

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belagavi – 590 018



Advanced Java Laboratory with Mini project (20CSEL57)

Report on

“Demonstration of Indian Knowledge System in Astronomy and Mathematics”

Submitted By

N.Dharshan

1GA20CS086

Nishant Kumar Thakur

1GA20CS089

Under the Guidance of

Mrs. Vanishree M L

Assistant Professor
Dept. of CSE



Department of Computer Science and Engineering
(Accredited by NBA 2022-2025)

GLOBAL ACADEMY OF TECHNOLOGY

Rajarajeshwarinagar, Bengaluru – 560 098

2022 - 2023





GLOBAL ACADEMY OF TECHNOLOGY

An Autonomous Institute, Affiliated to VTU Belagavi,
Approved by AICTE, Accredited by NAAC, 'A' Grade,
Ideal Homes Township, Rajarajeshwari Nagar, Bengaluru – 560 098



Department of Computer Science and Engineering

(Accredited by NBA 2022-2025)

CERTIFICATE

Certified that the V Semester Advanced Java Laboratory with Mini project entitled “**Demonstration of Indian Knowledge Systems in Astronomy and Mathematics**” carried out by **N.Dharshan [1GA20CS086]**, **Nishant Kumar Thakur [USN]** are bonafide students of Global Academy of Technology, Bachelor of Engineering in Computer Science and Engineering from Visvesvaraya Technological University, Belagavi during the year 2022-2023. It is certified that all the corrections/suggestions indicated have been incorporated in the report submitted.

Dr. K Anantha Padmanabha
Professor
Dept. of CSE
GAT, Bengaluru

Mrs. Vanishree M.L
Assistant Professor
Dept. of CSE
GAT, Bengaluru

Dr. Kumaraswamy S
Professor and HOD
Dept. of CSE
GAT, Bengaluru

Name and Signature of the Students

Name: N.Dharshan

Name: Nishant Kumar Thakur

Signature of Internal Examiner

Signature of External Examiner

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant encouragement and guidance crowned our efforts with success.

We consider ourselves proud, to be part of **Global Academy of Technology** family, the Institution which stood by our way in endeavors.

We express our deep and sincere thanks to our Principal **Dr. N. Rana Pratap Reddy** for his support.

We are grateful to **Dr. Kumaraswamy S**, Professor and HOD, Dept. of CSE for his inspiration and invaluable help in channelizing our efforts in right direction.

We wish to thank our internal guides **Dr. K Anantha Padmanabha**, Professor and **Mrs. Vanishree M L**, Assistant Professor, Dept. of CSE for guiding and correcting various documents of ours with attention and care. They have taken initiative to go through the document and make necessary corrections as and when needed.

We would like to thank the faculty members and supporting staff of the Department of CSE, GAT for providing all the support for completing the Project work.

Finally, we are grateful to our parents and friends for their unconditional support and help during our Project work.

Student Names

N.Dharshan [1GA20CS086]

Nishant Kumar Thakur [1GA20CS089]

ABSTRACT

The study of subjects like Mathematics is inalienable in the development of any civilization and we find copious references to Greek roots in modern mathematics, though similar contributions that merit in mention relatively fewer in number and find themselves a rather obscure even among students of the same in India, the cradle of many key contributions to Science and Mathematics.

On this note we attempt to study some key algorithms and their working through an implementation in Java with a visually lucid Swing UI implementation. We have based our work on some selected verses of classical authors including and not limiting to Lilavati, Sadratnamala, etc. We use JDBC, the persistence system offered by Java to store and readily access data in the same.

Our work tries to shed some light on the computational soundness of many algorithms by practical implementation and to encourage mathematical studies on the same in future.

TABLE OF CONTENTS

| SL. NO | DESCRIPTION | PAGE NO. |
|--------|--|-----------|
| 1 | ABSTRACT | |
| 2 | TABLE OF CONTENTS | |
| 3 | LIST OF FIGURES | |
| 4 | PROJECT REPORT OUTLINE | 1 |
| 5 | CHAPTER 1 INTRODUCTION | 2 |
| | 1.1 Introduction to Advanced Java Concepts | 2 |
| 6 | CHAPTER 2 SYSTEM DEFINITION | 5 |
| | 2.1 Problem Statement and Objectives | 5 |
| | 2.2 System Requirements Specification | 5 |
| 7 | CHAPTER 3 IMPLEMENTATION | 6 |
| | 3.1 Program Code | 6 |
| 8 | CHAPTER 4 TESTING AND RESULTS | 17 |
| | 4.1 Testing | 17 |
| | 4.2 Snapshots | 18 |
| 9 | CONCLUSION | 20 |
| 10 | BIBLIOGRAPHY | 21 |

List of Figures

| Figure No. | Title | Page No. |
|------------|-----------------|----------|
| Fig 4.2.1 | About | 18 |
| Fig 4.2.2 | View | 18 |
| Fig 4.2.3 | Background Info | 19 |
| Fig 4.2.4 | Working | 19 |

PROJECT REPORT OUTLINE

1 CHAPTER 1:

A brief description of Java highlighting its features.

2 CHAPTER 2:

System Requirements Specifications:

The Description of required Hardware and software for running the project application.

3 CHAPTER 3:

The Program Code.

4 CHAPTER 4:

Snapshots displaying outputs in each input case.

CHAPTER 1

INTRODUCTION

1.1 Introduction to Advanced Java Concepts

1.1.1 Collections

The Collection in Java is a framework that provides an architecture to store and manipulate the group of objects.

Java Collections can achieve all the operations that you perform on a data such as searching, sorting, insertion, manipulation, and deletion.

Java Collection means a single unit of objects. Java Collection framework provides many interfaces (Set, List, Queue, Deque) and classes (ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet).

Before the Collection Framework (or before JDK 1.2) was introduced, the standard methods for grouping Java objects (or collections) were Arrays or Vectors, or Hashtables. All these collections had no common interface. Therefore, though the main aim of all the collections is the same, the implementation of all these collections was defined independently and had no correlation among them. And also, it is very difficult for the users to remember all the different methods, syntax, and constructors present in every collection class.

1.1.2 Servlets

- Servlets are small programs that execute on the server side of a web connection.
- Applets dynamically extend the functionality of a web browser whereas servlets dynamically extend the functionality of a web server.
- Servlets are the Java programs that run on the Java-enabled web server or application server.
- They are used to handle the request obtained from the web server, process the request, produce the response, and then send response back to the web server.

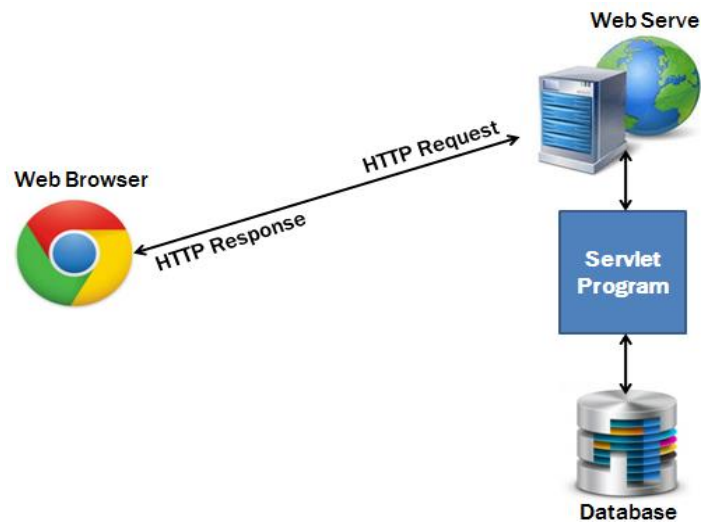


Fig 1.1 Working of Servlet

Execution of Servlet

- The clients send the request to the web server.
- The web server receives the request.
- The web server passes the request to the corresponding servlet.
- The servlet processes the request and generates the response in the form of output.
- The servlet sends the response back to the web server.
- The web server sends the response back to the client and the client browser displays it on the screen.

1.1.3 JSP

Java Server Pages (JSP) technology provides a simplified, fast way to create dynamic web content. Java Server Pages (JSP) technology enables Web developers and designers to rapidly develop and easily maintain, information, dynamic Web pages that leverage existing business systems. JSP technology uses HTML/XML-like tags that encapsulate the logic that generates the content for the page. A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tags, etc.

1.1.4 JDBC

JDBC or Java Database Connectivity is a Java API to connect and execute the query with the database. It is a specification from Sun microsystems that provides a standard abstraction (API or Protocol) for java applications to communicate with various databases. It provides the language with java database connectivity standards. It is used to write programs required to access databases. JDBC, along with the database driver, can access databases and spreadsheets. The enterprise data stored in a relational database can be accessed with the help of JDBC APIs.

JDBC is an API (Application programming interface) used in java programming to interact with databases. The classes and interfaces of JDBC allow the application to send requests made by users to the specified database.

CHAPTER 2

SYSTEM DEFINITION

2.1 PROBLEM STATEMENT AND OBJECTIVES

- Demonstrate the working of some of the Classical Algorithms through a Java Implementation
- Madhavan's Sine and Cose
- Sum of Squares
- Sum of Cubes
- Use a good UI through Swing

2.2 SYSTEM REQUIREMENTS SPECIFICATION

SOFTWARE REQUIREMENTS:

- Operating system :
- IDE :
- Programming Language :

HARDWARE REQUIREMENTS:

- Processor : Intel Core i5 or later versions
- Memory : 512 MB RAM
- 2GB Hard Disk Drive
- Mouse or other pointing device
- Keyboard
- Display device

MISCELLANEOUS REQUIREMENTS:

- All the required library files and the header files should be available in the include directory.

CHAPTER 3

IMPLEMENTATION

3.1 PROGRAM CODE

```
import javax.swing.*;
import javax.swing.event.*;
import javax.swing.table.DefaultTableCellRenderer;
import java.util.*;
import java.awt.*;
import java.awt.Color;
import java.awt.Font;
import java.io.UnsupportedEncodingException;
import java.awt.event.*;

class MyFrame extends JLabel {
    MyFrame(String str) {
        super(str);
        // this.
    }
}

public class Main extends DefaultTableCellRenderer implements ActionListener {

    static JFrame frame = new JFrame("Hayavadana");
    static Icon image;
    static JButton squareSubmit, cubeSubmit;
```

```
static JLabel nSqLabel, aSqLabel, dSqLabel, nCbLabel;
static JLabel sqHeader, cbHeader, sqRes, cbRes;
static JTextField nSqTField, aSqTField, dSqTField, nCbTField;
// static JTextArea dataItemText;
// static JPanel myDisplay = new JPanel();

public static void addNewFrame1() {

    JFrame squareAndCube = new JFrame("squareAndCube");
    // JButton squareSubmit, cubeSubmit;
    // JLabel nSqLabel, aSqLabel, dSqLabel, nCbLabel;
    // JLabel sqHeader, cbHeader;
    // JTextField nSqTField, aSqTField, dSqTField, nCbTField;

    squareAndCube.getContentPane().setBackground( Color.decode("#FFF9EE") );
    squareAndCube.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    squareAndCube.setSize(1233, 733);
    squareAndCube.setLayout(null);

    // for square
    sqHeader = new JLabel("Sum of Squares of n Terms:");
    sqHeader.setBounds(82, 150, 360, 35);
    sqHeader.setForeground(Color.decode("#075466"));
    sqHeader.setFont(new Font("Segoe UI Semibold", Font.BOLD , 21));
    JSeparator js = new JSeparator(HORIZONTAL);

    nSqLabel = new JLabel("Number of Terms:");
    nSqLabel.setBounds(89, 250, 160, 22);
    nSqLabel.setForeground(Color.decode("#075466"));
    nSqLabel.setFont(new Font("Segoe UI Semibold", Font.BOLD , 18));

    dSqLabel = new JLabel("Common Diff:");
    dSqLabel.setBounds(89, 292, 160, 22);
```

```
dSqLabel.setForeground(Color.decode("#075466"));
dSqLabel.setFont(new Font("Segoe UI Semibold", Font.BOLD , 18));
```

```
aSqLabel = new JLabel("First term:");
aSqLabel.setBounds(89, 334, 160, 22);
aSqLabel.setForeground(Color.decode("#075466"));
aSqLabel.setFont(new Font("Segoe UI Semibold", Font.BOLD , 18));
```

```
nSqTField = new JTextField();
nSqTField.setBounds(289, 250, 118, 35);
dSqTField = new JTextField();
dSqTField.setBounds(289, 292, 118, 35);
aSqTField = new JTextField();
aSqTField.setBounds(289, 334, 118, 35);
```

```
squareSubmit = new JButton("Submit");
squareSubmit.setBounds(500, 292, 118, 35);
squareSubmit.addActionListener(new Main());
```

```
sqRes = new JLabel();
sqRes.setBounds(625, 292, 75, 30);
```

```
squareAndCube.add(sqHeader);
squareAndCube.add(nSqLabel);
squareAndCube.add(dSqLabel);
squareAndCube.add(aSqLabel);
squareAndCube.add(js);
squareAndCube.add(nSqTField);
squareAndCube.add(dSqTField);
squareAndCube.add(aSqTField);
squareAndCube.add(squareSubmit);
squareAndCube.add(sqRes);
```

```
// For cube
cbHeader = new JLabel("Sum of Cubes:\n");
cbHeader.setBounds(90, 400, 360, 35);
cbHeader.setForeground(Color.decode("#075466"));
cbHeader.setFont(new Font("Segoe UI Semibold", Font.BOLD , 21));

nCbLabel = new JLabel("Number of Terms:\n");
nCbLabel.setBounds(90, 440, 360, 35);
nCbLabel.setForeground(Color.decode("#075466"));
nCbLabel.setFont(new Font("Segoe UI Semibold", Font.BOLD , 18));

nCbTField = new JTextField();
nCbTField.setBounds(289, 440, 118, 35);
cubeSubmit = new JButton("Submit");
cubeSubmit.setBounds(500, 440, 118, 35);
cubeSubmit.addActionListener(new Main());
cbRes = new JLabel();
cbRes.setBounds(625, 440, 75, 30);

squareAndCube.add(nCbLabel);
squareAndCube.add(cbHeader);
squareAndCube.add(nCbTField);
squareAndCube.add(cubeSubmit);
squareAndCube.add(cbRes);
squareAndCube.setVisible(true);
}

public static void addButtons(JPanel p, int x, int y) {
    // int width, height;
    for(int i = x; i <= 20; i++) {
```



```
        JButton b = new JButton("Kamala Srinivas");
        b.setBounds(x, y + i*20 + 5, 48, 20);
        p.add(b);
    }
    // JScrollPane sp =new JScrollPane(p);
    p.setVisible(true);

}

public static void addTable(JPanel view) {
    Image i;
    String data[][]={
        {"001","Madhavan's Sine, Cosine"},
        {"002","Sum of Square"},
        {"003","Sum of Cubes"}
    };
    String column[]={ "ID","NAME"};
    final JTable jt=new JTable(data,column);
    HashMap<String, Integer> hm = new HashMap<String, Integer>();
    hm.put("Madhavan's Sine, Cosine", 1);
    hm.put("Sum of Square", 2);
    hm.put("Sum of Cubes", 3);

    jt.setFont(new Font("Segoe UI", Font.BOLD , 18));
    // JScrollPane scrollPane = new JScrollPane(jt);
    jt.setSelectionEnabled(true);
    ListSelectionModel select= jt.getSelectionModel();
    select.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);

    jt.setRowHeight(24);

    select.addListSelectionListener(new ListSelectionListener() {
        public void valueChanged(ListSelectionEvent e) {
```

```
String Data = null;
int[] row = jt.getSelectedRows();
int[] columns = jt.getSelectedColumns();
//      for (int i = 0; i < row.length; i++) {
//      for (int j = 0; j < columns.length; j++) {
//      Data = (String) jt.getValueAt(row[i], columns[j]);
//      } }

Data = (String) jt.getValueAt(row[row.length - 1], columns[columns.length - 1]);

myfun(Data, hm.get(Data));
}

});
JScrollPane sp=new JScrollPane(jt);
// TODO: Add table
view.add(sp);
view.setVisible(true);
}

private static Object String(java.lang.String string) {
    return null;
}

public static void myfun(String Data, int key) {
    JLabel dataItemText = new JLabel(Data);
    dataItemText.setBackground(Color.decode("#B7D6DD"));
    // myDisplay.setBackground(Color.decode("#B7D6DD"));
    dataItemText.setBounds(767, 150, 60, 20);
    dataItemText.setFont(new Font("Segoe UI", Font.BOLD , 18));
    // myDisplay.add(dataItemText);
    // frame.add(dataItemText);
    JFrame f = new JFrame();
```

```
image = new ImageIcon("D:\\College\\Java\\images\\" + key + ".png");
JLabel jl = new JLabel(image);
System.out.println("Table element selected is: " + Data);
f.setBounds(767, 150, 349, 449);
f.add(jl);
f.setVisible(true);
switch(key) {
    case 1:
        addNewFrame1();
        break;
    case 2:
        addNewFrame1();
        break;
    default:
        addNewFrame1();
        break;
}
}

public static void main(String[] args) {
    System.out.println("Srimathe Ramanujaya Nama:");
    sandbox();
}

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == squareSubmit) {
        System.out.println("Kamala Srinivas");

        int n = Integer.valueOf(nSqTField.getText());
        int a = Integer.valueOf(aSqTField.getText());
        int d = Integer.valueOf(dSqTField.getText());
        float m = (float)((((2*n)-1)*Math.pow(d,2)))/6;
        m += a*d;
```

```
        m*=(n-1);
        m+=Math.pow(a, 2);
        m*=n;
        float sum=m;
        System.out.println(sum);
        String s = String.valueOf(sum);
        sqRes.setText(s);
    }

    else {
        int n = Integer.valueOf(nCbTField.getText());
        float sum=(n*(n+1))/2;
        sum=sum*sum;
        System.out.println(sum);
        String s = String.valueOf(sum);
        cbRes.setText(s);
    }
}

public static void sandbox() {

    frame.getContentPane().setBackground( Color.decode("#FFF9EE") );
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(1233, 733);
    frame.setLayout(null);

    JLabel title;
    title = new JLabel("Lilavati ");
    title.setBounds(112,66, 195,50);
    title.setForeground(Color.decode("#075466"));
    title.setFont(new Font("Segoe UI Semibold", Font.BOLD , 40));
    frame.add(title);
```

```
title = new JLabel("Hayavadana");
title.setBounds(986,86, 175,50);
title.setForeground(Color.decode("#D90166"));
title.setFont(new Font("Segoe UI Semibold", Font.BOLD , 26));
frame.add(title);
```

```
// tabbed pane
```

```
JTabbedPane tp=new JTabbedPane();
tp.setBounds(52,120,592,515);
tp.setFont(new Font("Segoe", Font.CENTER_BASELINE, 20));
tp.setBackground(Color.decode("#FFF9EE"));
```

```
// individual panes
```

```
// About
```

```
JPanel about =new JPanel();
about.setBackground(Color.decode("#FFF9EE"));
```

String at = "Lilavati is a treatise on mathematics by the Indian mathematician Bhaskara II, written in 1150. It is the first volume of his main work, the Siddhanta Shiromani, alongside the Bijaganita, the Grahaganita, and the Goladhyaya. There is a legend that says that Bhaskara composed this text for his daughter, who was called Lilavati. The book contains thirteen chapters, mainly definitions, arithmetical terms, interest computation, arithmetical and geometrical progressions, plane geometry, solid geometry, the shadow of the gnomon, the Kuttaka - a method to solve indeterminate equations, and combinations. Lilavati includes a number of methods of computing numbers such as multiplications, squares, and progressions, with examples using kings and elephants, objects which a common man could understand.";

```
JTextArea aboutText = new JTextArea(at);
aboutText.setBackground(Color.decode("#FFF9EE"));
aboutText.setBounds(126, 265, 468, 449);
aboutText.setLineWrap(true);
aboutText.setFont(new Font("Segoe UI", Font.BOLD , 18));
about.add(aboutText);
```

```
// // action
// JPanel action = new JPanel();
// addNewFrame1();

// view
JPanel view =new JPanel();
addTable(view);
// addButtons(view, 114, 241);

// info
JPanel info=new JPanel();
info.setBackground(Color.decode("#FFF9EE"));
// TODO: Add Info
```

String it = "Bhāskara II (c. 1114–1185), also known as Bhāskarāchārya ('Bhāskara, the teacher'), and as Bhāskara II to avoid confusion with Bhāskara I, was an Indian mathematician and astronomer. From verses, in his main work, Siddhānta Shiromani (सिद्धांतशिरोमणी), it can be inferred that he was born in 1114 in Vijjadavida (Vijjalavida) and living in the Sahyadri mountain ranges of Western Ghats, believed to be the town of Patan in Chalisgaon, located in present-day Khandesh region of Maharashtra by scholars.[6] He is the only ancient mathematician who has been immortalized on a monument. In a temple in Maharashtra, an inscription supposedly created by his grandson Changadeva, lists Bhaskaracharya's ancestral lineage for several generations before him as well as two generations after him.[7][8] Colebrooke who was the first European to translate (1817) Bhaskaracharya II's mathematical classics refers to the family as Maharashtrian Brahmins residing on the banks of the Godavari.";

```
JTextArea infoText = new JTextArea(it);
infoText.setBackground(Color.decode("#FFF9EE"));
infoText.setBounds(126, 265, 468, 449);
infoText.setLineWrap(true);
infoText.setFont(new Font("Segoe UI", Font.BOLD , 18));
info.add(infoText);
```

```
    tp.add("About", about);  
    tp.add("View", view);  
    tp.add("Background Info", info);  
    // tp.add("Actions", action);  
  
    frame.add(tp);  
    frame.setVisible(true);  
}  
  
}
```

CHAPTER 4

TESTING AND RESULTS

Testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate one or more properties of interest. Testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools. Some prefer saying Software testing as a White Box and Black Box Testing.

4.1 TESTING

1. Unit Testing

Individual components are tested to ensure that they operate correctly. Each component is tested independently, without other system components.

2. Module Testing

A module is a collection of dependent components such as a object class, an abstract data type or some looser collection of procedures and functions. They are module related components, so can be tested without other system modules.

3. System Testing

This is concerned with finding errors that result from unanticipated interaction between sub-system interface problems.

4. Acceptance Testing

The system is tested with data supplied by the system customer rather than simulated test data.

4.2 SNAPSHOTS

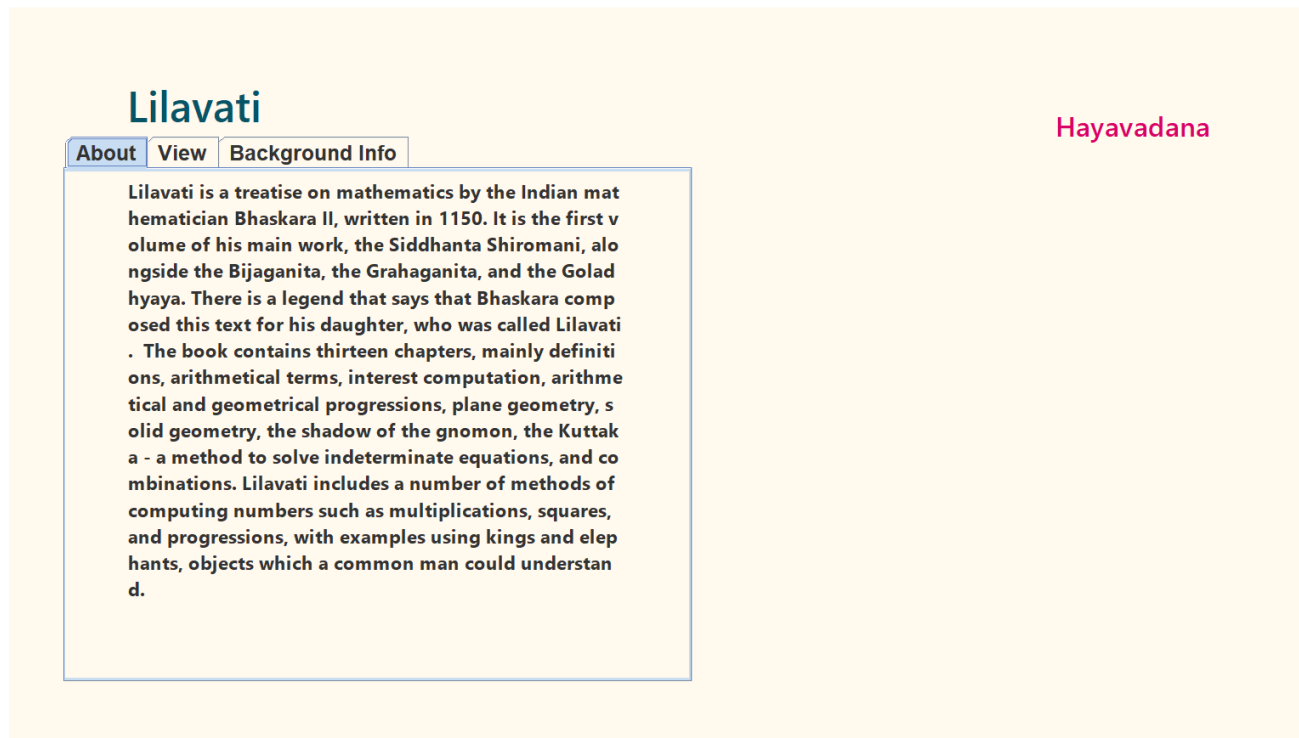


Fig 4.2.1: About

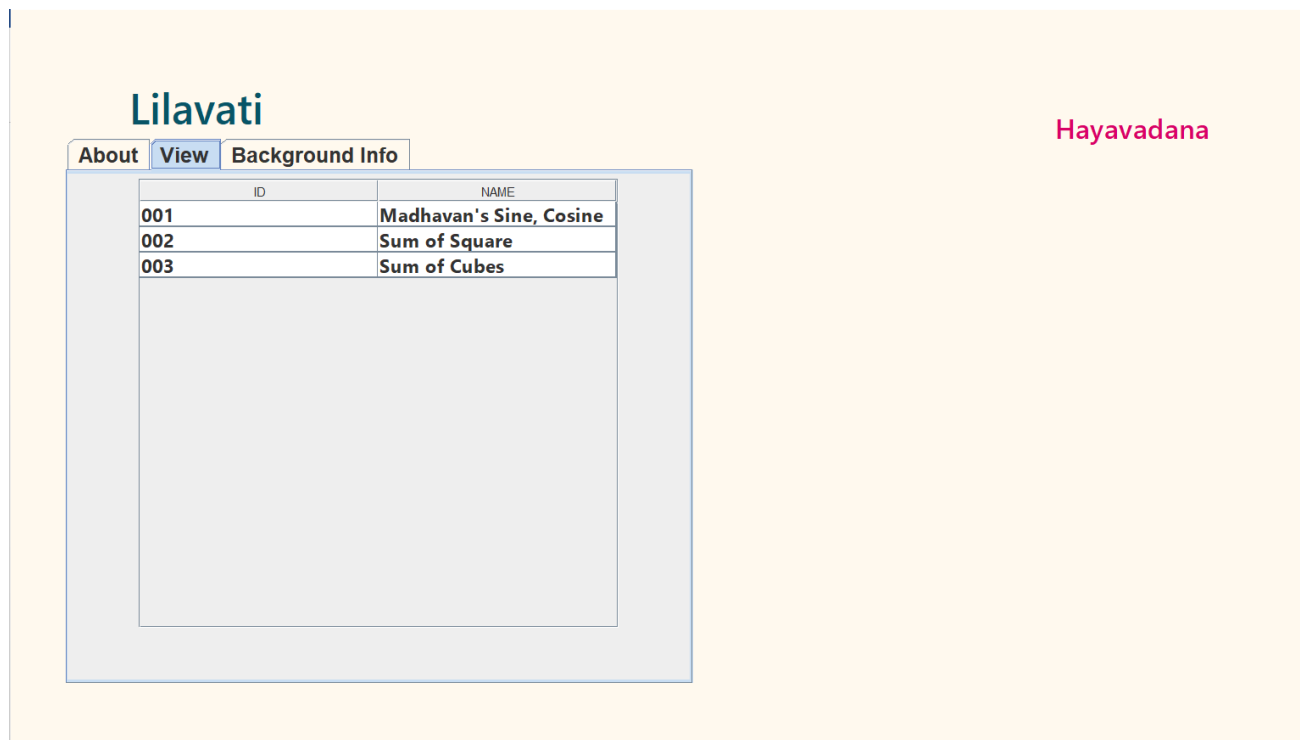


Fig 4.2.2: View

Lilavati

About
View
Background Info

Bhāskara II (c. 1114–1185), also known as Bhāskarāchārya ('Bhāskara, the teacher'), and as Bhāskara II to avoid confusion with Bhāskara I, was an Indian mathematician and astronomer. From verses, in his main work, Siddhānta Shiromani (सिद्धान्तशिरोमणि), it can be inferred that he was born in 1114 in Vijjadavida (Vijjalavida) and living in the Sahyadri mountain ranges of Western Ghats, believed to be the town of Patan in Chalisgaon, located in present-day Khandesh region of Maharashtra by scholars.[6] He is the only ancient mathematician who has been immortalized on a monument. In a temple in Maharashtra, an inscription supposedly created by his grandson Changadeva, lists Bhaskaracharya's ancestral lineage for several generations before him as well as two generations after him.[7][8] Colebrooke who was the first European to translate (1817) Bhaskaracharya II's mathematical classics refers to the family as Maharashtrian Brahmins residing on the banks of the Sahyadri.

Hayavadana

Fig 4.2.3: Background Info

Sum of Squares of n Terms:

Number of Terms:

Common Diff:

First term:

Submit

Sum of Cubes:

Number of Terms:

Submit

द्विगुणैकोनपदोत्तरकृतिहतिषष्टांशमुखचयहतयुतिः ।
व्येकेपदद्वा मुखकृतिसहिता पदताडितेष्टकृतिचितिका ॥

Fig 4.2.4: Working

CONCLUSION

We believe that the skills and the advancement in Mathematics our ancestors have seen, if implemented now would make a great difference in our understanding of many subjects and not limiting to just Mathematics but also Physics, Computational Fields, etc.

BIBLIOGRAPHY

- [1] Herbert Schildt, JAVA the Complete Reference, McGraw Hill, 12th Edition.
- [2] Jim Keogh, J2EE The Complete Reference, McGraw Hill Education; 1st Edition.
- [3] Pride of India: A Glimpse into India's Scientific Heritage, Samskrita Bharati
- [4] G´erard G. Emch, R. Sridharan, M. D. Srinivas, Contributions to the History of Indian Mathematics
- [5] Dr. S. Madhavan , Sadratnamala Of Sankara Varman (Text On Indian Astronomy And Mathematics) With English Translation And Notes, The Kuppuswamisastri Research Institute, Chennai
- [6] Dr. Shivangi Upadhyay, Lilavati (Bhaskara II)
- [7] The verses are sourced from the original works
- [8] <https://www.sanskritfromhome.org/course-details/lilavati-7208>
- [9] <https://www.sanskritfromhome.org/course-details/indian-mathematics-part1-32533>