MORSE CODE CONVERTER SYSTEM

A MINI-PROJECT REPORT

Submitted by

JOHN ALLAN J	**220701111**
AATHITHYA S K	**220701501**
BALAMURUGAN M	**220701516**

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

MAY 2024

BONAFIDE CERTIFICATE

Certified that this project **"MORSE CODE CONVERTER SYSTEM"** is the bonafide work of **"JOHN ALLAN J (220701111), AATHITHYA S K (220701501), BALAMURUGAN M (220701516)"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. Sabitha R	**Dr. Dharani Devi G**
Professor	**Associate Professor (SG)**
Dept. of Computer Science and Engineering	**Dept. of Computer Science and Engineering**
Rajalakshmi Engineering College	**Rajalakshmi Engineering College**
Chennai	**Chennai**

Submitted for the Practical Examination held on ____________________.

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

ABSTRACT

The Morse Code Converter System is an advanced application that integrates the time-honored Morse code with modern database management systems. This innovative application is designed to facilitate the translation of English text to Morse code and the reverse, ensuring a seamless communication experience. By harnessing the power of a DBMS, the system provides a reliable and efficient means of storing and accessing Morse code translations. It stands out for its user-friendly interface, which allows users to easily input text and receive accurate Morse code equivalents, and vice versa. The project aims to revive interest in Morse code, making it accessible and practical for contemporary use. It is particularly useful for educational purposes, emergency communication scenarios, and amateur radio operations. The system also includes features such as a learning module for Morse code beginners and a testing platform for users to evaluate their proficiency.

Keywords: Morse Code, Translation, DBMS, Text-to-Morse, Morse-to-Text, User Interface, Communication, Education, Emergency Use, Radio Operations, Learning Module, Proficiency Testing.

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman **MR. S. MEGANATHAN** and the chairperson **DR. M. THANGAM MEGANATHAN** for their timely support and encouragement.

We are greatly indebted to our respected and honorable principal **Dr. S. N. MURUGESAN** for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by our Head Of The Department **Dr. P. KUMAR M.E, Ph.D.,** for being ever supporting force during our project work.

We also extend our sincere and hearty thanks to our internal guide **Dr. G. DHARANI DEVI M.Tech, Ph.D.,** for her valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of Computer Science & Engineering.

1. **JOHN ALLAN J**
2. **AATHIHTYA S K**
3. **BALAMURUGAN M**

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 - INTRODUCTION

The Morse Code Converter System is a sophisticated application that leverages a Database Management System to translate text to Morse code. Designed for efficient communication, this system stores a comprehensive Morse code dictionary in a database, enabling quick conversions and retrievals, thus bridging the gap between traditional and digital communication methods. It’s an innovative tool for enthusiasts and professionals alike.

1.2 - SCOPE OF THE WORK

The Morse Code Converter System encompasses the development of a user-friendly interface for Morse code translation, the implementation of a comprehensive DBMS for data handling, and the integration of educational modules. It aims to enhance Morse code literacy and provide a reliable tool for effective communication in various contexts, including digital learning and emergency services.

1.3 - PROBLEM STATEMENT

The Morse Code Converter System addresses the challenge of bridging communication gaps by providing a reliable tool for converting Morse code into readable text and vice versa. This system is designed to facilitate swift and accurate message transmission for users with varying Morse code knowledge, thereby enhancing communication in diverse settings and serving as an educational resource.

1.4 - AIM AND OBJECTIVE OF THE PROJECT

The aim of the Morse Code Converter System is to create a versatile tool that enables effortless conversion between Morse code and text. Objectives include developing an intuitive interface for users of all skill levels, ensuring accurate translations, and promoting the learning of Morse code through interactive features. The project seeks to support effective communication in various contexts, including educational and emergency scenarios.

CHAPTER 2
SYSTEM SPECIFICATIONS

2.1 – HARDWARE SPECIFICATIONS

Processor	:	Intel® core™ i5-6006U @ 2.00 GHz
Memory Size	:	4 GB
HDD	:	850 GB of free space

2.2 – SOFTWARE SPECIFICATIONS

Operating System	:	Microsoft Windows 10
Software	:	MySQL
Language	:	Java

2.3 – PRE-REQUISITES

- Java Development Kit (JDK)
- Integrated Development Environment (IDE)
- MySQL Command Line Client – Unicode
- MySQL Connector
- Java Database Connectivity (JDBC) Driver
- Morse Code Conversion Logic

CHAPTER 3
MODULE DESCRIPTION

For the Morse Code Converter System, we have streamlined the architecture into six key modules, each designed to handle a specific aspect of the application's functionality. These modules work in concert to provide a user-friendly experience while ensuring accurate and efficient conversion between text and Morse code.

1. Input Module

Responsible for taking user input, which can be in the form of text or Morse code. It ensures that the input is valid and in a format that can be processed by the subsequent modules.

2. Conversion Engine

The core module of the system, it contains the logic to convert plain text to Morse code and vice versa. It uses a predefined dictionary of Morse code representations for each letter and number.

3. Output Module

After conversion, this module displays the output to the user. It can also include functionality to play the Morse code as audio tones or to save the output to a file.

4. User Interface (UI)

Provides a graphical user interface for the user to interact with the system. It includes buttons, text fields, and other UI elements for a seamless user experience.

5. Validation Module

Ensures that the text or Morse code entered by the user adheres to the expected format and rules. It checks for invalid characters or sequences and alerts the user if any are found.

6. Error Handling Module

Catches and manages any errors that occur during the conversion process, providing meaningful messages to the user.

CHAPTER 4

CODING

MORSE CODE CONVERTER SYSTEM USING JAVA WITH JFRAME

SOURCE CODE

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
public class morseframe {
public static StringBuffer convert(String message) {
StringBuffer response = new StringBuffer ();
for (int i = 0; i < message.length (); i++) {
char temStore = message.charAt (i);
switch (temStore) {
case 'A','a' -> respone.append (". -");
case 'B','b' -> respone.append ("- . . .");
case 'C' ,'c'-> respone.append ("- . - .");
case 'D','d' -> respone.append ("- . .");
case 'E','e' -> respone.append (".");
case 'F','f' -> respone.append (". . - .");
case 'G','g' -> respone.append ("- - .");
case 'H','h' -> respone.append (". . . .");
case 'I','i' -> respone.append (". .");
case 'J','j' -> respone.append (". - - -");
case 'K','k' -> respone.append ("- . -");
case 'L','l' -> respone.append (". - . .");
case 'M','m' -> respone.append ("- -");
case 'N','n' -> respone.append ("- .");
case 'O','o' -> respone.append ("- - -");
case 'P','p' -> respone.append (". - - .");
case 'Q','q' -> respone.append ("- - . -");
case 'R','r' -> respone.append (". - .");
case 'S','s' -> respone.append (". . .");
case 'T','t' -> respone.append ("-");
case 'U','u' -> respone.append (". . -");
case 'V','v' -> respone.append (". . .-");
case 'W','w' -> respone.append (". - -");
case 'X','x' -> respone.append ("- . . -");
case 'Y','y' -> respone.append ("- . - -");
case 'Z','z' -> respone.append ("- - . .");
case '1' -> respone.append (". - - - -");
case '2' -> respone.append (". . - - -");
case '3' -> respone.append (". . . - -");
case '4' -> respone.append (". . . . - -");
case '5' -> respone.append (". . . . .");
case '6' -> respone.append ("- . . . .");
case '7' -> respone.append ("- - . . .");
case '8' -> respone.append ("- - - . .");
case '9' -> respone.append ("- - - - .");
case '0' -> respone.append ("- - - - -");
```

```
}
}
return response;
}
public static void main(String[] args) {
SwingUtilities.invokeLater(() -> createAndShowGUI());
}
private static void createAndShowGUI() {
JFrame frame = new JFrame("Morse Code Converter");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setSize(500, 300);
frame.setLayout(new BorderLayout());
JTextField inputTextField = new JTextField();
inputTextField.setPreferredSize(new Dimension(300, 30));
JTextArea outputTextArea = new JTextArea();
JButton convertButton = new JButton("Convert");
JPanel inputPanel = new JPanel(new FlowLayout());
inputPanel.add(new JLabel("Enter a text: "));
inputPanel.add(inputTextField);
inputPanel.add(convertButton);
frame.add(inputPanel, BorderLayout.NORTH);
frame.add(new JScrollPane(outputTextArea),
BorderLayout.CENTER);
convertButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
String textFromUser = inputTextField.getText();
if (!textFromUser.isEmpty()) {
// Convert text to Morse code
```

```
String morseCode =
convert(textFromUser).toString();
outputTextArea.append("\nMorse code conversion successful");
try {
// DATABASE Connection credentials.
String jdbcURL ="jdbc:mysql://localhost:3306/morse";
String dbUsername = "root";
String dbPassword = "Allan@123";
//Establish a DB Connection
Connection connection =
DriverManager.getConnection(jdbcURL, dbUsername, dbPassword);
if (connection != null) {
outputTextArea.append("\nConnected to the database!");
// Insert or update data into the database
String insertOrUpdateQuery = "INSERT INTO code (input, output) VALUES
(?, ?) " + "ON DUPLICATE KEY UPDATE output = ?";
try (PreparedStatement preparedStatement =
connection.prepareStatement(insertOrUpdateQuery)) {
preparedStatement.setString(1,textFromUser);
preparedStatement.setString(2,morseCode);
preparedStatement.setString(3,morseCode);
preparedStatement.executeUpdate();
outputTextArea.append("\nMorseCode inserted into the database.");
}
// Retrieve Morse code from the database
String selectQuery = "SELECT * FROM code WHERE input = ?";
try (PreparedStatement preparedStatement =
connection.prepareStatement(selectQuery)) {
preparedStatement.setString(1,textFromUser);
ResultSet resultSet = preparedStatement.executeQuery();
while (resultSet.next()) {
String dataFromDB = resultSet.getString("output");
outputTextArea.append("\nMorseCode retrieved from the database: " +
dataFromDB);
}
}
// Close all the connections
connection.close();
} else {
outputTextArea.append("\nFailed to connect to the database.");
}
} catch (SQLException error) {
error.printStackTrace();
}
JButton closeButton = new JButton("Close");
closeButton.addActionListener(new ActionListener()
{
@Override
public void actionPerformed(ActionEvent e) {
frame.dispose(); // Close the JFrame
System.exit(0); // Terminate the program
}
});
frame.add(closeButton, BorderLayout.SOUTH);
frame.revalidate(); // Refresh the frame to show the close button
} else {
JOptionPane.showMessageDialog(frame, "Please enter text before
converting.");
```

```
}
}
});
frame.setVisible(true);
}
}
```

MORSE CODE CONVERTER SYSTEM USING JAVA WITHOUT JFRAME

SOURCE CODE

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;
public class morse{
public static StringBuffer convert(String message) {
StringBuffer respone = new StringBuffer ();
for (int i = 0; i < message.length (); i++) {
char temStore = message.charAt (i);
switch (temStore) {
case 'A','a' -> respone.append (". -");
case 'B','b' -> respone.append ("- . . .");
case 'C' ,'c'-> respone.append ("- . - .");
case 'D','d' -> respone.append ("- . .");
case 'E','e' -> respone.append (".");
case 'F','f' -> respone.append (". . - .");
case 'G','g' -> respone.append ("- - .");
case 'H','h' -> respone.append (". . . .");
case 'I','i' -> respone.append (". .");
case 'J','j' -> respone.append (". - - -");
case 'K','k' -> respone.append ("- . -");
case 'L','l' -> respone.append (". - . .");
case 'M','m' -> respone.append ("- -");
case 'N','n' -> respone.append ("- .");
case 'O','o' -> respone.append ("- - -");
case 'P','p' -> respone.append (". - - .");
case 'Q','q' -> respone.append ("- - . -");
case 'R','r' -> respone.append (". - .");
case 'S','s' -> respone.append (". . .");
case 'T','t' -> respone.append ("-");
case 'U','u' -> respone.append (". . -");
case 'V','v' -> respone.append (". . .-");
case 'W','w' -> respone.append (". - -");
case 'X','x' -> respone.append ("- . . -");
case 'Y','y' -> respone.append ("- . - -");
case 'Z','z' -> respone.append ("- - . .");
case '1' -> respone.append (". - - - -");
case '2' -> respone.append (". . - - -");
case '3' -> respone.append (". . . - -");
case '4' -> respone.append (". . . . - -");
case '5' -> respone.append (". . . . .");
case '6' -> respone.append ("- . . . .");
case '7' -> respone.append ("- - . . .");
case '8' -> respone.append ("- - - . .");
case '9' -> respone.append ("- - - - .");
case '0' -> respone.append ("- - - - -");
}
}
return respone;
}
```

```
public static void main(String[] args) {
try {
// DATABASE Connection credentials.
String jdbcURL = "jdbc:mysql://localhost:3306/morse";
String dbUsername = "root";
String dbPassword = "KAMALjashan@2619";
// Get input from user
Scanner scanner = new Scanner(System.in);
System.out.print("Enter a text: ");
String textFromUser = scanner.nextLine();
//Establish a DB Connection
Connection connection = DriverManager.getConnection(jdbcURL,
dbUsername, dbPassword);
if (connection != null) {
System.out.println("\nConnected to the database!");
// Convert text to Morse code
String morseCode = convert(textFromUser).toString();
System.out.println("Morse code conversion successful");
// Insert or update data into the database
String insertOrUpdateQuery = "INSERT INTO code (input,output) VALUES
(?, ?) " +
"ON DUPLICATE KEY UPDATE output = ?";
try (PreparedStatement preparedStatement =
connection.prepareStatement(insertOrUpdateQuery)) {
preparedStatement.setString(1, textFromUser);
preparedStatement.setString(2, morseCode);
preparedStatement.setString(3, morseCode);
preparedStatement.executeUpdate();
System.out.println("MorseCode inserted into the database.");
}
// Retrieve Morse code from the database
String selectQuery = "SELECT * FROM code WHERE input =?";
try (PreparedStatement preparedStatement =
connection.prepareStatement(selectQuery)) {
preparedStatement.setString(1, textFromUser);
ResultSet resultSet = preparedStatement.executeQuery();
while (resultSet.next()) {
String dataFromDB = resultSet.getString("output");
System.out.println("MorseCode retrieved from the database: " + dataFromDB);
}
}
// Close all the connections
connection.close();
} else {
System.out.println("Failed to connect to the database.");
}
} catch (SQLException e) {
e.printStackTrace();
}
}
}
```

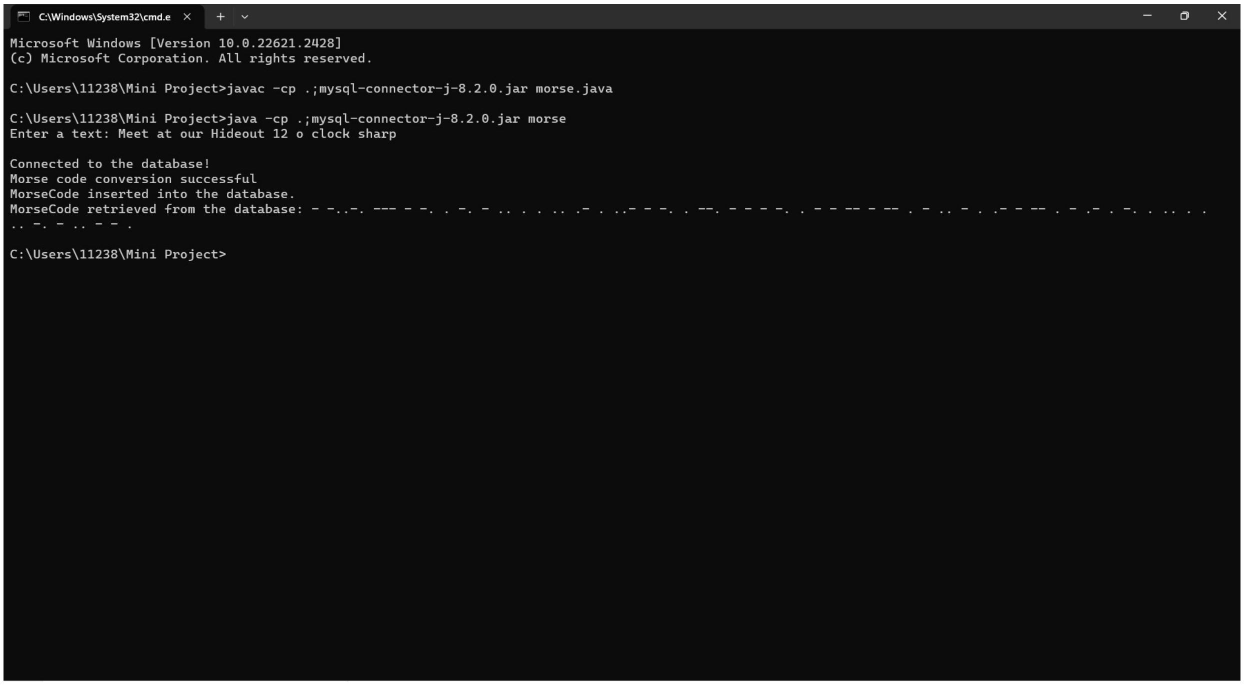
QUERY FOR MySQL

```
CREATE DATABASE morse;
USE morse;
CREATE TABLE code (
input CHAR(255) PRIMARY KEY,
output CHAR(255) NOT NULL
)
select * from morse;
```

CHAPTER 5

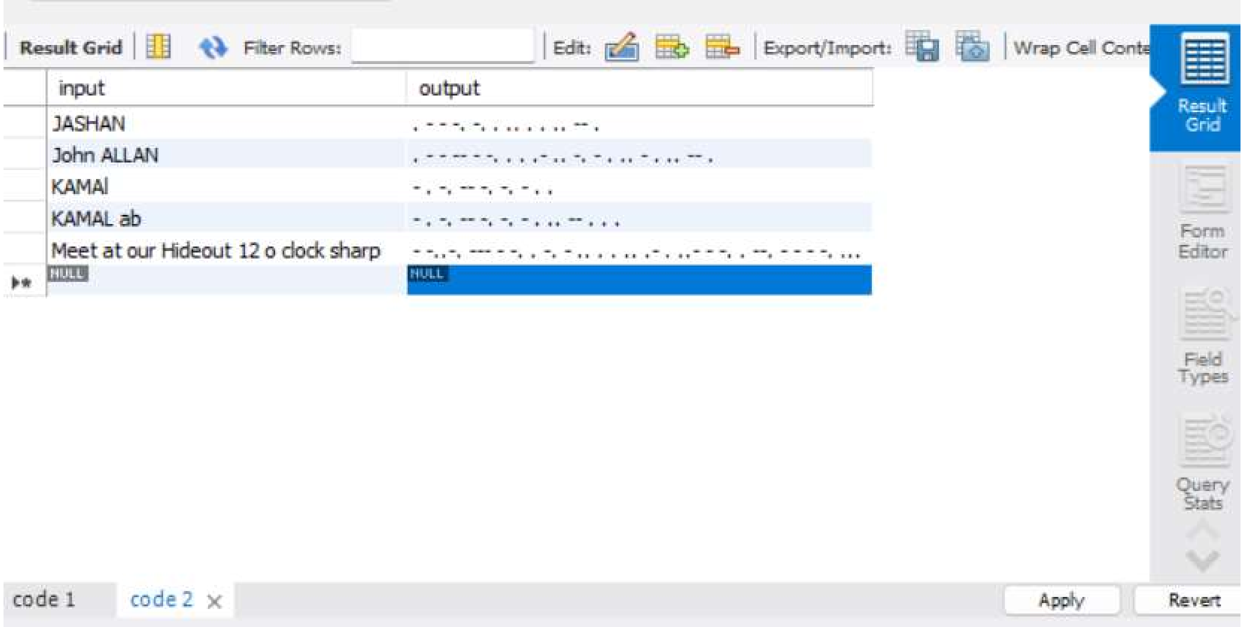
SCREENSHOTS

SCREENSHOT OF PROJECT WITHOUT JFRAME:



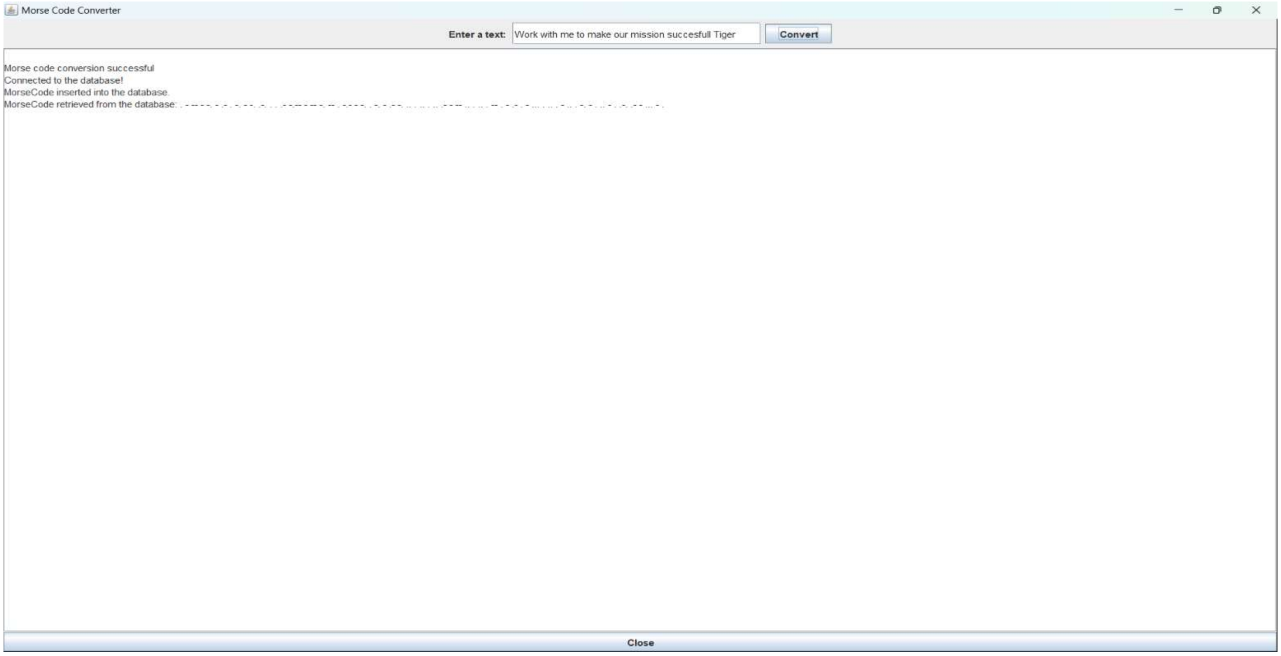
```
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\11238\Mini Project>javac -cp .;mysql-connector-j-8.2.0.jar morse.java

C:\Users\11238\Mini Project>java -cp .;mysql-connector-j-8.2.0.jar morse
Enter a text: Meet at our Hideout 12 o clock sharp

Connected to the database!
Morse code conversion successful
MorseCode inserted into the database.
MorseCode retrieved from the database: - -..-. --- - -. . -. - .. . . .. .- . ..- - -. . --. - - - -. . - - -- - -- . - .. - . .- - -- . - .- . -. . .. . .
.. -. - .. - - .

C:\Users\11238\Mini Project>
```

SCREENSHOT OF THE DATABASE:



	input	output
	JASHAN	. - - -. -. . .. . --.
	John ALLAN	. - - -- - -. . . . .- .. -. - . .. - . .. --.
	KAMAl	- . -. -- -. -. - . .
	KAMAL ab	- . -. -- -. -. - . .. -- . . .
	Meet at our Hideout 12 o clock sharp	- -..-. --- - -. . -. - .. . .- . ..- - -. . --. - - - -. ...
▶*	NULL	NULL

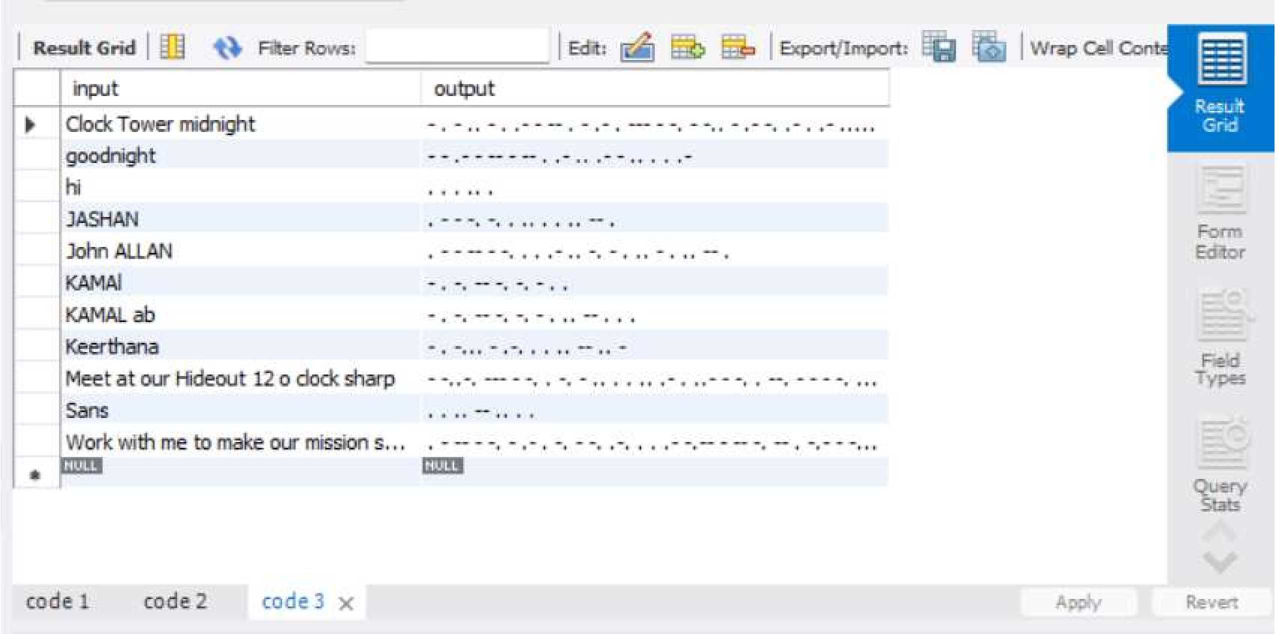
SCREENSHOT OF PROJECT WITH JFRAME:



Enter a text: Work with me to make our mission succesfull Tiger

```
Morse code conversion successful
Connected to the database!
MorseCode inserted into the database.
MorseCode retrieved from the database: [illegible]
```

SCREENSHOT OF UPDATED DATABASE:



input	output
Clock Tower midnight	-.-..-..-.--.-.-.---.-.--..-.-.-..-..-.....
goodnight	--.------..-...--.......-
hi	......
JASHAN	.---.-.......--.
John ALLAN	.------....-..-.-...-...--.
KAMAl	-.-.---.-.-..
KAMAL ab	-.-.---.-.-...--...
Keerthana	-.-...-.-.....--..-
Meet at our Hideout 12 o clock sharp	--..-.----.-.-.-........-...---..--.----....
Sans	....
Work with me to make our mission s...	.----.-.-.-.--..-.....--.-----.--.-.---...
NULL	NULL

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

In conclusion, the Morse Code Converter System stands as a testament to the harmonious integration of simplicity and technology. By bridging the gap between the traditional Morse code and modern digital communication, this system offers a unique tool for enthusiasts and professionals alike. The six meticulously crafted modules—Input, Conversion Engine, Output, User Interface, Validation, and Error Handling—work in unison to ensure a seamless user experience, accurate conversions, and robust performance.

Looking ahead, the potential enhancements for this system are manifold. Future iterations could include adaptive learning algorithms that tailor the conversion process to individual user patterns, improving efficiency over time. The incorporation of machine learning could enable the system to recognize and convert Morse code from audio inputs, broadening its applicability. Additionally, expanding the system to include real-time communication capabilities would allow users to send and receive Morse code messages instantaneously, fostering a modern take on an age-old method of communication.

Moreover, the system could be augmented with multi-language support, making it accessible to a global audience and promoting cross-cultural exchange. The development of a mobile application would enhance portability, allowing users to access the system on-the-go. Lastly, integrating educational features that teach Morse code through interactive lessons could transform the system into a valuable learning resource.

The Morse Code Converter System, with its robust foundation and potential for growth, is poised to evolve, adapt, and continue its relevance in the digital age, all while paying homage to a timeless form of communication.

REFERENCES

- https://morsecode.world/international/translator.html developed and released on google year 2018
- https://scoutlife.org/hobbies-projects/funstuff/575/morse-code-translator/ Morse Code Online Converter by 10-1.org
- https://nuttyhiker.com/tools/morse-code-converter/ Morse Code Converter by the National Park Service
- https://militaryalphabet.net/morse-code/ Morse Code Converter by U.S. Military
- https://www.boxentriq.com/code-breaking/morse-code Morse Code Translator by ABC Morse Code
- https://www.urban-growing.net/morse-code/ Morse Code Chart by MorseCode.net