Project Report

Submitted in partial fulfillment of the degree of

B.Tech CSE

By

Abhinay Giri(11900122030)
Akash Singh(11900122038)
Aviraj Roy(11900122047)

Second Year Students of

Siliguri Institute of Technology

Under the supervision of

Mr. Vishal Ray

Sikharthy Infotech Pvt. Ltd.

Department of Computer Science Engineering

Date: 15th Dec, 2023

We hereby forward the documentation prepared by us **Abhinay Giri, Akash Singh,** and **Aviraj Roy,** under the supervision of **Mr. Vishal Sir** entitled **Library Management System,** accepted as fulfilment of the requirement for the Degree of Bachelor of Technology (BTech) in **Computer Science Engineering** from **Siliguri Institute of Technology** affiliated to **Maulana Abul Kalam Azad University of Technology** (**MAKAUT**).

_____ _____

Mr. Vishal Ray
(Instructor & Trainer)

Project Guide

Sikharthy Infotech Pvt. Ltd.

Abhinay Giri
Akash Singh
Aviraj Roy

Department of Computer Science Engineering
Siliguri Institute of Technology

Library Management System

By

Abhinay Giri(11900122030)
Akash Singh(11900122038)
Aviraj Roy(11900122047)

UNDER THE GUIDANCE OF

Mr. Vishal Ray

Project Guide

Sikharthy Infotech Pvt. Ltd.

THE REPORT IS SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

B.TECH
IN
COMPUTER SCIENCE ENGINEERING

SILIGURI INSTITUTE OF TECHNOLOGY

AFFILIATED TO
Maulana Abul Kalam Azad University of Technology

Address: S.I.T Campus, Salbari, P.O. Sukna, Dt. Darjeeling, West Bengal - 734009, India

Phone: 0353-2778002 / 2778004

Email: register@sittechno.org, admission@sittechno.org

Website: https://sittechno.org/

Certificate of Approval

The foregoing project is hereby approved as a creditable study for the BTech in Computer Science Engineering and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval, the undersigned do not necessarily endorse or approve any statement made, opinion expressed, or conclusion therein, but approve this project only for the purpose for which it is submitted.

Final Examination for
Evaluation of the Project

Signatures of Examiners

ABSTRACT

The purpose of the project entitled "Library **Management System**" is to develop an Intelligent Library Management System revolutionizes library operations with user-friendly interfaces, personalized recommendations, multi-channel access, and seamless Learning Management System integration. Enhancing efficiency and user experience, ILMS modernizes the library landscape in the digital era. The main function of the system is to provide a seamless data management system for books.

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and participation of a large number of individuals in this attempt. Our project report has been structured under the valued suggestions, support, and guidance of **Mr. Vishal Ray**. Under his guidance, we have accomplished the challenging task in a very short time.

Finally, we express our sincere thankfulness to our family members for inspiring us all throughout and always encouraging us.

Abhinay Giri
Akash Singh
Aviraj Roy

Department of Computer Science Engineering

TABLE OF CONTENTS

Chapter 1: Introduction
1: Introduction

Chapter 2: What We Used
2.1: JDK-21
2.2: VS code

Chapter 3: Basics
3.1: Java variable
3.2: Java Data types
3.3: Java Loops
3.4: Java Array
3.5: Java String

Chapter 4: Java Methods
4.1: Java Method Parameters
4.2: Java Method Overloading

Chapter 5: Java Classes
5.1: Java OOPs
5.2: Java Classes/Objects
5.3: Java Classes Attributes
5.4: Java Constructors
5.5: Java Encapsulation
5.6: Java Polymorphism
5.7: Java Abstractions
5.8: Java Inheritance

Chapter 6: Conclusion

Chapter 7: References

INTRODUCTION

The Library Management System is a dedicated website designed for efficient book data management within libraries. Tailored to empower library owners, it provides seamless access to comprehensive information, streamlining operations and enhancing overall management. This digital solution ensures a user-friendly platform for comprehensive and effortless library administration.

2.1 JDK 21

Java Development Kit (JDK) is a robust software development kit essential for Java programming. It includes tools, libraries, and the Java Runtime Environment, facilitating the creation of powerful Java applications.

2.2 VS code

Visual Studio Code (VSCode) for Java is a lightweight, cross-platform code editor with rich features for Java development. It offers intelligent code completion, debugging support, and seamless integration with popular build tools. Its extensibility through extensions enhances productivity, making it a preferred choice for Java developers in diverse coding environments.

We used Microsoft Visual Studio Code (VS Code)

3. Chapter 3: Basics

Java is a versatile, object-oriented programming language known for its platform independence. It uses straightforward syntax, making it accessible for beginners. Key features include automatic memory management, strong type-checking, and a vast standard library. Programs are executed in a Java Virtual Machine (JVM), allowing Java applications to run on various devices. Java's portability and reliability make it widely used in web, mobile, and enterprise applications.

3.1JAVA Variables

Variables are containers for storing data values.

In Java, there are different types of variables, for example:

- String - stores text, such as "Hello". String values are surrounded by double quotes
- int - stores integers (whole numbers), without decimals, such as 123 or -123
- float - stores floating point numbers, with decimals, such as 19.99 or -19.99
- char - stores single characters, such as 'a' or 'B'. Char values are surrounded by single quotes
- boolean - stores values with two states: true or false

3.2 Java Data types

Data types are divided into two groups:

- Primitive data types - include byte, short, int, long, float, double, boolean, and char
- Non-primitive data types - such as String, Arrays, and Classes

3.3 JAVA loop

You can loop through the array elements with the for loop, and use the length property to specify how many times the loop should run.

The following example outputs all elements in the car array:

Example:- Get your own Java Server

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
for (int i = 0; i < cars.length; i++) {
```

```
  System.out.println(cars[i]);
}
```

3.4: Java Array

Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.

To declare an array, define the variable type with square brackets:

String[] cars;
We have now declared a variable that holds an array of strings. To insert values into it, you can place the values in a comma-separated list, inside curly braces:

String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
To create an array of integers, you could write:

int[] myNum = {10, 20, 30, 40};
Access the Elements of an Array
You can access an array element by referring to the index number.

3.5: Java String

Strings are used for storing text.

A String variable contains a collection of characters surrounded by double quotes:

ExampleGet your own Java Server

Create a variable of type String and assign it a value:

String greeting = "Hello";

String Length

A String in Java is an object, that contains methods that can perform certain operations on strings. For example, the length of a string can be found with the length() method:

Example

String txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

System. out.println("The length of the txt string is: " + txt.length());

4. Java Methods

4.1 Java Method Parameters

- Information can be passed to methods as parameters. Parameters act as variables inside the method.
- Parameters are specified after the method name, inside the parentheses. You can add as many parameters as you want, just separate them with a comma.
- The following example has a method that takes a String called fname as a parameter. When the method is called, we pass along a first name, which is used inside the method to print the full name:

4.2 Java Method Overloading

- With method overloading, multiple methods can have the same name with different parameters:

```
static int plusMethodInt(int x, int y) {
  return x + y;
}

static double plusMethodDouble(double x, double y) {
  return x + y;
}

public static void main(String[] args) {
  int myNum1 = plusMethodInt(8, 5);
```

```
    double myNum2 = plusMethodDouble(4.3, 6.26);
    System.out.println("int: " + myNum1);
    System.out.println("double: " + myNum2);

}
```

5. Java Classes

In Java, a class is a blueprint for creating objects. It encapsulates data and behavior, providing a template for object instantiation. Class members include fields (variables) and methods (functions), enabling structured and modular programming.

5.1 Java OOPs

OOPS stands for Object-Oriented Programming Structure.

Procedural programming is about writing procedures or methods that perform operations on the data, while object-oriented programming is about creating objects that contain both data and methods.

Object-oriented programming has several advantages over procedural programming:

- OOPS is faster and easier to execute
- OOPS provides a clear structure for the programs
- OOPS helps to keep the Java code DRY "Don't Repeat Yourself", and makes the code easier to maintain, modify and debug
- OOPS makes it possible to create full reusable applications with less code and shorter development time

5.2 Java Classes/Objects

Java is an object-oriented programming language.

Everything in Java is associated with classes and objects, along with its attributes and methods. For example: in real life, a car is an object. The car has attributes, such as weight and color, and methods, such as drive and brake.

A Class is like an object constructor, or a "blueprint" for creating objects.

5.3: Java Classes Attributes

we used the term "variable" for x in the example (as shown below). It is actually an attribute of the class. Or you could say that class attributes are variables within a class:

ExampleGet your own Java Server
Create a class called "Main" with two attributes: x and y:

```
public class Main {
  int x = 5;
  int y = 3;
}
```

Another term for class attributes is fields.

Accessing Attributes
You can access attributes by creating an object of the class, and by using the dot syntax (.):

The following example will create an object of the Main class, with the name myObj. We use the x attribute on the object to print its value:

Example:-
Create an object called "myObj" and print the value of x:

```
public class Main {
```

```
  int x = 5;

  public static void main(String[] args) {
    Main myObj = new Main();
    System.out.println(myObj.x);
  }
}
```

Modify Attributes
You can also modify attribute values:

Example:-
Set the value of x to 40:

```
public class Main {
  int x;

  public static void main(String[] args) {
    Main myObj = new Main();
    myObj.x = 40;
    System.out.println(myObj.x);
  }
}
```

5.4 Java Constructors

A constructor in Java is a special method that is used to initialize objects. The constructor is called when an object of a class is created. It can be used to set initial values for object attributes:

```
      // Create a Main class

public class Main {

  int x;  // Create a class attribute
```

```
  // Create a class constructor for the Main class
  public Main() {
    x = 5;  // Set the initial value for the class attribute x
  }
  public static void main(String[] args) {
    Main myObj = new Main(); // Create an object of class Main (This
will call the constructor)
    System.out.println(myObj.x); // Print the value of x
  }
}
```

5.5Java Encapsulation

The meaning of Encapsulation is to make sure that "sensitive" data is hidden from users. To achieve this, you must:

- declare class variables/attributes as private
- provide public get and set methods to access and update the value of a private variable

5.6 Java Polymorphism

Polymorphism means "many forms", and it occurs when we have many classes that are related to each other by inheritance.

Like we specified in the previous chapter; Inheritance lets us inherit attributes and methods from another class. Polymorphism uses those methods to perform different tasks. This allows us to perform a single action in different ways.

For example, think of a superclass called Animal that has a method called animalSound(). Subclasses of Animals could be Pigs, Cats, Dogs, Birds - And they also have their own implementation of an animal sound (the pig oinks, and the cat meows, etc.):

5.7 Java Abstractions

Data abstraction is the process of hiding certain details and showing only essential information to the user.
Abstraction can be achieved with either abstract classes or interfaces (which you will learn more about in the next chapter).

The abstract keyword is a non-access modifier, used for classes and methods:

- Abstract class: is a restricted class that cannot be used to create objects (to access it, it must be inherited from another class).
- Abstract method: can only be used in an abstract class, and it does not have a body. The body is provided by the subclass (inherited from).

An abstract class can have both abstract and regular methods:

5.8 Java Inheritance

In Java, it is possible to inherit attributes and methods from one class to another.
We group the "inheritance concept" into two categories:

- subclass (child) - the class that inherits from another class
- superclass (parent) - the class being inherited from

To inherit from a class, use the extends keyword.

In the example below, the Car class (subclass) inherits the attributes and methods from the Vehicle class (superclass):

```
     class Vehicle {
  protected String brand = "Ford";         // Vehicle attribute
  public void honk() {                     // Vehicle method
    System.out.println("Tuut, tuut!");
  }
}
class Car extends Vehicle {
  private String modelName = "Mustang";    // Car attribute
  public static void main(String[] args) {
    // Create a myCar object
    Car myCar = new Car();
    // Call the honk() method (from the Vehicle class) on the myCar
object
    myCar.honk();
   // Display the value of the brand attribute (from the Vehicle
class) and the value of the modelName from the Car class
    System.out.println(myCar.brand + " " + myCar.modelName);
  }
}
```

6. Conclusion

The project **“” Library Management System”** is to develop an Intelligent Library Management System that revolutionizes library operations with user-friendly interfaces, personalized recommendations, multi-channel access, and seamless Learning Management System integration. Enhancing efficiency and user experience, ILMS modernizes the library landscape in the digital era. The main function of the system is to provide a seamless data management system for books.

12.REFERENCES

1. https://www.w3schools.com/
2. https://www.codewithharry.com/
3. https://www.freecodecamp.org/