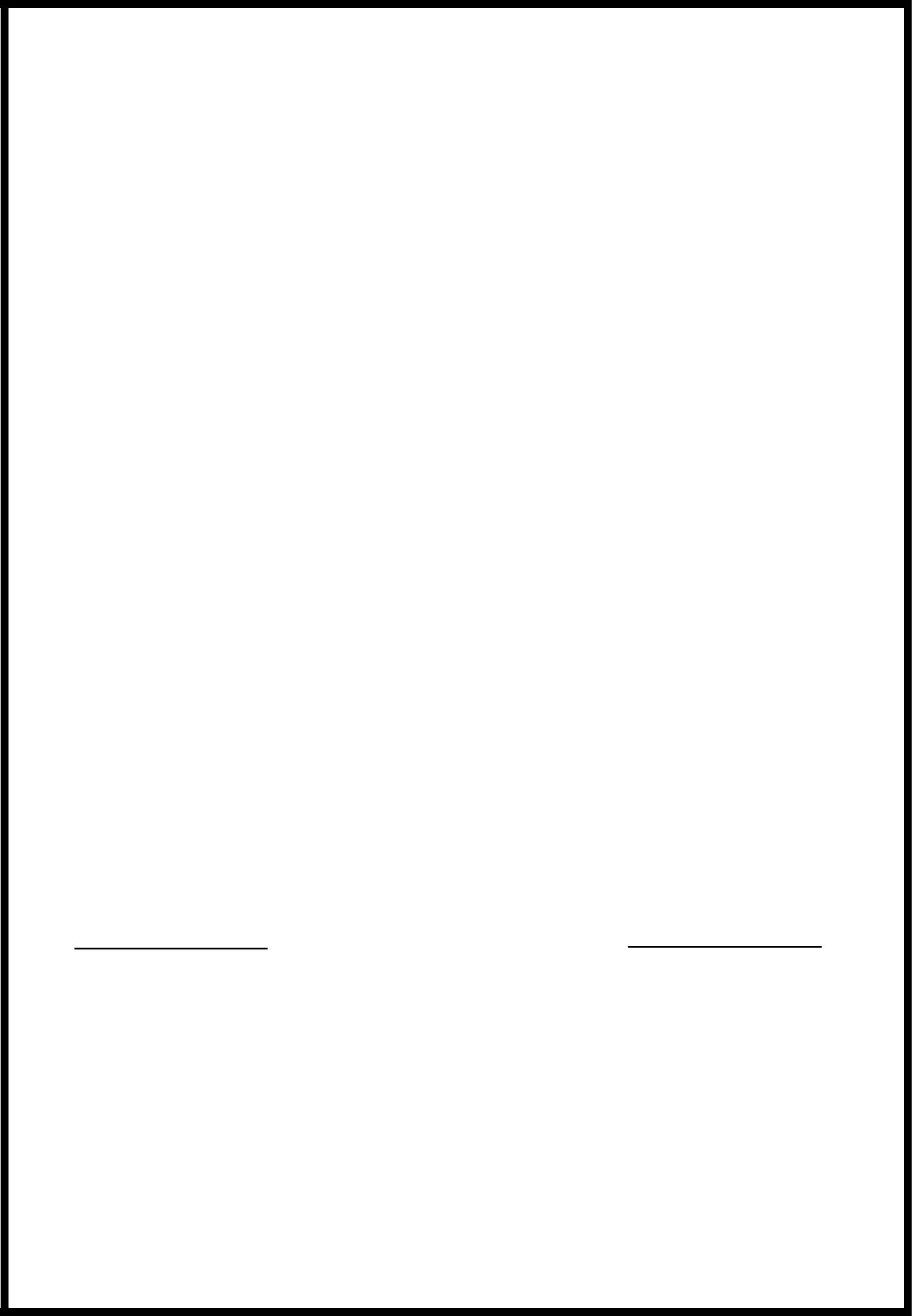
Sir MVIT Bengaluru

External Examination:

Name of Examiner Signature with Date

1)

2)



**DECLARATION**

We hereby declare that the entire project work embodied in this dissertation has been carried out by us and no part has been submitted for any degree or diploma of any institution previously.

Place: Bengaluru

Date:

Signature of Students:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

PRATHAM PAI B M

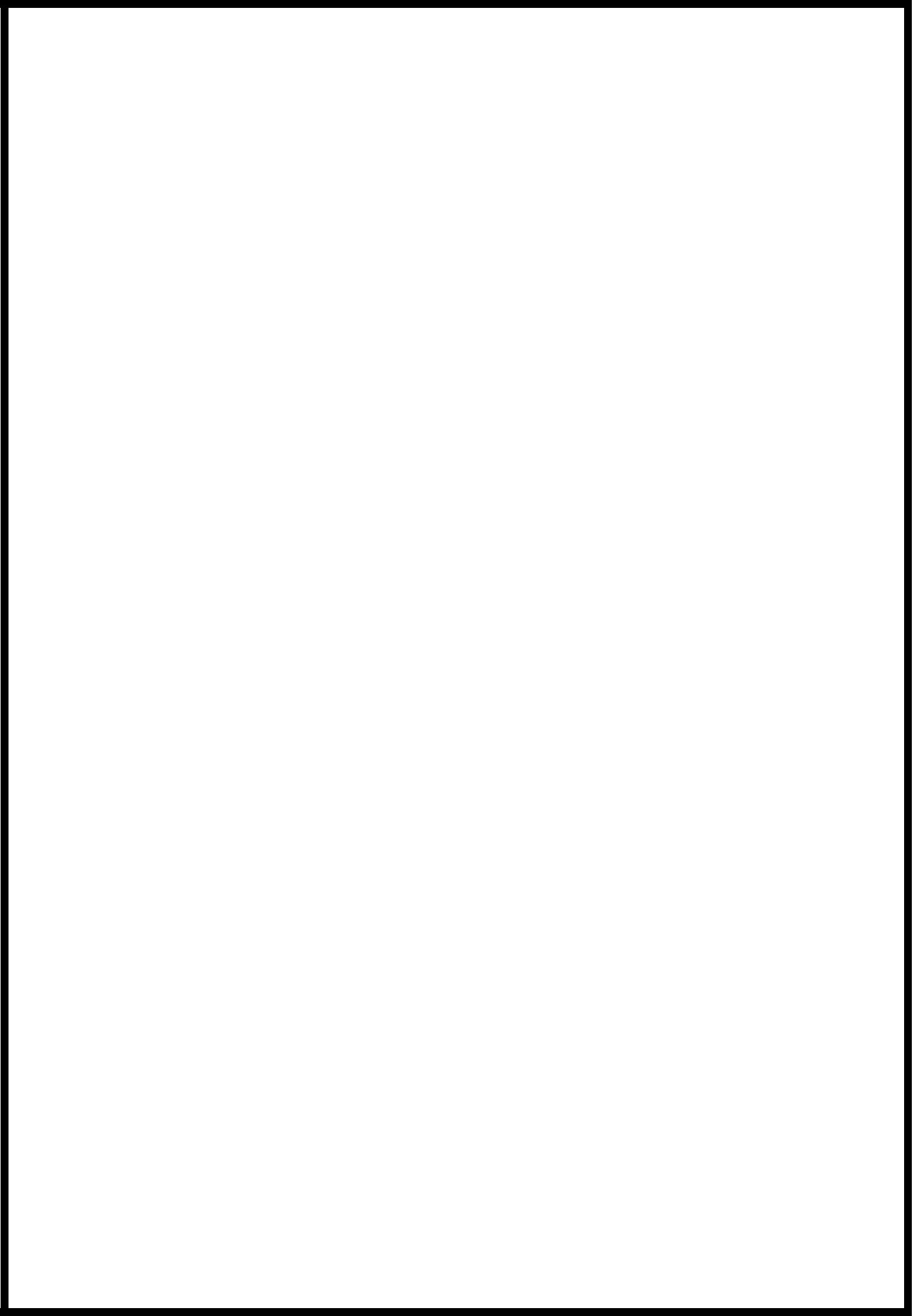
(1MV19IS044)

DEEPANSHU BACHLOO

(1MV19IS018)

SUSHANTH S RAO

(1MV19IS059)



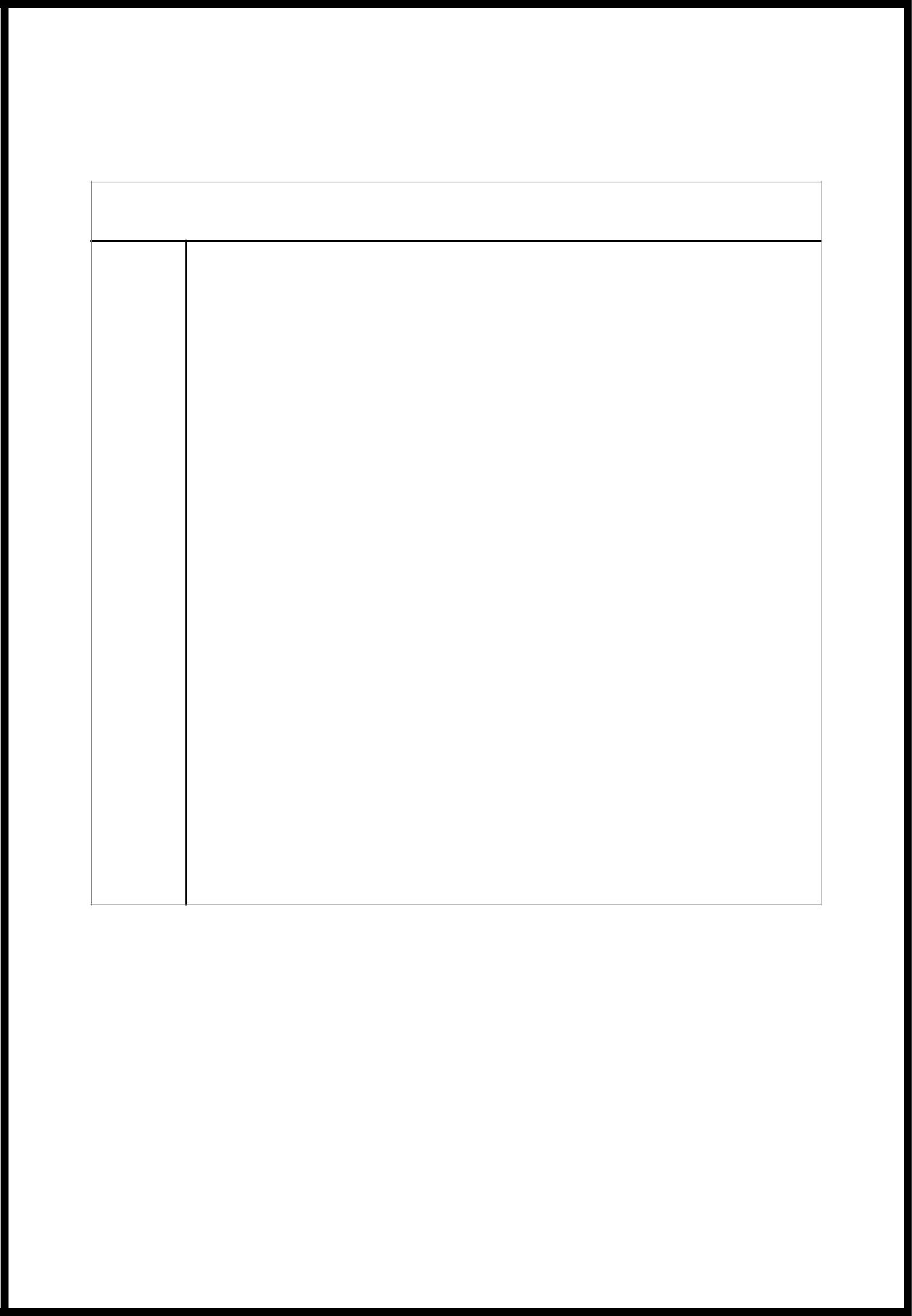
**ACKNOWLEDGMENT**

It gives us immense pleasure to express our sincere gratitude to the management of the Sir M. Visvesvaraya Institute of Technology, Bengaluru for providing the opportunity resources to accomplish our project work in their premises.

On the path of learning, the presence of an experienced guide is indispensable and we would like to thank our guide Mr Manohar R, Assistant Professor, Department of ISE, for his invaluable help and guidance. Heartfelt and sincere thanks to Dr. P. Vijay Karthik, HOD, Department of ISE, for his suggestions, constant support and encouragement. We would also like to convey our regards to Dr. V.R. Manjunath, Principal, Sir MVIT for providing us with the infrastructure and facilities needed to develop our project.

We would also like to thank the staﬀ of Department of Information Science and Engineering and lab-in-charges for their co-operation and suggestions. Finally, we would like to thank all our friends for their help and suggestions without which completing this project would not have been possible.

* DEEPANSHU BACHLOO
* PRATHAM PAI B M
* SUSHANTH S RAO



**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**INDEX**

CONTENTS

Introduction

Objective

Scope

Key Concepts:

File Structures

Indexing

Multilevel Indexing

Program Implementation:

Project Structure

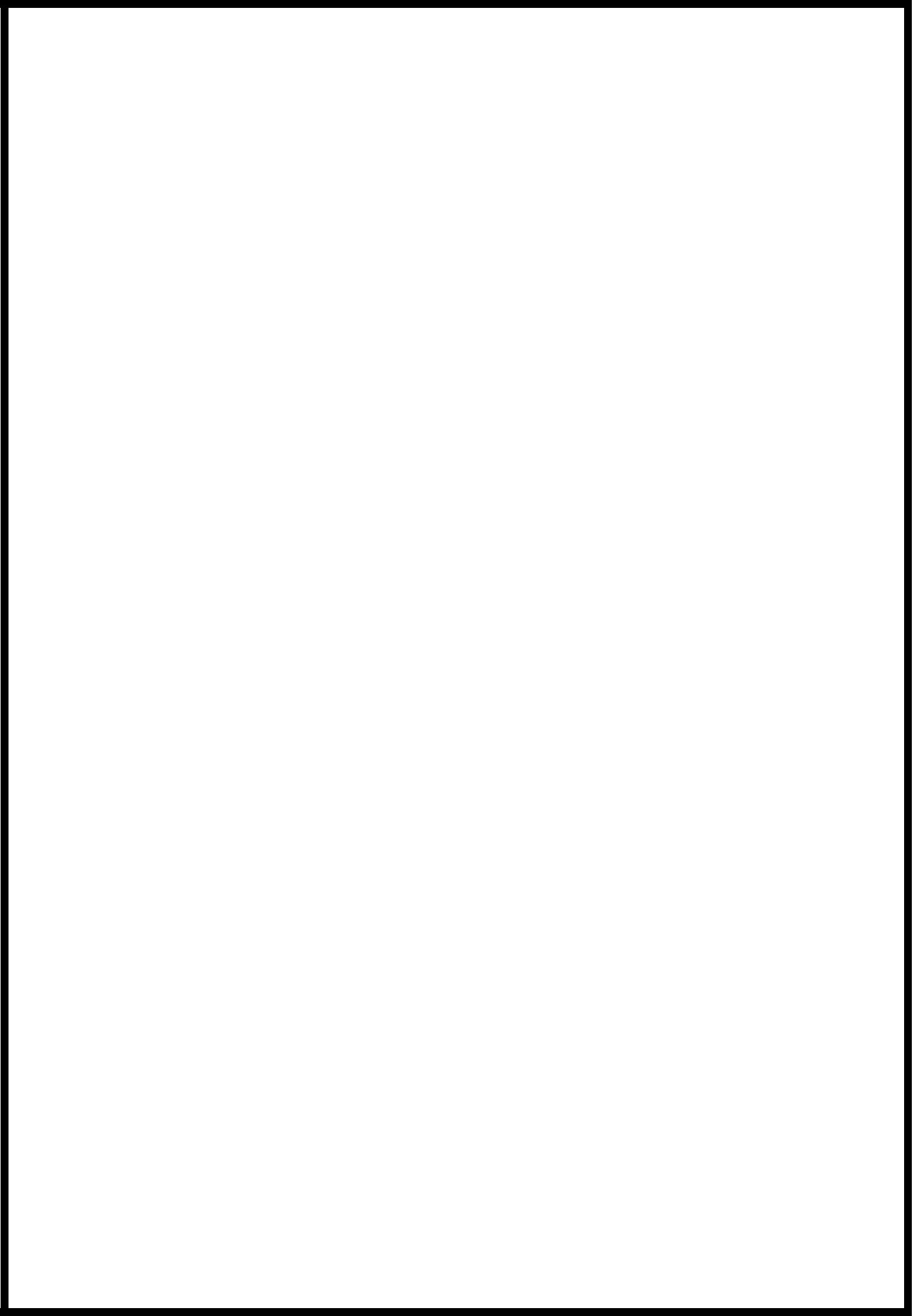
Header Files

Functionalities

Program Output

Conclusion

References

**INTRODUCTION**

The education plays a pivotal role in the life of a student. Especially in a growing country like India, education is developing and becoming a big part. Learning and Education is imbibed within our culture from the ancient times and a great importance is placed on academic success in our society. This has resulted in expansive rise of Coaching Centres.

These coaching institutes stretch and encompass a wide range. They include elementary coaching for students of grade 5 and up. But due to the significance of competitive examinations such as JEE and NEET, they take quite a large amount of share as they require extra and specialised learning. Moreover certain international exams and higher level studies such as GRE, IELTS, GMAT, CAT and so on are solely taught by coaching centres. These all makes the coaching centres very significant.

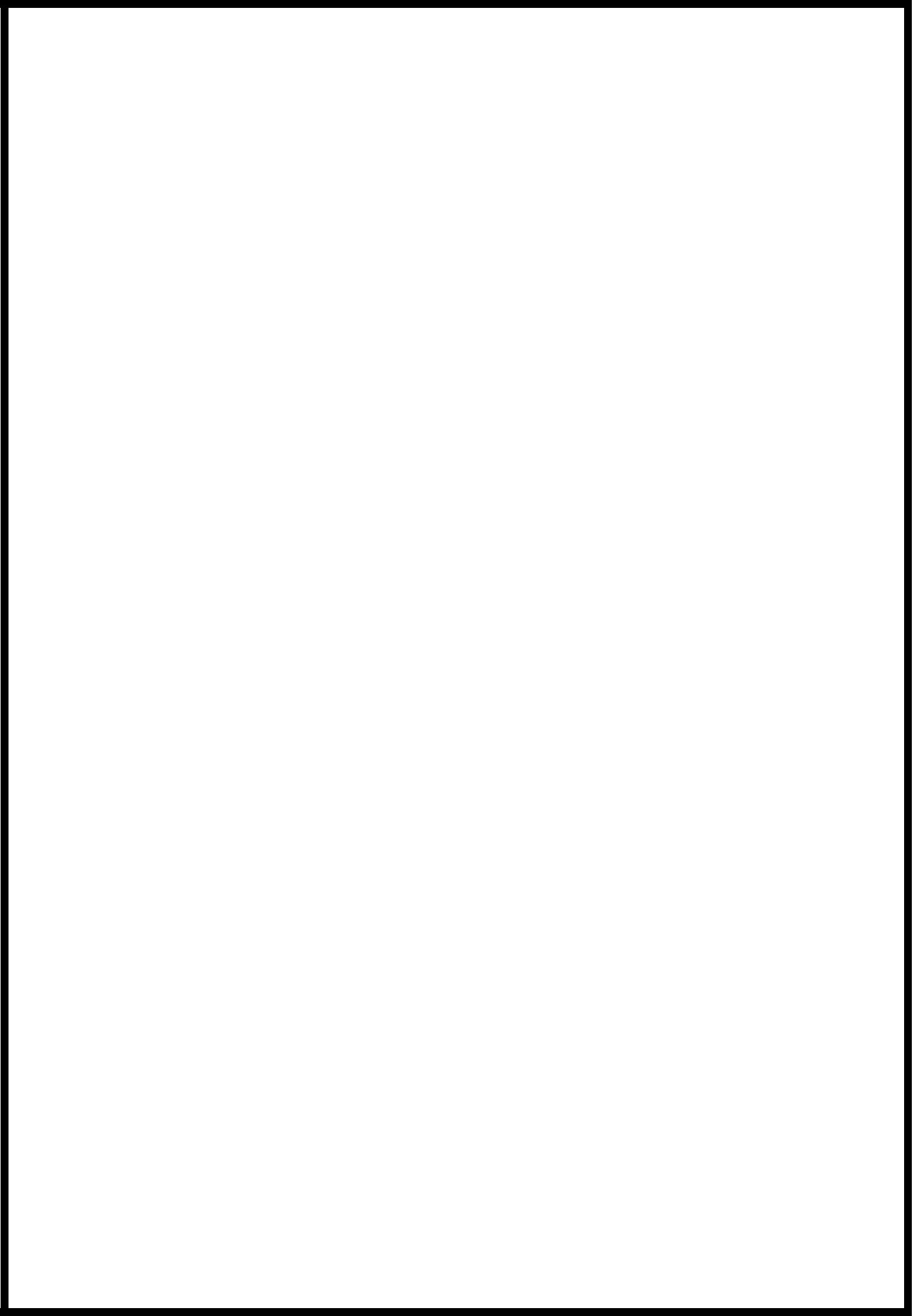
**Shape, square

Description automatically generatedOBJECTIVE**

The project objective is to help manage records of the Coaching Centre. This project is implemented by utilizing the concepts of

File Structures. The key topic it makes use of is Indexing. The data is stored in the files that is structured by means of indexing the entries. It allows for easy and convenient access to the records for retrieval and modification purposes.

The main two data to manage in this project are Students information and Educators information. This data is subjected to grow as new batches are started. Thus it becomes necessary to find a way to store these huge files. This project aims to do exactly that and provides an efficient way for storing the files.

**S****COPE**

There is a growing demand and market for the coaching institutes in the developing country like India. This can attributed to the importance placed on education as it is the tool of the intellectual.

The Indian Coaching industry is a 400 crore industry growing at Compound Annual Growth Rate of 34%. Indian literacy rate has risen from 65% to 73% in the last five years. However the brightest minds in India keeps rising and nourished with the help of coaching institutes. The statistics reveals an average Indian middle-class family spends 1/3rd of their income on their child’s education. With this importance, coaching centres have a lot of scope.

With the rise in online tutoring as well as increased strength of a coaching centre, it becomes necessary for it to keep track of their data. Many of these private coaching centres are either medium scaled or lack the modern technological infrastructure and instead rely on traditional way. With this project we can help them digitalize their records and store their data in an efficient manner so they can carry out their coaching in a much convenient manner and increase productivity

**KEY CONCEPTS**

**File Structures**

**Shape, square

Description automatically generated**

File Structure is a combination of representations for data in files and operation to access the data. File Structure is the organization of data in secondary storage structures such as disks. A successful file structure organizes your data and code with the goal of repeatability, making it easier for you and your collaborators to revisit, revise and develop your project. File structures are not fixed entities, but rather build a framework that communicates the function and purpose of elements within a project by separating concerns into a hierarchy of folders and using consistent, chronological, and descriptive names. An improvement in file structure design may make an application hundred times faster. The details of the representations of the data and the implementation of operations determine the efficiency of the file structure.

**Indexing**

**Shape, square

Description automatically generated**

File Indexing is a technique to efficiently retrieve records from the stored files based on some attributes on which the indexing has been done. There are several types of single level ordered indexes.

Primary index is an index specified on the ordering key field of a sorted file of records. Clustering index is an index built on the records that are sorted on the non-key field. It is since different records have different values in the key field, but for a non-key field, some records may share the same value.

Secondary index can be specified on any non-ordering field of a file. A file can have several secondary indexes in addition to its primary access path. These indexes do not affect the physical organisation of records.

**Multi-level Indexing**

**Shape, square

Description automatically generated**

The idea behind a multilevel index is to reduce the part of the index that we have to continue to search by blocking factor for the index, which is almost always larger than 2. Thus, the search space can be reduced much faster. Searching a multilevel index requires block accesses, which is a smaller number than for a binary search if the blocking factor of the index is bigger than 2.

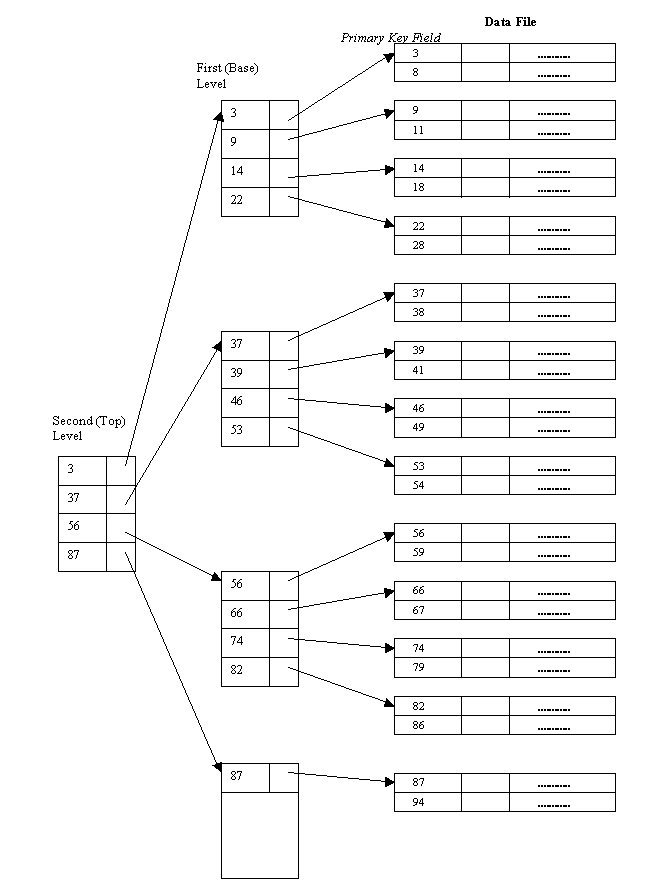
A multi-level index considers the index file, which we referred as a single-level ordered index or it’s referred as the first level (or) base level of the multi-level structure, as a sorted file with a distinct value for each primary key field. We can build a primary index for an index file itself (on top of the index at the first level). This new index to the first level is called the second level of the multilevel index. Because the second level is a primary index, we can use block anchors so that the second level has one entry for each block of the first-level index entries.

The above process can be repeated and a third level index can be created on top of the second level one. The third level, which is a primary index for the second level, has an entry for each second level block.

**Shape, square

Description automatically generated**The multilevel structure can be used on any type of index as long as the first-level index has distinct values for primary key field and fixed-length entries. The figure below depicts a multilevel index built on top of a primary index

**Shape, square

Description automatically generated**

**PROGRAM IMPLEMENTATION**

**Project Structure**

**Shape, square

Description automatically generated**

The coaching institute will have two disparate sets of data to keep track and store. Those would be Student and Teachers

**Student**

**Teacher**

|  |  |
| --- | --- |
| First Name | First Name |
| Last Name | Last Name |
| Registration ID | Teacher ID |
| Age | Qualification |
| Fee Status | Pay |
| Phone Number | Phone Number |
| Batch | Subject |
| Student Index | Teacher index |

**H****eader Files**

**Shape, square

Description automatically generated**

**#include <iostream>**

iostream is the standard input output stream. It contains definitions of objects like cout to insert data into a stream (output), cin to extract data from a stream (input) and to control the format of data

**#include <fstream>**

It mainly describes the file stream. This header file is used to handle the data being read from a file as input or data being written into the file as output.

**#include <string.h>**

The string. h header defines one variable type, one macro, and various functions for manipulating arrays of characters

**#include <conio.h>**

It is a header file that includes inbuilt functions like getch() and clrscr(). It stand for console input ouput. it takes input from keyboard and displays it on screen

**#include <cstdlib>**

It stands for C Standard General Utilities Library. This defines several general purpose functions, including dynamic memory management, searching, sorting and converting to facilitate efficient, high-performing, standardized C++ code across platforms

**INDEXING in the CODE**

**Shape, square

Description automatically generated**Primary Indexing is used in this project. We declare the INDEX as a Structure. The struct members “instudent” and “inteacher” are used throughout the program and functions to access, sort and modify the index file of student and teacher

Text

Description automatically generated

**Sort\_Index()**

**Shape, square

Description automatically generated**

Indexing is the process of associating a key with the location of a data record. An external sort uses the concept of a key sort, in which an index file is created which consist of key and pointer pairs.

The index file is sorted or organized using a tree structure, thereby imposing a logical order on the records without physically rearranging them.

Text

Description automatically generated

Text

Description automatically generated

**Retrieve\_Record()**

**Shape, square

Description automatically generated**

This facilitates the client who want to get every details of any particular number of student or teacher. They can specify their request and this function runs through the index and returns the data with very fast access.

Text

Description automatically generated

Chart

Description automatically generated with low confidence

**Search\_Index()**

**Shape, square

Description automatically generated**

This function allows to look for the primary index, whether it exists or not and return the corresponding record if it exists. Registraion ID for students and Teacher ID for teachers is the primary index

Text

Description automatically generated

Text

Description automatically generated

**Delete()**

**Shape, square

Description automatically generated**

This function is used to delete the record that is no longer needed. It checks if the inputted key exists or not. If it does, then that record is deleted from the file permanently. The other records and their indices are sorted then.

Text

Description automatically generated

**Shape, square

Description automatically generated**

Text

Description automatically generated

**Shape, square

Description automatically generated**

**Update\_Status()**

**Shape, square

Description automatically generated**

This is used to update the Fee status of a student. Oftentimes, students who opt in for coaching pay partial fees at the beginning and later pay the full fee. We maintain fee status as whether paid or unpaid. When the student pays the fees, we update it. The details of students who haven’t paid the fees can be retrieved which is helpful.

Text

Description automatically generated

**Main()**

**Shape, square

Description automatically generated**

The brief description of main() function implementation are as follows

* Local Declaration of Record Fields
* Creation of record and index file for students
* Structuring record of student files
* Key sorting the student index
* Creation of record and index file for teachers
* Structuring record of teacher files
* Key sorting the teacher index
* Get input whether client wants to access Students section (or) Teachers section
* Use the WHILE loop for getting data on repeat and exit out of the loop when user specifies the exit option
* Use the Switch case to jump to either Students section or Teachers section
* The same idea is repeated inside of the both sections
* Student Information
  1. New Admission: Get details of new students admitted and insert it into the file
  2. Find Student: Get the ID of student and check whether it exists or not
  3. Check Fee Status: Display the details of students who have paid or not paid the fee
  4. Remove: Delete the student record of specified ID from the file
  5. Display: Display the details of all the students or display the data batchwise
  6. Update Fee Status: Modify the fee status as paid or unpaid for a student
  7. Exit
* Teachers Information
  1. Create New Entry: Get details of the new teachers and insert it into the file
  2. Find Teacher: Search for a teacher using the key teacher ID
  3. Remove: Delete the details of the teacher that no longer works in the institute
  4. Display: Display the particulars of all the teachers or batchwise display the details
  5. Students Assigned: See the students assigned for a particular teacher to coach
  6. Exit

**Shape, square

Description automatically generated**

**Shape, square

Description automatically generatedPROGRAM OUTPUT**

A screenshot of a computer

Description automatically generated with medium confidence

**Text

Description automatically generated**

A screenshot of a computer

Description automatically generated with medium confidence

**Shape, square

Description automatically generated**

**Text

Description automatically generated**

**Shape, square

Description automatically generated**

A screenshot of a computer screen

Description automatically generated

Brief



**Shape, square

Description automatically generated**

A screenshot of a computer screen

Description automatically generated

**Main()**

**Shape, square

Description automatically generated**

A screenshot of a computer screen

Description automatically generated

**Text

Description automatically generated**

**Shape, square

Description automatically generated**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, chat or text message

Description automatically generated

**Graphical user interface, text, application

Description automatically generated**

**Shape, square

Description automatically generated**

**CONCLUSION**

The Coaching Institute Management System has been made to aid the storage of details of students and teachers in the coaching centre. The records are obtained and efficiently stored in the file for easier and quick access whenever needed. Indexing of records is employed in this project that keeps index of corresponding records. This takes lesser storage and provides faster access during data retrieval. This project makes the file storage of the institute convenient.

**Shape, square

Description automatically generated**

**REFERENCES**

**Shape, square

Description automatically generated**

1. An Object Oriented Approach with C++ by Michael J. Folk, Bill Zoellick, Greg Riccardi
2. File Structures Using C++ by K.R. Venugopal, K.G. Srinivas, P.M. Krishnaraj
3. Indexing and File Organization Course by University of Cape Town (<https://www.cs.uct.ac.za>)
4. File Structures by GeeksforGeeks.org
5. Working with C++ Files by w3schools.com