



*Green University of Bangladesh*

*Department of Computer Science and Engineering (CSE)  
Semester: (Fall, Year: 2023), B.Sc. in CSE (Day)*

---

## **Data Transmission Simulator**

---

*Course Title: Data Communication Lab  
Course Code: CSE - 308  
Section: 213 D3*

Students Details

<b>Name</b>	<b>ID</b>
MD SAIFUL ISLAM RIMON	213002039
NAZMUN NAHAR	191002251

*Submission Date: 09-01-24  
Course Teacher's Name: MS. RUSMITA HALIM CHAITY*

[For teachers use only: **Don't write anything inside this box**]

<u>Lab Project Status</u>	
<b>Marks:</b>	<b>Signature:</b>
<b>Comments:</b>	<b>Date:</b>

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Overview . . . . .	3
1.2	Motivation . . . . .	3
1.3	Problem Definition . . . . .	3
1.3.1	Problem Statement . . . . .	3
1.3.2	Complex Engineering Problem . . . . .	4
1.4	Design Goals/Objectives . . . . .	5
1.5	Application . . . . .	6
<b>2</b>	<b>Design/Development/Implementation of the Project</b>	<b>7</b>
2.1	Introduction . . . . .	7
2.2	Project Details . . . . .	7
2.3	Implementation . . . . .	9
2.4	Algorithms . . . . .	50
<b>3</b>	<b>Performance Evaluation</b>	<b>51</b>
3.1	Simulation Environment/ Simulation Procedure . . . . .	51
3.2	Results Analysis/Testing . . . . .	52
3.2.1	Home Page . . . . .	52
3.2.2	Line Coding . . . . .	53
3.2.3	Hamming Encoding & Decoding . . . . .	57
3.2.4	Analog to Digital Signal Conversion . . . . .	58
3.2.5	Character Stuffing & Destuffing . . . . .	59
3.2.6	IPv4 Conversion . . . . .	61
3.2.7	Cyclic Redundancy Check (CRC) . . . . .	63
3.3	Results Overall Discussion . . . . .	65
3.3.1	Complex Engineering Problem Discussion . . . . .	65

<b>4</b>	<b>Conclusion</b>	<b>67</b>
4.1	Discussion . . . . .	67
4.2	Limitations . . . . .	67
4.3	Scope of Future Work . . . . .	68

# Chapter 1

## Introduction

### 1.1 Overview

The goal of the "**Data Transmission Simulator**" project is to create a thorough modeling tool for different data transfer methods. A wide range of techniques will be covered by this simulator, such as bit and character stuffing and de-stuffing, encoding and decoding schemes such as NRZ-I, Manchester, AMI, and pseudo ternary, error detection and correction using Hamming Code, and Cyclic Redundancy Check (CRC) using Parity Checker. It will also have an IPv4 implementation for converting Decimal to Binary and vice versa.

### 1.2 Motivation

Efficient and error-free data transfer is critical in the digital era. It is essential for academics, professionals, and students studying computer science and telecommunications to comprehend and replicate these mechanisms. Nevertheless, interactive tools covering a wide variety of data transmission methods are hard to come across. Our study seeks to close this knowledge gap by offering a practical learning aid for simulating and comprehending these important methods.

### 1.3 Problem Definition

#### 1.3.1 Problem Statement

It is now quite difficult to fully comprehend and illustrate the different facets of data transmission in a research or instructional context. A comprehensive simulation tool is required, one that not only illustrates the theoretical elements but also enables users to interactively investigate and comprehend the real-world ramifications of various data transmission strategies.

### 1.3.2 Complex Engineering Problem

Table 1.1: Summary of the attributes touched by the mentioned projects

Name of the P Attributes	Explain how to address
<b>P1:</b> Depth of knowledge required	A solid understanding of digital communications theory, including error correction techniques and data encoding schemes, is required for this project. It's essential to be proficient in programming. Accurate simulation of data transmission processes requires a thorough understanding of networking protocols, stuffing and de-stuffing, encoding and decoding, IPv4, etc.
<b>P2:</b> Range of conflicting requirements	—
<b>P3:</b> Depth of analysis required	Conducting a concise performance analysis of data transmission techniques is essential, focusing on the effectiveness of error detection and correction algorithms like CRC and Hamming Code and then the analysis of stuffing and de-stuffing. It's important to assess user interaction and the simulation is understandable to the end user.
<b>P4:</b> Familiarity of issues	—
<b>P5:</b> Extent of applicable codes	This project includes lots of algorithms and techniques integrated within a single project which is helpful for educational purposes, research and so on. So, we can say that, this indicates a professional and standard project. This project includes lots of creative practice here.
<b>P6:</b> Extent of stakeholder involvement and conflicting requirements	—
<b>P7:</b> Interdependence	—

## 1.4 Design Goals/Objectives

The project's design goals and objectives are as follows:

1. **Comprehensive Coverage:** To include a wide range of data transmission techniques from basic stuffing and de-stuffing to complex encoding/decoding schemes.
2. **Interactive Simulation:** Provide a user-friendly interface for simulating and visualizing different data transmission scenarios.
3. **Accuracy and Reliability:** Ensure the simulations are accurate and reflect real-world scenarios.
4. **Educational Utility:** Design the simulator to be an effective educational tool for students and professionals alike.
5. **User-Friendly Interface:** Create an intuitive and user-friendly interface
6. **Extensibility:** Build the simulator in a way that allows future expansion and inclusion of more data transmission techniques.

## 1.5 Application

**Data Transmission Simulator** is designed for use in various facilities, including:

- **Educational Tool:** Ideal for educational institutions for teaching data communication and networking concepts.
- **Research:** Researchers can use the simulator to test and visualize different data transmission scenarios.
- **Professional Training:** Professionals in telecommunications and IT can use the simulator for training and development purposes.
- **Network Planning and Analysis:** Helps in planning and analyzing network data flow and error handling mechanisms.

# Chapter 2

## Design/Development/Implementation of the Project

### 2.1 Introduction

This project is a comprehensive Matlab language program designed for a **Data Transmission Simulator**. The code is written in Matlab language and is intended to run on the Matlab App Designer, specifically using the Matlab software.

This project is a robust example of a transmission simulator application written in Matlab language, showcasing efficient use of Matlab packages, Matlab code & tools, Matlab App Designer program, tools in App Designer Program & how to use those tools to embed all the functionalities of application development.

### 2.2 Project Details

#### Key Features:

1. **User Interface:** The code starts with a menu-driven interface which we are calling "Home page", offering various options like
  - (a) Line Coding
  - (b) Hamming Encoding & Decoding
  - (c) Analog to Digital Signal Conversion
  - (d) Character Stuffing & Destuffing
  - (e) IPv4 Conversion
  - (f) Bit Stuffing & Destuffing
  - (g) CRC Error Detection
2. **Easy to use:** The program consists of input boxes, output boxes, a graphical demonstration system for signals and buttons for different operations.



## Technical Overview:

- **Matlab Language:** The program is written in Matlab language, which provides high-level control over hardware and is ideal for user-level programming including GUI (Graphical User Interface).
- **Functions for Operations:** The code is organized into multiple functions, each handling a specific task. In the whole program, every single task is done in function including lite operations like: buttons, input box, output box etc.
- **Classes for Operations:** Classes are available here for different purposes like Public or Private declarations and others. Some of the functions, variables or other package are in public access and some are in private access based on the operation.
- **Callbacks for Operations:** Callbacks are available here for calling a specific task in the background of that particular object like: buttons, input boxes, output boxes, signal generators.



Figure 2.1: Features are embedded in this project

## 2.3 Implementation

The implementation of the Data Transmission Simulator in Matlab language encompasses various elements detailed in multiple subsections. This section will cover the main aspects, including the workflow and the tools and libraries used.

### The workflow

#### Start Screen and Home Page

- Upon launching the app, display a start screen that transitions to a home page.
- The home page should present a menu with the options listed (Line Coding, Hamming Encoding & Decoding, etc.).

#### Menu Selection

- Implement a menu-driven interface where users can select any of the options (a-g) to perform a specific task.

#### Task Modules

- For each option selected from the menu, open the corresponding module:
  - Line Coding: A module that converts data into a line code format, suitable for transmission.
  - Hamming Encoding & Decoding: A module for error-detecting and error-correcting code operations.
  - Analog to Digital Signal Conversion: A module that samples analog signals and converts them into digital.
  - Character Stuffing & Destuffing: A module that adds or removes special characters to/from data to provide a clear start and end signal.
  - IPv4 Conversion: A module that deals with the conversion and manipulation of IPv4 addresses.
  - Bit Stuffing & Destuffing: A module for inserting or removing bits from data to prevent misinterpretation.
  - CRC Error Detection: A module that calculates and checks CRC values for error detection in data transmission.

#### Input and Output

- Each module should provide input boxes for user input and display output boxes for results.
- Where applicable, provide graphical representation of the data or signals.

## **Operations and Interactivity**

- Include buttons and interactive elements for users to execute operations like 'encode', 'decode', 'convert', etc.
- Provide clear instructions or tooltips for users to understand what each button does.

## **Validation and Feedback**

- Include data validation to ensure users input data in the correct format.
- Provide immediate feedback on operations, such as success messages or error prompts.

## **Documentation**

- Offer a help section or documentation within the app to assist users with complex tasks or to provide information about the algorithms used.

## **Exit Workflow**

- Allow users to easily navigate back to the home page or exit the app from any module.

## **Implementation details (with screenshots and programming codes)**

## CODE [Home Page]:

```
1 classdef home < matlab.apps.AppBase
2
3     % Properties that correspond to app components
4     properties (Access = public)
5         UIFigure                matlab.ui.Figure
6         DeveloperButton          matlab.ui.control.Button
7         FeedbackButton           matlab.ui.control.Button
8         DocumentationButton       matlab.ui.control.Button
9         EXITButton               matlab.ui.control.Button
10        CRCButton                matlab.ui.control.Button
11        BITSTUFFINGDESTUFFINGButton  matlab.ui.control.Button
12        CHARACTERSTUFFINGDESTUFFINGButton  matlab.ui.control.Button
13        HAMMINGENCODINGDECODINGButton  matlab.ui.control.Button
14        IPv4CONVERSIONButton_2      matlab.ui.control.Button
15        ANALOGTODIGITALSIGNALCONVERSIONButton  matlab.ui.control.Button
16        LINECODINGButton            matlab.ui.control.Button
17        Label                      matlab.ui.control.Label
18        DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label  matlab.
19        ↪ ui.control.Label
20        Image2                    matlab.ui.control.Image
21        Image                      matlab.ui.control.Image
22        DATATRANSMISSIONSIMULATORLabel  matlab.ui.control.Label
23        Image3                    matlab.ui.control.Image
24    end
25
26    % Callbacks that handle component events
27    methods (Access = private)
28
29        % Button pushed function: LINECODINGButton
30        function LINECODINGButtonPushed(app, event)
31            app.callingapp = app1;
32        end
33
34        % Button pushed function: IPv4CONVERSIONButton_2
35        function IPv4CONVERSIONButton_2Pushed(app, event)
36            app.callingapp = ipv4;
37        end
38
39        % Button pushed function: HAMMINGENCODINGDECODINGButton
40        function HAMMINGENCODINGDECODINGButtonPushed(app, event)
41            app.callingapp = hammingcode;
42        end
43
44        % Button pushed function: ANALOGTODIGITALSIGNALCONVERSIONButton
45        function ANALOGTODIGITALSIGNALCONVERSIONButtonPushed(app, event)
46            app.callingapp = ADC;
47        end
48
49        % Button pushed function: CHARACTERSTUFFINGDESTUFFINGButton
50        function CHARACTERSTUFFINGDESTUFFINGButtonPushed(app, event)
51            app.callingapp = CharacterStuffingDestuffing;
52        end
53
54        % Button pushed function: BITSTUFFINGDESTUFFINGButton
55        function BITSTUFFINGDESTUFFINGButtonPushed(app, event)
56
57        end
58
59        % Button pushed function: CRCButton
60        function CRCButtonPushed(app, event)
61            app.callingapp = crcapp;
62        end
63
64        % Image clicked function: Image3
65        function Image3Clicked(app, event)
66
67        end
68
69        % Button pushed function: EXITButton
70        function EXITButtonPushed(app, event)
71            % Close all figure windows
```

```

71     figHandles = findall(0, 'Type', 'figure');
72     delete(figHandles);
73     end
74
75     % Button pushed function: DocumentationButton
76     function DocumentationButtonPushed(app, event)
77         % Specify the URL you want to open
78         url = 'https://drive.google.com/drive/folders/1BMGfCQGcDsBmbeMB_
           ↪ Ep5IJiPZcHgZz15?usp=sharing';
79
80         % Open the link in the default web browser
81         web(url, '-browser');
82     end
83
84     % Button pushed function: FeedbackButton
85     function FeedbackButtonPushed(app, event)
86         email = 'saifulislamrimon2014@gmail.com';
87
88         % Create a "mailto" link
89         mailtoLink = ['mailto:' email];
90
91         % Open the link in the default email client
92         web(mailtoLink, '-browser');
93     end
94
95     % Button pushed function: DeveloperButton
96     function DeveloperButtonPushed(app, event)
97         % Specify the URL you want to open
98         url = 'sites.google.com/view/mdsaifulislamrimon';
99
100        % Open the link in the default web browser
101        web(url, '-browser');
102    end
103 end
104
105 % Component initialization
106 methods (Access = private)
107
108 % Create UIFigure and components
109 function createComponents(app)
110
111     % Get the file path for locating images
112     pathToMLAPP = fileparts(mfilename('fullpath'));
113
114     % Create UIFigure and hide until all components are created
115     app.UIFigure = uifigure('Visible', 'off');
116     app.UIFigure.Position = [100 100 640 480];
117     app.UIFigure.Name = 'MATLAB App';
118
119     % Create Image3
120     app.Image3 = uiimage(app.UIFigure);
121     app.Image3.ImageClickedFcn = createCallbackFcn(app, @Image3Clicked,
           ↪ true);
122     app.Image3.Position = [-147 -17 937 672];
123     app.Image3.ImageSource = fullfile(pathToMLAPP, 'bg.jpg');
124
125     % Create DATATRANSMISSIONSIMULATORLabel
126     app.DATATRANSMISSIONSIMULATORLabel = uilabel(app.UIFigure);
127     app.DATATRANSMISSIONSIMULATORLabel.FontSize = 36;
128     app.DATATRANSMISSIONSIMULATORLabel.FontWeight = 'bold';
129     app.DATATRANSMISSIONSIMULATORLabel.Position = [17 353 618 47];
130     app.DATATRANSMISSIONSIMULATORLabel.Text = 'DATA TRANSMISSION
           ↪ SIMULATOR';
131
132     % Create Image
133     app.Image = uiimage(app.UIFigure);
134     app.Image.Position = [1 358 189 166];
135     app.Image.ImageSource = fullfile(pathToMLAPP, 'Logo-PNG.png');
136
137     % Create Image2
138     app.Image2 = uiimage(app.UIFigure);
139     app.Image2.Position = [422 339 219 199];
140     app.Image2.ImageSource = fullfile(pathToMLAPP, 'cse-dept-logo.png');

```

```

141
142 % Create
    ↳ DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label
143 app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label =
    ↳ uilabel(app.UIFigure);
144 app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
    ↳ HorizontalAlignment = 'center';
145 app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
    ↳ FontWeight = 'bold';
146 app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
    ↳ Position = [60 3 521 22];
147 app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
    ↳ Text = 'DEVELOPED BY: MD SAIFUL ISLAM RIMON (213002039) &
    ↳ NAZMUN NAHAR (191000000)';
148
149 % Create Label
150 app.Label = uilabel(app.UIFigure);
151 app.Label.HorizontalAlignment = 'center';
152 app.Label.FontWeight = 'bold';
153 app.Label.Position = [68 28 499 22];
154 app.Label.Text = 'PROJECT SUPERVISOR: RUSMITA HALIM CHAITY, LECTURER
    ↳ , DEPT. OF CSE, GUB';
155
156 % Create LINECODINGButton
157 app.LINECODINGButton = uibutton(app.UIFigure, 'push');
158 app.LINECODINGButton.ButtonPushedFcn = createCallbackFcn(app,
    ↳ @LINECODINGButtonPushed, true);
159 app.LINECODINGButton.BackgroundColor = [0.4824 0.8 0.2863];
160 app.LINECODINGButton.FontSize = 18;
161 app.LINECODINGButton.FontWeight = 'bold';
162 app.LINECODINGButton.Position = [39 272 178 37];
163 app.LINECODINGButton.Text = 'LINE CODING';
164
165 % Create ANALOGTODIGITALSIGNALCONVERSIONButton
166 app.ANALOGTODIGITALSIGNALCONVERSIONButton = uibutton(app.UIFigure, '
    ↳ push');
167 app.ANALOGTODIGITALSIGNALCONVERSIONButton.ButtonPushedFcn =
    ↳ createCallbackFcn(app,
    ↳ @ANALOGTODIGITALSIGNALCONVERSIONButtonPushed, true);
168 app.ANALOGTODIGITALSIGNALCONVERSIONButton.BackgroundColor = [0.4824
    ↳ 0.8 0.2863];
169 app.ANALOGTODIGITALSIGNALCONVERSIONButton.FontSize = 14;
170 app.ANALOGTODIGITALSIGNALCONVERSIONButton.FontWeight = 'bold';
171 app.ANALOGTODIGITALSIGNALCONVERSIONButton.Position = [37 209 180 42];
172 app.ANALOGTODIGITALSIGNALCONVERSIONButton.Text = {'ANALOG TO DIGITAL'
    ↳ ; 'SIGNAL CONVERSION'};
173
174 % Create IPv4CONVERSIONButton_2
175 app.IPv4CONVERSIONButton_2 = uibutton(app.UIFigure, 'push');
176 app.IPv4CONVERSIONButton_2.ButtonPushedFcn = createCallbackFcn(app,
    ↳ @IPv4CONVERSIONButton_2Pushed, true);
177 app.IPv4CONVERSIONButton_2.BackgroundColor = [0.4784 0.8 0.2902];
178 app.IPv4CONVERSIONButton_2.FontSize = 18;
179 app.IPv4CONVERSIONButton_2.FontWeight = 'bold';
180 app.IPv4CONVERSIONButton_2.Position = [39 154 178 37];
181 app.IPv4CONVERSIONButton_2.Text = 'IPv4 CONVERSION';
182
183 % Create HAMMINGENCODINGDECODINGButton
184 app.HAMMINGENCODINGDECODINGButton = uibutton(app.UIFigure, 'push');
185 app.HAMMINGENCODINGDECODINGButton.ButtonPushedFcn = createCallbackFcn
    ↳ (app, @HAMMINGENCODINGDECODINGButtonPushed, true);
186 app.HAMMINGENCODINGDECODINGButton.BackgroundColor = [0.4824 0.8
    ↳ 0.2863];
187 app.HAMMINGENCODINGDECODINGButton.FontSize = 14;
188 app.HAMMINGENCODINGDECODINGButton.FontWeight = 'bold';
189 app.HAMMINGENCODINGDECODINGButton.Position = [394 267 199 42];
190 app.HAMMINGENCODINGDECODINGButton.Text = {'HAMMING ENCODING &'; '
    ↳ DECODING'};
191
192 % Create CHARACTERSTUFFINGDESTUFFINGButton
193 app.CHARACTERSTUFFINGDESTUFFINGButton = uibutton(app.UIFigure, 'push'
    ↳ );

```

```

194 app.CHARACTERSTUFFINGDESTUFFINGButton.ButtonPushedFcn =
    ↳ createCallbackFcn(app, @CHARACTERSTUFFINGDESTUFFINGButtonPushed
    ↳ , true);
195 app.CHARACTERSTUFFINGDESTUFFINGButton.BackgroundColor = [0.4824 0.8
    ↳ 0.2863];
196 app.CHARACTERSTUFFINGDESTUFFINGButton.FontSize = 14;
197 app.CHARACTERSTUFFINGDESTUFFINGButton.FontWeight = 'bold';
198 app.CHARACTERSTUFFINGDESTUFFINGButton.Position = [394 209 199 42];
199 app.CHARACTERSTUFFINGDESTUFFINGButton.Text = {'CHARACTER STUFFING &';
    ↳ 'DESTUFFING'};

200
201 % Create BITSTUFFINGDESTUFFINGButton
202 app.BITSTUFFINGDESTUFFINGButton = uibutton(app.UIFigure, 'push');
203 app.BITSTUFFINGDESTUFFINGButton.ButtonPushedFcn = createCallbackFcn(
    ↳ app, @BITSTUFFINGDESTUFFINGButtonPushed, true);
204 app.BITSTUFFINGDESTUFFINGButton.BackgroundColor = [0.4784 0.8
    ↳ 0.2902];
205 app.BITSTUFFINGDESTUFFINGButton.FontSize = 14;
206 app.BITSTUFFINGDESTUFFINGButton.FontWeight = 'bold';
207 app.BITSTUFFINGDESTUFFINGButton.Position = [698 133 199 42];
208 app.BITSTUFFINGDESTUFFINGButton.Text = {'BIT STUFFING &'; 'DESTUFFING
    ↳ '};

209
210 % Create CRCButton
211 app.CRCButton = uibutton(app.UIFigure, 'push');
212 app.CRCButton.ButtonPushedFcn = createCallbackFcn(app,
    ↳ @CRCButtonPushed, true);
213 app.CRCButton.BackgroundColor = [0.4824 0.8 0.2863];
214 app.CRCButton.FontSize = 14;
215 app.CRCButton.FontWeight = 'bold';
216 app.CRCButton.Position = [394 153 199 38];
217 app.CRCButton.Text = 'CRC';

218
219 % Create EXITButton
220 app.EXITButton = uibutton(app.UIFigure, 'push');
221 app.EXITButton.ButtonPushedFcn = createCallbackFcn(app,
    ↳ @EXITButtonPushed, true);
222 app.EXITButton.BackgroundColor = [1 0 0];
223 app.EXITButton.FontSize = 14;
224 app.EXITButton.FontWeight = 'bold';
225 app.EXITButton.FontColor = [1 1 1];
226 app.EXITButton.Position = [504 79 89 38];
227 app.EXITButton.Text = 'EXIT';

228
229 % Create DocumentationButton
230 app.DocumentationButton = uibutton(app.UIFigure, 'push');
231 app.DocumentationButton.ButtonPushedFcn = createCallbackFcn(app,
    ↳ @DocumentationButtonPushed, true);
232 app.DocumentationButton.BackgroundColor = [1 0.4118 0.1608];
233 app.DocumentationButton.FontSize = 14;
234 app.DocumentationButton.FontWeight = 'bold';
235 app.DocumentationButton.FontColor = [1 1 1];
236 app.DocumentationButton.Position = [37 79 117 38];
237 app.DocumentationButton.Text = 'Documentation';

238
239 % Create FeedbackButton
240 app.FeedbackButton = uibutton(app.UIFigure, 'push');
241 app.FeedbackButton.ButtonPushedFcn = createCallbackFcn(app,
    ↳ @FeedbackButtonPushed, true);
242 app.FeedbackButton.BackgroundColor = [1 0.4118 0.1608];
243 app.FeedbackButton.FontSize = 14;
244 app.FeedbackButton.FontWeight = 'bold';
245 app.FeedbackButton.FontColor = [1 1 1];
246 app.FeedbackButton.Position = [180 79 117 38];
247 app.FeedbackButton.Text = 'Feedback';

248
249 % Create DeveloperButton
250 app.DeveloperButton = uibutton(app.UIFigure, 'push');
251 app.DeveloperButton.ButtonPushedFcn = createCallbackFcn(app,
    ↳ @DeveloperButtonPushed, true);
252 app.DeveloperButton.BackgroundColor = [1 0.4118 0.1608];
253 app.DeveloperButton.FontSize = 14;
254 app.DeveloperButton.FontWeight = 'bold';

```

```

255         app.DeveloperButton.FontColor = [1 1 1];
256         app.DeveloperButton.Position = [328 79 117 38];
257         app.DeveloperButton.Text = 'Developer';
258
259         % Show the figure after all components are created
260         app.UIFigure.Visible = 'on';
261     end
262 end
263
264 % App creation and deletion
265 methods (Access = public)
266
267     % Construct app
268     function app = home
269
270         % Create UIFigure and components
271         createComponents(app)
272
273         % Register the app with App Designer
274         registerApp(app, app.UIFigure)
275
276         if nargin == 0
277             clear app
278         end
279     end
280
281     % Code that executes before app deletion
282     function delete(app)
283
284         % Delete UIFigure when app is deleted
285         delete(app.UIFigure)
286     end
287 end
288 end

```



## CODE [Line Coding Page]:

```
1 classdef app1 < matlab.apps.AppBase
2
3     % Properties that correspond to app components
4     properties (Access = public)
5         UIFigure                matlab.ui.Figure
6         GridLayout              matlab.ui.container.GridLayout
7         LeftPanel               matlab.ui.container.Panel
8         Image2                  matlab.ui.control.Image
9         Image                   matlab.ui.control.Image
10        LINECODINGSCHMASLabel    matlab.ui.control.Label
11        DifferentialManchesterButton  matlab.ui.control.Button
12        InputEditField           matlab.ui.control.EditField
13        InputEditFieldLabel      matlab.ui.control.Label
14        ManchesterButton         matlab.ui.control.Button
15        PseudoTernaryButton      matlab.ui.control.Button
16        AMIButton               matlab.ui.control.Button
17        NRZLButton              matlab.ui.control.Button
18        NRZIButton              matlab.ui.control.Button
19        RightPanel               matlab.ui.container.Panel
20        DecodedBitsEditField     matlab.ui.control.EditField
21        DecodedBitsLabel        matlab.ui.control.Label
22        UIAxes                   matlab.ui.control.UIAxes
23        Label                    matlab.ui.control.Label
24        DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label    matlab.
            ↳ ui.control.Label
25
26     end
27
28     % Properties that correspond to apps with auto-reflow
29     properties (Access = private)
30         onePanelWidth = 576;
31
32     end
33
34     % Callbacks that handle component events
35     methods (Access = private)
36
37         % Button pushed function: NRZIButton
38         function NRZIButtonPushed(app, event)
39             bitStr = app.InputEditField.Value;
40             bits = str2num(bitStr)
41             %bits = [1 0 1 1 1 0 0 1];
42             bitrate = 1;
43             n = 1000;
44             T = length(bits)/bitrate;
45             N = n * length(bits);
46             dt = T/N;
47
48             t = 0:dt:T;
49             x = zeros(1, length(t));
50             lastbit = 1;
51
52             for i = 1:length(bits)
53                 if bits(i) == 1
54                     x((i-1)*n+1:i*n) = -lastbit;
55                     lastbit = -lastbit;
56                 else
57                     x((i-1)*n+1:i*n) = lastbit;
58                 end
59             end
60             plot(app.UIAxes, t, x, 'Linewidth', 3);
61
62         end
63
64         counter = 0;
65         lastbit = 1;
66         result = zeros(1, length(bits));
67
68         for i = 1:length(t)
69             if t(i) > counter
70                 counter = counter + 1;
71                 if x(i) ~= lastbit
72                     result(counter) = 1;
73                     lastbit = -lastbit;
74                 else
```

```

71         result(counter) = 0;
72     end
73 end
74 end
75
76 result_str = char('0' + result);
77 app.DecodedBitsEditField.Value = string(result_str);
78
79     end
80
81     % Button down function: UIAxes
82     function UIAxesButtonDown(app, event)
83
84     end
85
86     % Button pushed function: NRZLButton
87     function NRZLButtonPushed(app, event)
88 bitStr = app.InputEditField.Value;
89 bits = str2num(bitStr)
90 %bits = [1 0 1 1 1 0 0 1];
91 bitrate = 1;
92 n = 1000;
93 T = length(bits)/bitrate;
94 N = n*length(bits);
95 dt = T/N;
96 t = 0:dt:T;
97 x = zeros(1,length(t));
98 low = -1;
99 high = 1;
100
101 for i = 1:length(bits)
102     if bits(i) == 1
103         x((i-1)*n+1:i*n) = high;
104
105     else
106         x((i-1)*n+1:i*n) = low;
107     end
108 end
109
110 plot(app.UIAxes, x, 'Linewidth', 3);
111
112 counter = 0;
113 high = 1;
114
115 result = zeros(1, length(bits));
116
117 for i = 1:length(t)
118     if t(i) > counter
119         counter = counter + 1;
120         if x(i) ~= high
121             result(counter) = 0;
122
123         else
124             result(counter) = 1;
125         end
126     end
127 end
128 result_str = char('0' + result);
129 app.DecodedBitsEditField.Value = string(result_str);
130
131     end
132
133     % Button pushed function: AMIButton
134     function AMIButtonPushed(app, event)
135 bitStr = app.InputEditField.Value;
136 bits = str2num(bitStr)
137 %bits = [1 0 1 1 1 0 0 1];
138 bitrate = 1;
139 n = 1000;
140 T = length(bits)/bitrate;
141 N = n*length(bits);
142 dt = T/N;
143 t = 0:dt:T;

```

```

144 x = zeros(1,length(t));
145 lastbit = 1;
146
147 for i=1:length(bits)
148     if bits(i)==1
149         x((i-1)*n+1:i*n) = -lastbit;
150         lastbit = -lastbit;
151     end
152 end
153
154 plot(app.UIAxes,t, x, 'Linewidth', 3);
155
156 counter = 0;
157 lastbit = 1;
158
159 for i = 1:length(t)
160     if t(i) > counter
161         counter = counter + 1;
162         if x(i) == -lastbit
163             result(counter) = 1;
164             lastbit = -lastbit;
165         else
166             result(counter) = 0;
167         end
168     end
169 end
170
171 result_str = char('0' + result);
172 app.DecodedBitsEditField.Value = string(result_str);
173
174     end
175
176     % Value changed function: DecodedBitsEditField
177     function DecodedBitsEditFieldValueChanged(app, event)
178         value = app.DecodedBitsEditField.Value;
179
180     end
181
182     % Callback function
183     function NRZIDecodeButtonPushed(app, event)
184
185     end
186
187     % Value changed function: InputEditField
188     function InputEditFieldValueChanged(app, event)
189         value = app.InputEditField.Value;
190
191     end
192
193     % Callback function
194     function ControlSliderValueChanged(app, event)
195         value = app.ControlSlider.Value;
196         % Get the current zoom level from the slider
197
198
199         % Update the UIAxes zoom level
200         % xlim(app.UIAxes, [0, 10] * app.ControlSlider); % Adjust the limits
201         % ↪ accordingly
202         % ylim(app.UIAxes, [0, 10] * app.ControlSlider);
203
204         % Optionally, you can update other properties based on the zoom level
205         % For example, update the axis labels or other visual elements
206
207     end
208
209     % Button pushed function: PseudoTernaryButton
210     function PseudoTernaryButtonPushed(app, event)
211         bitStr = app.InputEditField.Value;
212         bits = str2num(bitStr);
213         %bits = [1 0 1 1 1 0 0 1];
214         bitrate = 1;
215         n = 1000;
216         T = length(bits)/bitrate;

```

```

216 N = n*length(bits);
217 dt = T/N;
218 t = 0:dt:T;
219 x = zeros(1,length(t));
220 lastbit = 1;
221
222 for i=1:length(bits)
223     if bits(i)== 0
224         x((i-1)*n+1:i*n) = -lastbit;
225         lastbit = -lastbit;
226     end
227 end
228
229 plot(app.UIAxes,t, x, 'Linewidth', 3);
230
231 counter = 0;
232 lastbit = 0;
233
234 for i = 1:length(t)
235     if t(i) > counter
236         counter = counter + 1;
237         if x(i) == -lastbit
238             result(counter) = 1;
239             lastbit = -lastbit;
240         else
241             result(counter) = 0;
242         end
243     end
244 end
245
246 result_str = char('0' + result);
247 app.DecodedBitsEditField.Value = string(result_str);
248
249     end
250
251     % Button pushed function: ManchesterButton
252     function ManchesterButtonPushed(app, event)
253 bitStr = app.InputEditField.Value;
254 bits = str2num(bitStr)
255 bitrate = 1;
256 n = 1000;
257 T = length(bits)/bitrate;
258 N = n*length(bits);
259 dt = T/N;
260 t = 0:dt:T;
261 x = zeros(1,length(t));
262
263 for i = 1:length(bits)
264     if bits(i)== 1
265         x((i-1)*n+1:(i-1)*n+n/2) = 1;
266         x((i-1)*n+n/2:i*n) = -1;
267     else
268         x((i-1)*n+1:(i-1)*n+n/2) = -1;
269         x((i-1)*n+n/2:i*n) = 1;
270     end
271 end
272
273 plot(app.UIAxes,t, x, 'Linewidth', 3);
274
275
276 counter = 0;
277 for i = 1:length(t)
278     if t(i) > counter
279         counter = counter + 1;
280
281         if x(i) > 0
282             result(counter) = x(i);
283         else
284             result(counter) = 0;
285         end
286     end
287 end
288

```

```

289 result_str = char('0' + result);
290 app.DecodedBitsEditField.Value = string(result_str);
291
292     end
293
294     % Callback function
295     function IPV4ConversionButtonPushed(app, event)
296
297     end
298
299     % Callback function
300     function HammingCodeButtonPushed(app, event)
301
302     end
303
304     % Callback function
305     function AnalogToDigitalSignalButtonPushed(app, event)
306
307     end
308
309     % Button pushed function: DifferentialManchesterButton
310     function DifferentialManchesterButtonPushed(app, event)
311 bitStr = app.InputEditField.Value;
312 bits = str2num(bitStr)
313 bitrate = 1;
314 n = 1000;
315 T = length(bits)/bitrate;
316 N = n*length(bits);
317 dt = T/N;
318 t = 0:dt:T;
319 x = zeros(1,length(t));
320
321 lastbit = 1;
322 for i=1:length(bits)
323 if bits(i)==0
324 x((i-1)*n+1:(i-1)*n+n/2) = -lastbit;
325 x((i-1)*n+n/2:i*n) = lastbit;
326 else
327 x((i-1)*n+1:(i-1)*n+n/2) = lastbit;
328 x((i-1)*n+n/2:i*n) = -lastbit;
329 lastbit = -lastbit;
330 end
331 end
332
333 plot(app.UIAxes,t, x, 'Linewidth', 3);
334
335
336 count = 0;
337 lastbit = 1;
338 for i = 1:length(t)
339 if t(i)>count
340 count = count + 1;
341 if x(i)==lastbit
342 result(count) = 1;
343 lastbit = -lastbit;
344 else result(count) = 0;
345 end
346 end
347 end
348
349 result_str = char('0' + result);
350 app.DecodedBitsEditField.Value = string(result_str);
351
352     end
353
354     % Changes arrangement of the app based on UIFigure width
355     function updateAppLayout(app, event)
356         currentFigureWidth = app UIFigure.Position(3);
357         if(currentFigureWidth <= app.onePanelWidth)
358             % Change to a 2x1 grid
359             app.GridLayout.RowHeight = {480, 480};
360             app.GridLayout.ColumnWidth = {'1x'};
361             app.RightPanel.Layout.Row = 2;

```

```

362         app.RightPanel.Layout.Column = 1;
363     else
364         % Change to a 1x2 grid
365         app.GridLayout.RowHeight = {'1x'};
366         app.GridLayout.ColumnWidth = {220, '1x'};
367         app.RightPanel.Layout.Row = 1;
368         app.RightPanel.Layout.Column = 2;
369     end
370 end
371
372 % Component initialization
373 methods (Access = private)
374
375     % Create UIFigure and components
376     function createComponents(app)
377
378         % Create UIFigure and hide until all components are created
379         app UIFigure = uifigure('Visible', 'off');
380         app UIFigure.AutoResizeChildren = 'off';
381         app UIFigure.Position = [100 100 640 480];
382         app UIFigure.Name = 'MATLAB App';
383         app UIFigure.SizeChangedFcn = createCallbackFcn(app, @updateAppLayout
384             ↪ , true);
385
386         % Create GridLayout
387         app.GridLayout = uigridlayout(app UIFigure);
388         app.GridLayout.ColumnWidth = {220, '1x'};
389         app.GridLayout.RowHeight = {'1x'};
390         app.GridLayout.ColumnSpacing = 0;
391         app.GridLayout.RowSpacing = 0;
392         app.GridLayout.Padding = [0 0 0 0];
393         app.GridLayout.Scrollable = 'on';
394
395         % Create LeftPanel
396         app.LeftPanel = uipanel(app.GridLayout);
397         app.LeftPanel.Layout.Row = 1;
398         app.LeftPanel.Layout.Column = 1;
399
400         % Create NRZIButton
401         app.NRZIButton = uibutton(app.LeftPanel, 'push');
402         app.NRZIButton.ButtonPushedFcn = createCallbackFcn(app,
403             ↪ @NRZIButtonPushed, true);
404         app.NRZIButton.IconAlignment = 'center';
405         app.NRZIButton.BackgroundColor = [0.4784 0.8 0.2902];
406         app.NRZIButton.FontSize = 14;
407         app.NRZIButton.FontWeight = 'bold';
408         app.NRZIButton.Position = [60 311 100 23];
409         app.NRZIButton.Text = 'NRZ-I';
410
411         % Create NRZLButton
412         app.NRZLButton = uibutton(app.LeftPanel, 'push');
413         app.NRZLButton.ButtonPushedFcn = createCallbackFcn(app,
414             ↪ @NRZLButtonPushed, true);
415         app.NRZLButton.BackgroundColor = [0.4784 0.8 0.2902];
416         app.NRZLButton.FontSize = 14;
417         app.NRZLButton.FontWeight = 'bold';
418         app.NRZLButton.Position = [60 270 100 25];
419         app.NRZLButton.Text = 'NRZ-L';
420
421         % Create AMIButton
422         app.AMIButton = uibutton(app.LeftPanel, 'push');
423         app.AMIButton.ButtonPushedFcn = createCallbackFcn(app,
424             ↪ @AMIButtonPushed, true);
425         app.AMIButton.BackgroundColor = [0.4784 0.8 0.2902];
426         app.AMIButton.FontSize = 14;
427         app.AMIButton.FontWeight = 'bold';
428         app.AMIButton.Position = [60 234 100 25];
429         app.AMIButton.Text = 'AMI';
430
431         % Create PseudoTernaryButton
432         app.PseudoTernaryButton = uibutton(app.LeftPanel, 'push');

```

```

430     app.PseodoTernaryButton.ButtonPushedFcn = createCallbackFcn(app,
        ↳ @PseodoTernaryButtonPushed, true);
431     app.PseodoTernaryButton.BackgroundColor = [0.4784 0.8 0.2902];
432     app.PseodoTernaryButton.FontSize = 14;
433     app.PseodoTernaryButton.FontWeight = 'bold';
434     app.PseodoTernaryButton.Position = [53 193 116 25];
435     app.PseodoTernaryButton.Text = 'PseodoTernary';
436
437     % Create ManchesterButton
438     app.ManchesterButton = uibutton(app.LeftPanel, 'push');
439     app.ManchesterButton.ButtonPushedFcn = createCallbackFcn(app,
        ↳ @ManchesterButtonPushed, true);
440     app.ManchesterButton.BackgroundColor = [0.4784 0.8 0.2902];
441     app.ManchesterButton.FontSize = 14;
442     app.ManchesterButton.FontWeight = 'bold';
443     app.ManchesterButton.Position = [60 150 100 25];
444     app.ManchesterButton.Text = 'Manchester';
445
446     % Create InputEditFieldLabel
447     app.InputEditFieldLabel = uilabel(app.LeftPanel);
448     app.InputEditFieldLabel.HorizontalAlignment = 'right';
449     app.InputEditFieldLabel.Position = [24 34 32 22];
450     app.InputEditFieldLabel.Text = 'Input';
451
452     % Create InputEditField
453     app.InputEditField = uieditfield(app.LeftPanel, 'text');
454     app.InputEditField.ValueChangedFcn = createCallbackFcn(app,
        ↳ @InputEditFieldValueChanged, true);
455     app.InputEditField.Placeholder = '1 0 1 1 1 0 0 1';
456     app.InputEditField.Position = [71 34 123 22];
457
458     % Create DifferentialManchesterButton
459     app.DifferentialManchesterButton = uibutton(app.LeftPanel, 'push');
460     app.DifferentialManchesterButton.ButtonPushedFcn = createCallbackFcn(
        ↳ app, @DifferentialManchesterButtonPushed, true);
461     app.DifferentialManchesterButton.BackgroundColor = [0.4784 0.8
        ↳ 0.2902];
462     app.DifferentialManchesterButton.FontSize = 14;
463     app.DifferentialManchesterButton.FontWeight = 'bold';
464     app.DifferentialManchesterButton.Position = [60 90 100 42];
465     app.DifferentialManchesterButton.Text = {'Differential'; 'Manchester'
        ↳ };
466
467     % Create LINECODINGSCHEMASLabel
468     app.LINECODINGSCHEMASLabel = uilabel(app.LeftPanel);
469     app.LINECODINGSCHEMASLabel.FontSize = 14;
470     app.LINECODINGSCHEMASLabel.FontWeight = 'bold';
471     app.LINECODINGSCHEMASLabel.Position = [25 352 171 22];
472     app.LINECODINGSCHEMASLabel.Text = 'LINE CODING SCHEMAS';
473
474     % Create Image
475     app.Image = uiimage(app.LeftPanel);
476     app.Image.Position = [109 373 100 100];
477     app.Image.ImageSource = 'Logo-PNG-1.png';
478
479     % Create Image2
480     app.Image2 = uiimage(app.LeftPanel);
481     app.Image2.Position = [37 386 66 74];
482     app.Image2.ImageSource = 'images__1_-removebg-preview.png';
483
484     % Create RightPanel
485     app.RightPanel = uipanel(app.GridLayout);
486     app.RightPanel.Layout.Row = 1;
487     app.RightPanel.Layout.Column = 2;
488
489     % Create UIAxes
490     app.UIAxes = uiaxes(app.RightPanel);
491     title(app.UIAxes, 'OUTPUT FIGURE')
492     xlabel(app.UIAxes, 'X')
493     ylabel(app.UIAxes, 'Y')
494     zlabel(app.UIAxes, 'Z')
495     app.UIAxes.ButtonDownFcn = createCallbackFcn(app, @UIAxesButtonDown,
        ↳ true);

```

```

496     app.UIAxes.Position = [16 74 389 236];
497
498     % Create DecodedBitsLabel
499     app.DecodedBitsLabel = uilabel(app.RightPanel);
500     app.DecodedBitsLabel.HorizontalAlignment = 'right';
501     app.DecodedBitsLabel.FontSize = 18;
502     app.DecodedBitsLabel.Position = [13 228 107 23];
503     app.DecodedBitsLabel.Text = 'DecodedBits';
504
505     % Create DecodedBitsEditField
506     app.DecodedBitsEditField = uieditfield(app.RightPanel, 'text');
507     app.DecodedBitsEditField.ValueChangedFcn = createCallbackFcn(app,
508         ↪ @DecodedBitsEditFieldValueChanged, true);
509     app.DecodedBitsEditField.Editable = 'off';
510     app.DecodedBitsEditField.HorizontalAlignment = 'center';
511     app.DecodedBitsEditField.FontSize = 18;
512     app.DecodedBitsEditField.Placeholder = 'Decoded Bits will be shown
513         ↪ here';
514     app.DecodedBitsEditField.Position = [135 224 238 27];
515
516     % Create
517     ↪ DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label
518     app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label =
519     ↪ uilabel(app.UIFigure);
520     app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
521     ↪ HorizontalAlignment = 'center';
522     app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
523     ↪ FontSize = 8;
524     app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
525     ↪ FontWeight = 'bold';
526     app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
527     ↪ Position = [243 3 348 22];
528     app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
529     ↪ Text = 'DEVELOPED BY: MD SAIFUL ISLAM RIMON (213002039) &
530     ↪ NAZMUN NAHAR (191000000)';
531
532     % Create Label
533     app.Label = uilabel(app.UIFigure);
534     app.Label.HorizontalAlignment = 'center';
535     app.Label.FontSize = 8;
536     app.Label.FontWeight = 'bold';
537     app.Label.Position = [248 28 333 22];
538     app.Label.Text = 'PROJECT SUPERVISOR: RUSMITA HALIM CHAITY, LECTURER
539     ↪ , DEPT. OF CSE, GUB';
540
541     % Show the figure after all components are created
542     app.UIFigure.Visible = 'on';
543 end
544
545 % App creation and deletion
546 methods (Access = public)
547
548     % Construct app
549     function app = app1
550
551         % Create UIFigure and components
552         createComponents(app)
553
554         % Register the app with App Designer
555         registerApp(app, app.UIFigure)
556
557         if nargin == 0
558             clear app
559         end
560     end
561
562     % Code that executes before app deletion
563     function delete(app)
564
565         % Delete UIFigure when app is deleted
566         delete(app.UIFigure)
567     end

```



```
558 |         end
559 |     end
```

## CODE [Hamming Encoding & Decoding Page]:

```

1  classdef hammingcode < matlab.apps.AppBase
2
3      % Properties that correspond to app components
4      properties (Access = public)
5          UIFigure                matlab.ui.Figure
6          Label1                  matlab.ui.control.Label
7          DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label  matlab.
            ↳ ui.control.Label
8          Image3                  matlab.ui.control.Image
9          Image2                  matlab.ui.control.Image
10         NUMBEROFFPARITYBITSEditField  matlab.ui.control.EditField
11         NUMBEROFFPARITYBITSEditFieldLabel  matlab.ui.control.Label
12         LENGTHOFFHAMMINGCODEEditField  matlab.ui.control.EditField
13         LENGTHOFFHAMMINGCODEEditFieldLabel  matlab.ui.control.Label
14         CORRECTEDBITSTRINGEditField  matlab.ui.control.EditField
15         CORRECTEDBITSTRINGEditFieldLabel  matlab.ui.control.Label
16         POSITIONOFFERROREditField  matlab.ui.control.EditField
17         POSITIONOFFERROREditFieldLabel  matlab.ui.control.Label
18         P4EditField_2            matlab.ui.control.EditField
19         P4EditField_2Label       matlab.ui.control.Label
20         P2EditField_4            matlab.ui.control.EditField
21         P2EditField_4Label       matlab.ui.control.Label
22         P1EditField_4            matlab.ui.control.EditField
23         P1EditField_4Label       matlab.ui.control.Label
24         ERRORDETECTIONANDCORRECTIONButton  matlab.ui.control.Button
25         ERRORDETECTIONLabel      matlab.ui.control.Label
26         RECEIVEDBITSTRINGEditField  matlab.ui.control.EditField
27         RECEIVEDBITSTRINGEditFieldLabel  matlab.ui.control.Label
28         SENDEDBITSTRINGEditField  matlab.ui.control.EditField
29         SENDEDBITSTRINGEditFieldLabel  matlab.ui.control.Label
30         P4EditField              matlab.ui.control.EditField
31         P4EditFieldLabel          matlab.ui.control.Label
32         P2EditField_3            matlab.ui.control.EditField
33         P2EditField_3Label       matlab.ui.control.Label
34         P1EditField_3            matlab.ui.control.EditField
35         P1EditField_3Label       matlab.ui.control.Label
36         GENERATESENDINGBITSTRINGButton  matlab.ui.control.Button
37         BITHAMMINGCODEPARITYBITSANDERRORDETECTIONLabel  matlab.ui.control.Label
38         BITSTRINGEditField       matlab.ui.control.EditField
39         Image                     matlab.ui.control.Image
40         BITSTRINGEditFieldLabel  matlab.ui.control.Label
41     end
42
43     properties (Access = public)
44         n
45         nh % Description
46         displayedValue
47     end
48
49
50
51     % Callbacks that handle component events
52     methods (Access = private)
53
54         % Value changed function: P1EditField_3
55         function P1EditField_3ValueChanged(app, event)
56             value = app.P1EditField_3.Value;
57
58         end
59
60         % Value changed function: P2EditField_3
61         function P2EditField_3ValueChanged(app, event)
62             value = app.P2EditField_3.Value;
63
64         end
65
66         % Value changed function: P4EditField
67         function P4EditFieldValueChanged(app, event)
68             value = app.P4EditField.Value;
69
70         end

```

```

71
72 % Value changed function: SENDEDBITSTRINGEditField
73 function SENDEDBITSTRINGEditFieldValueChanged(app, event)
74     value = app.SENDEDBITSTRINGEditField.Value;
75
76 end
77
78 % Value changed function: P1EditField_4
79 function P1EditField_4ValueChanged(app, event)
80     value = app.P1EditField_4.Value;
81
82 end
83
84 % Value changed function: P2EditField_4
85 function P2EditField_4ValueChanged(app, event)
86     value = app.P2EditField_4.Value;
87
88 end
89
90 % Value changed function: P4EditField_2
91 function P4EditField_2ValueChanged(app, event)
92     value = app.P4EditField_2.Value;
93
94 end
95
96 % Value changed function: POSITIONOFERROREditField
97 function POSITIONOFERROREditFieldValueChanged(app, event)
98     value = app.POSITIONOFERROREditField.Value;
99
100 end
101
102 % Value changed function: CORRECTEDBITSTRINGEditField
103 function CORRECTEDBITSTRINGEditFieldValueChanged(app, event)
104     value = app.CORRECTEDBITSTRINGEditField.Value;
105
106 end
107
108 % Button pushed function: GENERATESENDINGBITSTRINGButton
109 function GENERATESENDINGBITSTRINGButtonPushed(app, event)
110     % Convert the entire code to Matlab code
111
112 maxp = 5;
113 a = zeros(1, 50);
114 temp = zeros(1, 70);
115 temp2 = zeros(1, 70);
116 t = 0;
117 i = 0;
118 j = 0;
119 k = 0;
120 nd = 0;
121 n = 0;
122 app.n = n;
123 nh = 0;
124 app.nh = nh;
125 sum = 0;
126 pos = 0;
127
128 nd = 4;
129 a = app.BITSTRINGEditField.Value;
130 a = str2num(a);
131
132 for i = 1:nd
133     for k = 0:maxp-1
134         t = 2^k - 1;
135         if j == t
136             temp(j+1) = 0;
137             j = j + 1;
138         end
139     end
140     temp(j+1) = a(i);
141     j = j + 1;
142 end
143 app.nh = j;

```

```

144 app.LENGTHOFHAMMINGCODEEditField.Value = string(app.nh);
145
146 app.n = app.nh - nd;
147 app.NUMBEROFPARITYBITSEditField.Value = string(app.n);
148
149
150 b = zeros(1, app.n);
151 m = app.n - 1;
152 for k = 0:app.n-1
153     t = 2^k - 1;
154     i = t;
155     while i < app.nh
156         for j = 0:t
157             sum = sum + temp(i+1);
158             i = i + 1;
159             if i >= app.nh
160                 break;
161             end
162         end
163         if i >= app.nh
164             break;
165         end
166         for j = 0:t
167             i = i + 1;
168             if i >= app.nh
169                 break;
170             end
171         end
172         if i >= app.nh
173             break;
174         end
175     end
176     temp(t+1) = mod(sum, 2);
177     sum = 0;
178
179     if(t+1 == 1)
180         app.P1EditField_3.Value = string(temp(t+1));
181     end
182
183     if(t+1 == 2)
184         app.P2EditField_3.Value = string(temp(t+1));
185     end
186
187     if(t+1 == 4)
188         app.P4EditField.Value = string(temp(t+1));
189     end
190 end
191
192 %fprintf('\nHamming code: Sender side: ');
193 for i = 1:app.nh
194     app.SENDEDBITSTRINGEditField.Value = app.SENDEDBITSTRINGEditField.Value +
195         ↪ string(temp(i)) + " ";
196     drawnow;
197 end
198 app.displayedValue = app.SENDEDBITSTRINGEditField.Value;
199
200
201
202     end
203
204     % Button pushed function: ERRORDETECTIONANDCORRECTIONButton
205     function ERRORDETECTIONANDCORRECTIONButtonPushed(app, event)
206
207         %fprintf('\nHamming code: Receiver side: ');
208         a = app.RECEIVEDBITSTRINGEditField.Value;
209         tempCellArray = strsplit(a, ' ');
210         temp2 = str2double(tempCellArray);
211
212         sum = 0;
213         n = app.n;
214         nh = app.nh;
215         m = n;

```

```

216 b = zeros(1, n);
217 pos = 0;
218
219 for k = 0:n-1
220     t = 2^k - 1;
221
222     for i = t+1:nh %t+1
223         for j = 1:t+1
224             sum = sum + temp2(i);
225             i = i + 1;
226             if i >= nh
227                 break;
228             end
229         end
230
231         if i >= nh
232             break;
233         end
234
235         for j = 1:t+1
236             i = i + 1;
237             if i >= nh
238                 break;
239             end
240         end
241
242         if i >= nh
243             break;
244         end
245     end
246     b(m) = mod(sum, 2);
247     sum = 0;
248     %fprintf('P%d: %d\n', t+1, b(m));
249     if(t+1 == 1)
250         app.P1EditField_4.Value = string(b(m));
251     end
252
253     if(t+1 == 2)
254         app.P2EditField_4.Value = string(b(m));
255     end
256
257     if(t+1 == 4)
258         app.P4EditField_2.Value = string(b(m));
259     end
260     m = m - 1;
261 end
262
263 for m = 1:n
264     pos = pos + b(n-m+1) * 2^(m-1);
265 end
266 %fprintf('Position of Error: %d\n', pos);
267
268 app.POSITIONOFERROREditField.Value = string(pos);
269
270 if temp2(pos) == 0
271     temp2(pos) = 1;
272 else
273     temp2(pos) = 0;
274 end
275
276 %fprintf('\nHamming code: Receiver side: Error Corrected: ');
277
278 % for i = 1:nh
279 %     fprintf('%d ', temp2(i));
280 % end
281
282 % for i = 1:nh
283 % app.CORRECTEDBITSTRINGEditField.Value = app.CORRECTEDBITSTRINGEditField.
284 %     Value + string(temp2(i)) + " ";
285 % drawnow;
286 % end
287 app.CORRECTEDBITSTRINGEditField.Value = app.displayedValue;

```

```

288
289
290 %fprintf('\n-----\n');
291 end
292
293 % Value changed function: BITSTRINGEditField
294 function BITSTRINGEditFieldValueChanged(app, event)
295     value = app.BITSTRINGEditField.Value;
296
297 end
298
299 % Value changed function: LENGTHOFHAMMINGCODEEditField
300 function LENGTHOFHAMMINGCODEEditFieldValueChanged(app, event)
301     value = app.LENGTHOFHAMMINGCODEEditField.Value;
302
303 end
304
305 % Value changed function: NUMBEROFPARITYBITSEditField
306 function NUMBEROFPARITYBITSEditFieldValueChanged(app, event)
307     value = app.NUMBEROFPARITYBITSEditField.Value;
308
309 end
310
311 % Value changed function: RECEIVEDBITSTRINGEditField
312 function RECEIVEDBITSTRINGEditFieldValueChanged(app, event)
313     value = app.RECEIVEDBITSTRINGEditField.Value;
314
315 end
316 end
317
318 % Component initialization
319 methods (Access = private)
320
321 % Create UIFigure and components
322 function createComponents(app)
323
324     % Create UIFigure and hide until all components are created
325     app.UIFigure = uifigure('Visible', 'off');
326     app.UIFigure.Position = [100 100 640 480];
327     app.UIFigure.Name = 'MATLAB App';
328
329     % Create BITSTRINGEditFieldLabel
330     app.BITSTRINGEditFieldLabel = uilabel(app.UIFigure);
331     app.BITSTRINGEditFieldLabel.HorizontalAlignment = 'right';
332     app.BITSTRINGEditFieldLabel.Position = [33 373 72 22];
333     app.BITSTRINGEditFieldLabel.Text = 'BIT STRING';
334
335     % Create Image
336     app.Image = uiimage(app.UIFigure);
337     app.Image.Position = [-171 -102 979 647];
338     app.Image.ImageSource = 'bg.jpg';
339
340     % Create BITSTRINGEditField
341     app.BITSTRINGEditField = uieditfield(app.UIFigure, 'text');
342     app.BITSTRINGEditField.ValueChangedFcn = createCallbackFcn(app,
343         ↳ @BITSTRINGEditFieldValueChanged, true);
344     app.BITSTRINGEditField.Placeholder = '1 0 1 0';
345     app.BITSTRINGEditField.Position = [120 373 229 22];
346
347     % Create BITHAMMINGCODEPARITYBITSANDERRORDETECTIONLabel
348     app.BITHAMMINGCODEPARITYBITSANDERRORDETECTIONLabel = uilabel(app.
349         ↳ UIFigure);
350     app.BITHAMMINGCODEPARITYBITSANDERRORDETECTIONLabel.
351         ↳ HorizontalAlignment = 'center';
352     app.BITHAMMINGCODEPARITYBITSANDERRORDETECTIONLabel.WordWrap = 'on';
353     app.BITHAMMINGCODEPARITYBITSANDERRORDETECTIONLabel.FontSize = 14;
354     app.BITHAMMINGCODEPARITYBITSANDERRORDETECTIONLabel.FontWeight = 'bold
355         ↳ ';
356     app.BITHAMMINGCODEPARITYBITSANDERRORDETECTIONLabel.Position = [140
357         ↳ 421 362 34];
358     app.BITHAMMINGCODEPARITYBITSANDERRORDETECTIONLabel.Text = '4 BIT
359         ↳ HAMMING CODE PARITY BITS AND ERROR DETECTION';

```

```

355 % Create GENERATESENDINGBITSTRINGButton
356 app.GENERATESENDINGBITSTRINGButton = uibutton(app.UIFigure, 'push');
357 app.GENERATESENDINGBITSTRINGButton.ButtonPushedFcn =
    ↳ createCallbackFcn(app, @GENERATESENDINGBITSTRINGButtonPushed,
    ↳ true);
358 app.GENERATESENDINGBITSTRINGButton.BackgroundColor = [0.4 0.8
    ↳ 0.1216];
359 app.GENERATESENDINGBITSTRINGButton.Position = [370 373 245 23];
360 app.GENERATESENDINGBITSTRINGButton.Text = 'GENERATE SENDING BIT
    ↳ STRING';
361
362 % Create P1EditField_3Label
363 app.P1EditField_3Label = uilabel(app.UIFigure);
364 app.P1EditField_3Label.HorizontalAlignment = 'right';
365 app.P1EditField_3Label.Position = [63 310 25 22];
366 app.P1EditField_3Label.Text = 'P1';
367
368 % Create P1EditField_3
369 app.P1EditField_3 = uieditfield(app.UIFigure, 'text');
370 app.P1EditField_3.ValueChangedFcn = createCallbackFcn(app,
    ↳ @P1EditField_3ValueChanged, true);
371 app.P1EditField_3.Eitable = 'off';
372 app.P1EditField_3.Position = [103 310 38 22];
373
374 % Create P2EditField_3Label
375 app.P2EditField_3Label = uilabel(app.UIFigure);
376 app.P2EditField_3Label.HorizontalAlignment = 'right';
377 app.P2EditField_3Label.Position = [63 268 25 22];
378 app.P2EditField_3Label.Text = 'P2';
379
380 % Create P2EditField_3
381 app.P2EditField_3 = uieditfield(app.UIFigure, 'text');
382 app.P2EditField_3.ValueChangedFcn = createCallbackFcn(app,
    ↳ @P2EditField_3ValueChanged, true);
383 app.P2EditField_3.Eitable = 'off';
384 app.P2EditField_3.Position = [103 268 38 22];
385
386 % Create P4EditFieldLabel
387 app.P4EditFieldLabel = uilabel(app.UIFigure);
388 app.P4EditFieldLabel.HorizontalAlignment = 'right';
389 app.P4EditFieldLabel.Position = [63 230 25 22];
390 app.P4EditFieldLabel.Text = 'P4';
391
392 % Create P4EditField
393 app.P4EditField = uieditfield(app.UIFigure, 'text');
394 app.P4EditField.ValueChangedFcn = createCallbackFcn(app,
    ↳ @P4EditFieldValueChanged, true);
395 app.P4EditField.Eitable = 'off';
396 app.P4EditField.Position = [103 230 38 22];
397
398 % Create SENDEDBITSTRINGEditFieldLabel
399 app.SENDEDBITSTRINGEditFieldLabel = uilabel(app.UIFigure);
400 app.SENDEDBITSTRINGEditFieldLabel.HorizontalAlignment = 'right';
401 app.SENDEDBITSTRINGEditFieldLabel.Position = [253 241 129 22];
402 app.SENDEDBITSTRINGEditFieldLabel.Text = 'SENDED BIT STRING';
403
404 % Create SENDEDBITSTRINGEditField
405 app.SENDEDBITSTRINGEditField = uieditfield(app.UIFigure, 'text');
406 app.SENDEDBITSTRINGEditField.ValueChangedFcn = createCallbackFcn(app,
    ↳ @SENDEDBITSTRINGEditFieldValueChanged, true);
407 app.SENDEDBITSTRINGEditField.Eitable = 'off';
408 app.SENDEDBITSTRINGEditField.Position = [389 241 172 22];
409
410 % Create RECEIVEDBITSTRINGEditFieldLabel
411 app.RECEIVEDBITSTRINGEditFieldLabel = uilabel(app.UIFigure);
412 app.RECEIVEDBITSTRINGEditFieldLabel.HorizontalAlignment = 'right';
413 app.RECEIVEDBITSTRINGEditFieldLabel.Position = [33 153 137 22];
414 app.RECEIVEDBITSTRINGEditFieldLabel.Text = 'RECEIVED BIT STRING';
415
416 % Create RECEIVEDBITSTRINGEditField
417 app.RECEIVEDBITSTRINGEditField = uieditfield(app.UIFigure, 'text');
418 app.RECEIVEDBITSTRINGEditField.ValueChangedFcn = createCallbackFcn(
    ↳ app, @RECEIVEDBITSTRINGEditFieldValueChanged, true);

```

```

419 app.RECEIVEDBITSTRINGEditField.Placeholder = '1 0 0 1 0 1 1';
420 app.RECEIVEDBITSTRINGEditField.Position = [177 153 172 22];
421
422 % Create ERRORDETECTIONLabel
423 app.ERRORDETECTIONLabel = uilabel(app.UIFigure);
424 app.ERRORDETECTIONLabel.HorizontalAlignment = 'center';
425 app.ERRORDETECTIONLabel.FontSize = 14;
426 app.ERRORDETECTIONLabel.FontWeight = 'bold';
427 app.ERRORDETECTIONLabel.Position = [140 193 362 22];
428 app.ERRORDETECTIONLabel.Text = 'ERROR DETECTION';
429
430 % Create ERRORDETECTIONANDCORRECTIONButton
431 app.ERRORDETECTIONANDCORRECTIONButton = uibutton(app.UIFigure, 'push'
    ↳ );
432 app.ERRORDETECTIONANDCORRECTIONButton.ButtonPushedFcn =
    ↳ createCallbackFcn(app, @ERRORDETECTIONANDCORRECTIONButtonPushed
    ↳ , true);
433 app.ERRORDETECTIONANDCORRECTIONButton.BackgroundColor = [0.4 0.8
    ↳ 0.1216];
434 app.ERRORDETECTIONANDCORRECTIONButton.Position = [370 153 245 23];
435 app.ERRORDETECTIONANDCORRECTIONButton.Text = 'ERROR DETECTION AND
    ↳ CORRECTION';
436
437 % Create P1EditField_4Label
438 app.P1EditField_4Label = uilabel(app.UIFigure);
439 app.P1EditField_4Label.HorizontalAlignment = 'right';
440 app.P1EditField_4Label.Position = [63 123 25 22];
441 app.P1EditField_4Label.Text = 'P1';
442
443 % Create P1EditField_4
444 app.P1EditField_4 = uieditfield(app.UIFigure, 'text');
445 app.P1EditField_4.ValueChangedFcn = createCallbackFcn(app,
    ↳ @P1EditField_4ValueChanged, true);
446 app.P1EditField_4.Eitable = 'off';
447 app.P1EditField_4.Position = [103 123 38 22];
448
449 % Create P2EditField_4Label
450 app.P2EditField_4Label = uilabel(app.UIFigure);
451 app.P2EditField_4Label.HorizontalAlignment = 'right';
452 app.P2EditField_4Label.Position = [63 90 25 22];
453 app.P2EditField_4Label.Text = 'P2';
454
455 % Create P2EditField_4
456 app.P2EditField_4 = uieditfield(app.UIFigure, 'text');
457 app.P2EditField_4.ValueChangedFcn = createCallbackFcn(app,
    ↳ @P2EditField_4ValueChanged, true);
458 app.P2EditField_4.Eitable = 'off';
459 app.P2EditField_4.Position = [103 90 38 22];
460
461 % Create P4EditField_2Label
462 app.P4EditField_2Label = uilabel(app.UIFigure);
463 app.P4EditField_2Label.HorizontalAlignment = 'right';
464 app.P4EditField_2Label.Position = [63 58 25 22];
465 app.P4EditField_2Label.Text = 'P4';
466
467 % Create P4EditField_2
468 app.P4EditField_2 = uieditfield(app.UIFigure, 'text');
469 app.P4EditField_2.ValueChangedFcn = createCallbackFcn(app,
    ↳ @P4EditField_2ValueChanged, true);
470 app.P4EditField_2.Eitable = 'off';
471 app.P4EditField_2.Position = [103 58 38 22];
472
473 % Create POSITIONOFERROREditFieldLabel
474 app.POSITIONOFERROREditFieldLabel = uilabel(app.UIFigure);
475 app.POSITIONOFERROREditFieldLabel.Position = [277 107 159 22];
476 app.POSITIONOFERROREditFieldLabel.Text = 'POSITION OF ERROR';
477
478 % Create POSITIONOFERROREditField
479 app.POSITIONOFERROREditField = uieditfield(app.UIFigure, 'text');
480 app.POSITIONOFERROREditField.ValueChangedFcn = createCallbackFcn(app,
    ↳ @POSITIONOFERROREditFieldValueChanged, true);
481 app.POSITIONOFERROREditField.Eitable = 'off';
482 app.POSITIONOFERROREditField.Position = [415 107 170 22];

```



```

483
484 % Create CORRECTEDBITSTRINGEditFieldLabel
485 app.CORRECTEDBITSTRINGEditFieldLabel = uilabel(app.UIFigure);
486 app.CORRECTEDBITSTRINGEditFieldLabel.HorizontalAlignment = 'right';
487 app.CORRECTEDBITSTRINGEditFieldLabel.Position = [255 58 151 22];
488 app.CORRECTEDBITSTRINGEditFieldLabel.Text = 'CORRECTED BIT STRING';
489
490 % Create CORRECTEDBITSTRINGEditField
491 app.CORRECTEDBITSTRINGEditField = uieditfield(app.UIFigure, 'text');
492 app.CORRECTEDBITSTRINGEditField.ValueChangedFcn = createCallbackFcn(
    ↳ app, @CORRECTEDBITSTRINGEditFieldValueChanged, true);
493 app.CORRECTEDBITSTRINGEditField.Eutable = 'off';
494 app.CORRECTEDBITSTRINGEditField.Position = [413 58 172 22];
495
496 % Create LENGTHOFHAMMINGCODEEditFieldLabel
497 app.LENGTHOFHAMMINGCODEEditFieldLabel = uilabel(app.UIFigure);
498 app.LENGTHOFHAMMINGCODEEditFieldLabel.HorizontalAlignment = 'right';
499 app.LENGTHOFHAMMINGCODEEditFieldLabel.Position = [209 283 173 22];
500 app.LENGTHOFHAMMINGCODEEditFieldLabel.Text = 'LENGTH OF HAMMING CODE'
    ↳ ;
501
502 % Create LENGTHOFHAMMINGCODEEditField
503 app.LENGTHOFHAMMINGCODEEditField = uieditfield(app.UIFigure, 'text');
504 app.LENGTHOFHAMMINGCODEEditField.ValueChangedFcn = createCallbackFcn(
    ↳ app, @LENGTHOFHAMMINGCODEEditFieldValueChanged, true);
505 app.LENGTHOFHAMMINGCODEEditField.Eutable = 'off';
506 app.LENGTHOFHAMMINGCODEEditField.Position = [389 283 172 22];
507
508 % Create NUMBEROFFPARITYBITSEditFieldLabel
509 app.NUMBEROFFPARITYBITSEditFieldLabel = uilabel(app.UIFigure);
510 app.NUMBEROFFPARITYBITSEditFieldLabel.HorizontalAlignment = 'right';
511 app.NUMBEROFFPARITYBITSEditFieldLabel.Position = [227 316 153 22];
512 app.NUMBEROFFPARITYBITSEditFieldLabel.Text = 'NUMBER OF PARITY BITS';
513
514 % Create NUMBEROFFPARITYBITSEditField
515 app.NUMBEROFFPARITYBITSEditField = uieditfield(app.UIFigure, 'text');
516 app.NUMBEROFFPARITYBITSEditField.ValueChangedFcn = createCallbackFcn(
    ↳ app, @NUMBEROFFPARITYBITSEditFieldValueChanged, true);
517 app.NUMBEROFFPARITYBITSEditField.Eutable = 'off';
518 app.NUMBEROFFPARITYBITSEditField.Position = [387 316 172 22];
519
520 % Create Image2
521 app.Image2 = uiimage(app.UIFigure);
522 app.Image2.Position = [33 394 71 80];
523 app.Image2.ImageSource = 'images__1_-removebg-preview.png';
524
525 % Create Image3
526 app.Image3 = uiimage(app.UIFigure);
527 app.Image3.Position = [538 394 93 80];
528 app.Image3.ImageSource = 'Logo-PNG-1.png';
529
530 % Create
    ↳ DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label
531 app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label =
    ↳ uilabel(app.UIFigure);
532 app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
    ↳ HorizontalAlignment = 'center';
533 app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
    ↳ FontWeight = 'bold';
534 app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
    ↳ Position = [60 3 521 22];
535 app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
    ↳ Text = 'DEVELOPED BY: MD SAIFUL ISLAM RIMON (213002039) &
    ↳ NAZMUN NAHAR (191000000)';
536
537 % Create Label
538 app.Label = uilabel(app.UIFigure);
539 app.Label.HorizontalAlignment = 'center';
540 app.Label.FontWeight = 'bold';
541 app.Label.Position = [68 28 499 22];
542 app.Label.Text = 'PROJECT SUPERVISOR: RUSMITA HALIM CHAITY, LECTURER
    ↳ , DEPT. OF CSE, GUB';
543

```

```

544         % Show the figure after all components are created
545         app UIFigure.Visible = 'on';
546     end
547 end
548
549 % App creation and deletion
550 methods (Access = public)
551
552     % Construct app
553     function app = hammingcode
554
555         % Create UIFigure and components
556         createComponents(app)
557
558         % Register the app with App Designer
559         registerApp(app, app UIFigure)
560
561         if nargin == 0
562             clear app
563         end
564     end
565
566     % Code that executes before app deletion
567     function delete(app)
568
569         % Delete UIFigure when app is deleted
570         delete(app UIFigure)
571     end
572 end
573 end

```

## CODE [Analog to Digital Signal Conversion Page]:

```
1 classdef ADC < matlab.apps.AppBase
2
3     % Properties that correspond to app components
4     properties (Access = public)
5         UIFigure          matlab.ui.Figure
6         Image3             matlab.ui.control.Image
7         Image2             matlab.ui.control.Image
8         Label              matlab.ui.control.Label
9         DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label  matlab.
            ↳ ui.control.Label
10        BitCountEditField  matlab.ui.control.EditField
11        BitCountEditFieldLabel  matlab.ui.control.Label
12        FrequencyEditField  matlab.ui.control.EditField
13        FrequencyEditFieldLabel  matlab.ui.control.Label
14        CONVERTButton      matlab.ui.control.Button
15        UIAxes             matlab.ui.control.UIAxes
16    end
17
18    % Callbacks that handle component events
19    methods (Access = private)
20
21        % Button pushed function: CONVERTButton
22        function CONVERTButtonPushed(app, event)
23            cla(app.UIAxes);
24            f = str2num(app.FrequencyEditField.Value) ;
25            n = str2num(app.BitCountEditField.Value) ;
26            q = f/2^(n-1) ;
27            t = 0:0.1:f ;
28            y = abs((f/2) * sin(t));
29
30            x0 = fix(y/q);
31
32            y0 = dec2bin(x0,n) ;
33            y1 = x0 * q;
34            plot(app.UIAxes,t, y, 'r', 'Linewidth', 3);
35            %plot (t,y,'r') ;
36            hold(app.UIAxes, 'on');
37            %plot (t, y1,'b') ;
38            plot(app.UIAxes,t, y1, 'b', 'Linewidth', 3);
39            end
40
41        % Value changed function: FrequencyEditField
42        function FrequencyEditFieldValueChanged(app, event)
43            value = app.FrequencyEditField.Value;
44
45        end
46
47        % Value changed function: BitCountEditField
48        function BitCountEditFieldValueChanged(app, event)
49            value = app.BitCountEditField.Value;
50
51        end
52    end
53
54    % Component initialization
55    methods (Access = private)
56
57        % Create UIFigure and components
58        function createComponents(app)
59
60            % Create UIFigure and hide until all components are created
61            app.UIFigure = uifigure('Visible', 'off');
62            app.UIFigure.Position = [100 100 640 480];
63            app.UIFigure.Name = 'MATLAB App';
64
65            % Create UIAxes
66            app.UIAxes = uiaxes(app.UIFigure);
67            title(app.UIAxes, 'ANALOG TO DIGITAL CONVERSION')
68            xlabel(app.UIAxes, 'X')
69            ylabel(app.UIAxes, 'Y')
70            zlabel(app.UIAxes, 'Z')
```

```

71 app.UIAxes.Position = [75 140 478 310];
72
73 % Create CONVERTButton
74 app.CONVERTButton = uibutton(app.UIFigure, 'push');
75 app.CONVERTButton.ButtonPushedFcn = createCallbackFcn(app,
    ↳ @CONVERTButtonPushed, true);
76 app.CONVERTButton.BackgroundColor = [0.4784 0.8 0.2902];
77 app.CONVERTButton.FontSize = 14;
78 app.CONVERTButton.FontWeight = 'bold';
79 app.CONVERTButton.Position = [481 94 100 25];
80 app.CONVERTButton.Text = 'CONVERT';
81
82 % Create FrequencyEditFieldLabel
83 app.FrequencyEditFieldLabel = uilabel(app.UIFigure);
84 app.FrequencyEditFieldLabel.HorizontalAlignment = 'right';
85 app.FrequencyEditFieldLabel.Position = [48 95 62 22];
86 app.FrequencyEditFieldLabel.Text = 'Frequency';
87
88 % Create FrequencyEditField
89 app.FrequencyEditField = uieditfield(app.UIFigure, 'text');
90 app.FrequencyEditField.ValueChangedFcn = createCallbackFcn(app,
    ↳ @FrequencyEditFieldValueChanged, true);
91 app.FrequencyEditField.Placeholder = '10';
92 app.FrequencyEditField.Position = [125 95 100 22];
93
94 % Create BitCountEditFieldLabel
95 app.BitCountEditFieldLabel = uilabel(app.UIFigure);
96 app.BitCountEditFieldLabel.HorizontalAlignment = 'right';
97 app.BitCountEditFieldLabel.Position = [271 95 54 22];
98 app.BitCountEditFieldLabel.Text = 'Bit Count';
99
100 % Create BitCountEditField
101 app.BitCountEditField = uieditfield(app.UIFigure, 'text');
102 app.BitCountEditField.ValueChangedFcn = createCallbackFcn(app,
    ↳ @BitCountEditFieldValueChanged, true);
103 app.BitCountEditField.Placeholder = '3';
104 app.BitCountEditField.Position = [340 95 100 22];
105
106 % Create
    ↳ DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label
107 app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label =
    ↳ uilabel(app.UIFigure);
108 app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
    ↳ HorizontalAlignment = 'center';
109 app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
    ↳ FontWeight = 'bold';
110 app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
    ↳ Position = [60 3 521 22];
111 app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
    ↳ Text = 'DEVELOPED BY: MD SAIFUL ISLAM RIMON (213002039) &
    ↳ NAZMUN NAHAR (191000000)';
112
113 % Create Label
114 app.Label = uilabel(app.UIFigure);
115 app.Label.HorizontalAlignment = 'center';
116 app.Label.FontWeight = 'bold';
117 app.Label.Position = [68 28 499 22];
118 app.Label.Text = 'PROJECT SUPERVISOR: RUSMITA HALIM CHAITY, LECTURER
    ↳ , DEPT. OF CSE, GUB';
119
120 % Create Image2
121 app.Image2 = uiimage(app.UIFigure);
122 app.Image2.Position = [22 381 71 80];
123 app.Image2.ImageSource = 'images__1_-removebg-preview.png';
124
125 % Create Image3
126 app.Image3 = uiimage(app.UIFigure);
127 app.Image3.Position = [548 381 93 80];
128 app.Image3.ImageSource = 'Logo-PNG-1.png';
129
130 % Show the figure after all components are created
131 app.UIFigure.Visible = 'on';
132 end

```

```

133     end
134
135     % App creation and deletion
136     methods (Access = public)
137
138         % Construct app
139         function app = ADC
140
141             % Create UIFigure and components
142             createComponents(app)
143
144             % Register the app with App Designer
145             registerApp(app, app.UIFigure)
146
147             if nargin == 0
148                 clear app
149             end
150         end
151
152         % Code that executes before app deletion
153         function delete(app)
154
155             % Delete UIFigure when app is deleted
156             delete(app.UIFigure)
157         end
158     end
159 end

```

## CODE [Character Stuffing & Destuffing Page]:

```
1 classdef CharacterStuffingDestuffing < matlab.apps.AppBase
2
3     % Properties that correspond to app components
4     properties (Access = public)
5         UIFigure          matlab.ui.Figure
6         Image3             matlab.ui.control.Image
7         Image2             matlab.ui.control.Image
8         Label              matlab.ui.control.Label
9         DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label  matlab.
           ↳ ui.control.Label
10        CharacterStuffingDestuffingLabel  matlab.ui.control.Label
11        OUTPUTLabel        matlab.ui.control.Label
12        OUTPUTEditField    matlab.ui.control.EditField
13        DESTUFFINGButton    matlab.ui.control.Button
14        STUFFINGButton      matlab.ui.control.Button
15        INPUTEditField      matlab.ui.control.EditField
16        INPUTEditFieldLabel matlab.ui.control.Label
17        Image              matlab.ui.control.Image
18        OUTPUTEditFieldLabel matlab.ui.control.Label
19    end
20
21
22    properties (Access = public)
23        globalvariable;
24        d;
25    end
26
27
28    % Callbacks that handle component events
29    methods (Access = private)
30
31        % Value changed function: OUTPUTEditField
32        function OUTPUTEditFieldValueChanged(app, event)
33            value = app.OUTPUTEditField.Value;
34
35        end
36
37        % Value changed function: INPUTEditField
38        function INPUTEditFieldValueChanged(app, event)
39            value = app.INPUTEditField.Value;
40
41        end
42
43        % Button pushed function: STUFFINGButton
44        function STUFFINGButtonPushed(app, event)
45            i = 1;
46            j = 1;
47            app.globalvariable = 0;
48            app.d = blanks(100);
49            l = 'DLEETX';
50            sd = blanks(100);
51            ds = blanks(100);
52
53
54            app.d = app.INPUTEditField.Value;
55
56            sd(1:6) = 'DLESTX';
57            j = 7;
58
59            while i <= length(app.d)
60                if app.d(i) == 'D' && app.d(i + 1) == 'L' && app.d(i + 2) == 'E'
61                    sd(j:j+5) = 'DLEDLE';
62                    j = j + 6;
63                    i = i + 3;
64                else
65                    sd(j) = app.d(i);
66                    j = j + 1;
67                    i = i + 1;
68                end
69            end
70        end
71    end
```

```

71 sd(j) = char(0);
72 ds = sd;
73
74 app.globalvariable = j;
75 ds = strcat(ds, 1);
76
77 app.OUTPUTEditField.Value = ds;
78
79     end
80
81     % Button pushed function: DESTUFFINGButton
82     function DESTUFFINGButtonPushed(app, event)
83 ds = app.INPUTEditField.Value;
84 i = 1;
85 l = 'DLEETX';
86 sd(1:6) = 'DLESTX';
87 j = app.globalvariable;
88 p = blanks(100); %addedlater
89 sd = blanks(100);
90 ds = blanks(100);
91 k = j;
92 while j <= length(ds) && ds(j) ~= 0
93     if ds(j) == 'D' && ds(j + 1) == 'L' && ds(j + 2) == 'E'
94         if ds(j + 3) == 'D' && ds(j + 4) == 'L' && ds(j + 5) == 'E'
95             % Remove "DLE DLE" by shifting remaining characters
96             ds(k:j) = ds(j + 6:end);
97             % Update loop condition and string length
98             j = k;
99             ds = ds(1:k);
100         else
101             % Found single "DLE", copy character (check index bounds)
102             if i <= length(p)
103                 ds(k) = p(i);
104                 i = i + 1;
105                 k = k + 1;
106             end
107         end
108     else
109         % Copy character (check index bounds)
110         if i <= length(p)
111             ds(k) = p(i);
112             i = i + 1;
113             k = k + 1;
114         end
115     end
116     j = j + 1;
117 end
118
119 app.OUTPUTEditField.Value = app.d;
120
121
122     end
123 end
124
125 % Component initialization
126 methods (Access = private)
127
128     % Create UIFigure and components
129     function createComponents(app)
130
131         % Get the file path for locating images
132         pathToMLAPP = fileparts(mfilename('fullpath'));
133
134         % Create UIFigure and hide until all components are created
135         app.UIFigure = uifigure('Visible', 'off');
136         app.UIFigure.Position = [100 100 640 480];
137         app.UIFigure.Name = 'MATLAB App';
138
139         % Create OUTPUTEditFieldLabel
140         app.OUTPUTEditFieldLabel = uilabel(app.UIFigure);
141         app.OUTPUTEditFieldLabel.HorizontalAlignment = 'right';
142         app.OUTPUTEditFieldLabel.FontSize = 24;
143         app.OUTPUTEditFieldLabel.FontWeight = 'bold';

```

```

144 app.OUTPUTEditFieldLabel.Position = [81 317 104 31];
145 app.OUTPUTEditFieldLabel.Text = 'OUTPUT';
146
147 % Create Image
148 app.Image = uiimage(app.UIFigure);
149 app.Image.Position = [-172 -100 985 649];
150 app.Image.ImageSource = fullfile(pathToMLAPP, 'bg.jpg');
151
152 % Create INPUTEditFieldLabel
153 app.INPUTEditFieldLabel = uilabel(app.UIFigure);
154 app.INPUTEditFieldLabel.HorizontalAlignment = 'right';
155 app.INPUTEditFieldLabel.FontSize = 24;
156 app.INPUTEditFieldLabel.FontWeight = 'bold';
157 app.INPUTEditFieldLabel.Position = [108 214 77 31];
158 app.INPUTEditFieldLabel.Text = 'INPUT';
159
160 % Create INPUTEditField
161 app.INPUTEditField = uieditfield(app.UIFigure, 'text');
162 app.INPUTEditField.ValueChangedFcn = createCallbackFcn(app,
    ↳ @INPUTEditFieldValueChanged, true);
163 app.INPUTEditField.FontSize = 24;
164 app.INPUTEditField.Placeholder = 'UNITEDLEDSTATES';
165 app.INPUTEditField.Position = [200 195 361 61];
166
167 % Create STUFFINGButton
168 app.STUFFINGButton = uibutton(app.UIFigure, 'push');
169 app.STUFFINGButton.ButtonPushedFcn = createCallbackFcn(app,
    ↳ @STUFFINGButtonPushed, true);
170 app.STUFFINGButton.BackgroundColor = [0.4784 0.8 0.2902];
171 app.STUFFINGButton.FontSize = 18;
172 app.STUFFINGButton.FontWeight = 'bold';
173 app.STUFFINGButton.Position = [106 72 105 30];
174 app.STUFFINGButton.Text = 'STUFFING';
175
176 % Create DESTUFFINGButton
177 app.DESTUFFINGButton = uibutton(app.UIFigure, 'push');
178 app.DESTUFFINGButton.ButtonPushedFcn = createCallbackFcn(app,
    ↳ @DESTUFFINGButtonPushed, true);
179 app.DESTUFFINGButton.BackgroundColor = [0.9412 0.302 0.302];
180 app.DESTUFFINGButton.FontSize = 18;
181 app.DESTUFFINGButton.FontWeight = 'bold';
182 app.DESTUFFINGButton.Position = [400 72 130 30];
183 app.DESTUFFINGButton.Text = 'DESTUFFING';
184
185 % Create OUTPUTEditField
186 app.OUTPUTEditField = uieditfield(app.UIFigure, 'text');
187 app.OUTPUTEditField.ValueChangedFcn = createCallbackFcn(app,
    ↳ @OUTPUTEditFieldValueChanged, true);
188 app.OUTPUTEditField.Editable = 'off';
189 app.OUTPUTEditField.FontSize = 24;
190 app.OUTPUTEditField.Placeholder = 'Output will show here';
191 app.OUTPUTEditField.Position = [200 298 361 61];
192
193 % Create OUTPUTLabel
194 app.OUTPUTLabel = uilabel(app.UIFigure);
195 app.OUTPUTLabel.FontSize = 24;
196 app.OUTPUTLabel.FontWeight = 'bold';
197 app.OUTPUTLabel.Position = [80 316 104 31];
198 app.OUTPUTLabel.Text = 'OUTPUT';
199
200 % Create CharacterStuffingDestuffingLabel
201 app.CharacterStuffingDestuffingLabel = uilabel(app.UIFigure);
202 app.CharacterStuffingDestuffingLabel.FontSize = 24;
203 app.CharacterStuffingDestuffingLabel.FontWeight = 'bold';
204 app.CharacterStuffingDestuffingLabel.Position = [147 412 370 31];
205 app.CharacterStuffingDestuffingLabel.Text = 'Character Stuffing &
    ↳ Destuffing';
206
207 % Create
208 app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label =
    ↳ uilabel(app.UIFigure);

```



```

209     app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
210         ↳ HorizontalAlignment = 'center';
211     app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
212         ↳ FontWeight = 'bold';
213     app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
214         ↳ Position = [60 3 521 22];
215     app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
216         ↳ Text = 'DEVELOPED BY: MD SAIFUL ISLAM RIMON (213002039) &
217         ↳ NAZMUN NAHAR (191000000)';
218
219     % Create Label
220     app.Label = uilabel(app.UIFigure);
221     app.Label.HorizontalAlignment = 'center';
222     app.Label.FontWeight = 'bold';
223     app.Label.Position = [68 28 499 22];
224     app.Label.Text = 'PROJECT SUPERVISOR: RUSMITA HALIM CHAITY, LECTURER
225         ↳ , DEPT. OF CSE, GUB';
226
227     % Create Image2
228     app.Image2 = uiimage(app.UIFigure);
229     app.Image2.Position = [43 381 71 80];
230     app.Image2.ImageSource = 'images__1_-removebg-preview.png';
231
232     % Create Image3
233     app.Image3 = uiimage(app.UIFigure);
234     app.Image3.Position = [537 381 93 80];
235     app.Image3.ImageSource = 'Logo-PNG-1.png';
236
237     % Show the figure after all components are created
238     app.UIFigure.Visible = 'on';
239 end
240
241 % App creation and deletion
242 methods (Access = public)
243
244     % Construct app
245     function app = CharacterStuffingDestuffing
246
247         % Create UIFigure and components
248         createComponents(app)
249
250         % Register the app with App Designer
251         registerApp(app, app.UIFigure)
252
253         if nargin == 0
254             clear app
255         end
256     end
257
258     % Code that executes before app deletion
259     function delete(app)
260
261         % Delete UIFigure when app is deleted
262         delete(app.UIFigure)
263     end
264 end
265 end

```

## CODE [IPv4 Conversion Page]:

```
1 classdef ipv4 < matlab.apps.AppBase
2
3     % Properties that correspond to app components
4     properties (Access = public)
5         UIFigure                matlab.ui.Figure
6         Label                    matlab.ui.control.Label
7         DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label    matlab.
            ↳ ui.control.Label
8         Image3                    matlab.ui.control.Image
9         Image2                    matlab.ui.control.Image
10        IPv4CONVERSIONLabel        matlab.ui.control.Label
11        DecimalIPEditField          matlab.ui.control.EditField
12        DecimalIPEditFieldLabel    matlab.ui.control.Label
13        BinaryIPToDecimalButton     matlab.ui.control.Button
14        BinaryIPEditField           matlab.ui.control.EditField
15        BinaryIPEditFieldLabel      matlab.ui.control.Label
16        InputEditField              matlab.ui.control.EditField
17        InputEditFieldLabel         matlab.ui.control.Label
18        DecimalIPToBinaryButton     matlab.ui.control.Button
19        Image                        matlab.ui.control.Image
20    end
21
22    % Callbacks that handle component events
23    methods (Access = private)
24
25        % Value changed function: InputEditField
26        function InputEditFieldValueChanged(app, event)
27            value = app.InputEditField.Value;
28
29        end
30
31        % Button pushed function: DecimalIPToBinaryButton
32        function DecimalIPToBinaryButtonPushed(app, event)
33            dec = app.InputEditField.Value;
34            octets = str2double(strsplit(dec, ','));
35            binaryIP = dec2bin(octets, 8);
36            binaryIP = reshape(binaryIP', 1, []);
37            binaryIP = regexprep(binaryIP, '(\d{8})', '$1 ');
38            app.BinaryIPEditField.Value = string(binaryIP);
39
40        end
41
42        % Value changed function: BinaryIPEditField
43        function BinaryIPEditFieldValueChanged(app, event)
44            value = app.BinaryIPEditField.Value;
45
46        end
47
48        % Button pushed function: BinaryIPToDecimalButton
49        function BinaryIPToDecimalButtonPushed(app, event)
50            binaryIP = app.InputEditField.Value;
51
52            % Remove whitespaces and split the binary string into octets
53            binaryOctets = strsplit(strtrim(binaryIP), ' ');
54
55            % Convert each binary octet to decimal
56            decimalOctets = zeros(1, numel(binaryOctets));
57            for i = 1:numel(binaryOctets)
58                decimalOctets(i) = bin2dec(binaryOctets{i});
59            end
60
61            % Join decimal octets into an IPv4 address
62            ipv4Address = join(string(decimalOctets), '.');
63            app.DecimalIPEditField.Value = string(ipv4Address);
64
65
66
67        end
68
69        % Value changed function: DecimalIPEditField
70        function DecimalIPEditFieldValueChanged(app, event)
```

```

71         value = app.DecimalIPEditField.Value;
72
73     end
74 end
75
76 % Component initialization
77 methods (Access = private)
78
79     % Create UIFigure and components
80     function createComponents(app)
81
82         % Create UIFigure and hide until all components are created
83         app UIFigure = uifigure('Visible', 'off');
84         app UIFigure.Position = [100 100 640 480];
85         app UIFigure.Name = 'MATLAB App';
86
87         % Create Image
88         app.Image = uimage(app UIFigure);
89         app.Image.Position = [-211 -93 1059 654];
90         app.Image.ImageSource = 'bg.jpg';
91
92         % Create DecimalIPToBinaryButton
93         app.DecimalIPToBinaryButton = uibutton(app UIFigure, 'push');
94         app.DecimalIPToBinaryButton.ButtonPushedFcn = createCallbackFcn(app,
95             ↪ @DecimalIPToBinaryButtonPushed, true);
96         app.DecimalIPToBinaryButton.BackgroundColor = [1 0.4118 0.1608];
97         app.DecimalIPToBinaryButton.FontWeight = 'bold';
98         app.DecimalIPToBinaryButton.Position = [84 117 473 38];
99         app.DecimalIPToBinaryButton.Text = {'Decimal IP '; ' To Binary'};
100
101         % Create InputEditFieldLabel
102         app.InputEditFieldLabel = uilabel(app UIFigure);
103         app.InputEditFieldLabel.HorizontalAlignment = 'right';
104         app.InputEditFieldLabel.FontSize = 18;
105         app.InputEditFieldLabel.Position = [92 173 45 23];
106         app.InputEditFieldLabel.Text = 'Input';
107
108         % Create InputEditField
109         app.InputEditField = uieditfield(app UIFigure, 'text');
110         app.InputEditField.ValueChangedFcn = createCallbackFcn(app,
111             ↪ @InputEditFieldValueChanged, true);
112         app.InputEditField.FontSize = 18;
113         app.InputEditField.Placeholder = '192.168.10.10';
114         app.InputEditField.Position = [152 170 397 30];
115
116         % Create BinaryIPEditFieldLabel
117         app.BinaryIPEditFieldLabel = uilabel(app UIFigure);
118         app.BinaryIPEditFieldLabel.HorizontalAlignment = 'right';
119         app.BinaryIPEditFieldLabel.FontSize = 24;
120         app.BinaryIPEditFieldLabel.FontWeight = 'bold';
121         app.BinaryIPEditFieldLabel.Position = [22 327 109 31];
122         app.BinaryIPEditFieldLabel.Text = 'Binary IP';
123
124         % Create BinaryIPEditField
125         app.BinaryIPEditField = uieditfield(app UIFigure, 'text');
126         app.BinaryIPEditField.ValueChangedFcn = createCallbackFcn(app,
127             ↪ @BinaryIPEditFieldValueChanged, true);
128         app.BinaryIPEditField.Editable = 'off';
129         app.BinaryIPEditField.FontSize = 18;
130         app.BinaryIPEditField.FontWeight = 'bold';
131         app.BinaryIPEditField.Position = [146 311 403 71];
132
133         % Create BinaryIPToDecimalButton
134         app.BinaryIPToDecimalButton = uibutton(app UIFigure, 'push');
135         app.BinaryIPToDecimalButton.ButtonPushedFcn = createCallbackFcn(app,
136             ↪ @BinaryIPToDecimalButtonPushed, true);
137         app.BinaryIPToDecimalButton.BackgroundColor = [0.0941 0.6588 0.0941];
138         app.BinaryIPToDecimalButton.FontWeight = 'bold';
139         app.BinaryIPToDecimalButton.Position = [84 71 473 38];
140         app.BinaryIPToDecimalButton.Text = {'Binary IP '; ' To Decimal'};
141
142         % Create DecimalIPEditFieldLabel
143         app.DecimalIPEditFieldLabel = uilabel(app UIFigure);

```

```

140     app.DecimalIPEditFieldLabel.HorizontalAlignment = 'right';
141     app.DecimalIPEditFieldLabel.FontSize = 24;
142     app.DecimalIPEditFieldLabel.FontWeight = 'bold';
143     app.DecimalIPEditFieldLabel.Position = [5 241 126 31];
144     app.DecimalIPEditFieldLabel.Text = 'Decimal IP';
145
146     % Create DecimalIPEditField
147     app.DecimalIPEditField = uieditfield(app.UIFigure, 'text');
148     app.DecimalIPEditField.ValueChangedFcn = createCallbackFcn(app,
149         ↳ @DecimalIPEditFieldValueChanged, true);
150     app.DecimalIPEditField.Editable = 'off';
151     app.DecimalIPEditField.FontSize = 24;
152     app.DecimalIPEditField.FontWeight = 'bold';
153     app.DecimalIPEditField.Position = [146 225 403 71];
154
155     % Create IPv4CONVERSIONLabel
156     app.IPv4CONVERSIONLabel = uilabel(app.UIFigure);
157     app.IPv4CONVERSIONLabel.FontSize = 24;
158     app.IPv4CONVERSIONLabel.FontWeight = 'bold';
159     app.IPv4CONVERSIONLabel.Position = [209 431 224 31];
160     app.IPv4CONVERSIONLabel.Text = 'IPv4 CONVERSION';
161
162     % Create Image2
163     app.Image2 = uiimage(app.UIFigure);
164     app.Image2.Position = [22 380 71 84];
165     app.Image2.ImageSource = 'images__1_-removebg-preview.png';
166
167     % Create Image3
168     app.Image3 = uiimage(app.UIFigure);
169     app.Image3.Position = [522 371 100 100];
170     app.Image3.ImageSource = 'Logo-PNG-1.png';
171
172     % Create
173     ↳ DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label
174     app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label =
175     ↳ uilabel(app.UIFigure);
176     app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
177     ↳ HorizontalAlignment = 'center';
178     app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
179     ↳ FontWeight = 'bold';
180     app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
181     ↳ Position = [60 0 521 22];
182     app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
183     ↳ Text = 'DEVELOPED BY: MD SAIFUL ISLAM RIMON (213002039) &
184     ↳ NAZMUN NAHAR (191000000)';
185
186     % Create Label
187     app.Label = uilabel(app.UIFigure);
188     app.Label.HorizontalAlignment = 'center';
189     app.Label.FontWeight = 'bold';
190     app.Label.Position = [68 25 499 22];
191     app.Label.Text = 'PROJECT SUPERVISOR: RUSMITA HALIM CHAITY, LECTURER
192     ↳ , DEPT. OF CSE, GUB';
193
194     % Show the figure after all components are created
195     app.UIFigure.Visible = 'on';
196
197 end
198
199 % App creation and deletion
200 methods (Access = public)
201
202     % Construct app
203     function app = ipv4
204
205         % Create UIFigure and components
206         createComponents(app)
207
208         % Register the app with App Designer
209         registerApp(app, app.UIFigure)
210
211         if nargin == 0
212             clear app
213

```

```
204         end
205     end
206
207     % Code that executes before app deletion
208     function delete(app)
209
210         % Delete UIFigure when app is deleted
211         delete(app.UIFigure)
212     end
213 end
214 end
```

## CODE [Cyclic Redundancy Check (CRC) Page]:

```

1  classdef crcapp < matlab.apps.AppBase
2
3      % Properties that correspond to app components
4      properties (Access = public)
5          UIFigure                matlab.ui.Figure
6          InputBitStringLabel      matlab.ui.control.Label
7          Label                    matlab.ui.control.Label
8          DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label  matlab.
           ↳ ui.control.Label
9
10         Image3                   matlab.ui.control.Image
11         Image2                   matlab.ui.control.Image
12         CyclicRedundancyCheckCRCLabel  matlab.ui.control.Label
13         Lamp                     matlab.ui.control.Lamp
14         ErrorStatusEditField      matlab.ui.control.EditField
15         ErrorStatusEditFieldLabel  matlab.ui.control.Label
16         CheckButton              matlab.ui.control.Button
17         ReceivedCodewordEditField  matlab.ui.control.EditField
18         ReceivedCodewordEditFieldLabel  matlab.ui.control.Label
19         GenerateTransmittingCodewordButton  matlab.ui.control.Button
20         TransmittingCodewordEditField  matlab.ui.control.EditField
21         TransmittingCodewordLabel  matlab.ui.control.Label
22         InputPolynomialinBinaryEditField  matlab.ui.control.EditField
23         InputPolynomialinBinaryEditFieldLabel  matlab.ui.control.Label
24         InputBitStingEditField      matlab.ui.control.EditField
25         Image                     matlab.ui.control.Image
26         InputBitStingEditFieldLabel  matlab.ui.control.Label
27     end
28
29     properties (Access = public)
30         gen_poly
31     end
32
33     methods (Access = public)
34
35         function crc_matlab_sender(app)
36             % Define the data and polynomial
37             data = app.InputBitStingEditField.Value;
38             app.gen_poly = app.InputPolynomialinBinaryEditField.Value;
39
40             data_length = length(data);
41             N = length(app.gen_poly);
42
43             % Append zeros to the data
44             appended_data = [data, repmat('0', 1, N-1)];
45
46             % Perform CRC
47             remainder = crc_division(app, appended_data, app.gen_poly);
48
49             % Generate codeword
50             codeword = [data, remainder];
51
52             app.TransmittingCodewordEditField.Value = codeword;
53
54             end
55
56         function crc_matlab_receiver(app)
57             % Define the data and polynomial
58             received_data = app.ReceivedCodewordEditField.Value;
59             received_remainder = crc_division(app, received_data, app.gen_poly);
60
61             if any(received_remainder ~= '0')
62                 app.ErrorStatusEditField.Value = 'Error Detected';
63                 app.Lamp.Color = [1.00,0.00,0.00];
64
65             else
66                 app.ErrorStatusEditField.Value = 'No Error Detected';
67                 app.Lamp.Color = [0.00,1.00,0.00];
68             end
69
70             end

```

```

71
72
73 function remainder = crc_division(app, input_data, gen_poly)
74     N = length(gen_poly);
75     temp_data = input_data(1:N);
76
77     for i = 1:length(input_data) - N + 1
78         if temp_data(1) == '1'
79             temp_data = xor_strings(app, temp_data, gen_poly);
80         end
81
82         if i+N <= length(input_data)
83             temp_data = [temp_data(2:end), input_data(i+N)];
84         else
85             temp_data = temp_data(2:end);
86         end
87     end
88
89     remainder = temp_data;
90 end
91
92 function result = xor_strings(app, a, b)
93     result = char(xor(a-'0', b-'0') + '0');
94 end
95
96     end
97
98
99 % Callbacks that handle component events
100 methods (Access = private)
101
102     % Value changed function: InputBitStingEditField
103     function InputBitStingEditFieldValueChanged(app, event)
104         value = app.InputBitStingEditField.Value;
105
106     end
107
108     % Value changed function: InputPolynomialinBinaryEditField
109     function InputPolynomialinBinaryEditFieldValueChanged(app, event)
110         value = app.InputPolynomialinBinaryEditField.Value;
111
112     end
113
114     % Button pushed function: GenerateTransmittingCodewordButton
115     function GenerateTransmittingCodewordButtonPushed(app, event)
116         crc_matlab_sender(app);
117
118
119     end
120
121     % Value changed function: TransmittingCodewordEditField
122     function TransmittingCodewordEditFieldValueChanged(app, event)
123         value = app.TransmittingCodewordEditField.Value;
124
125     end
126
127     % Value changed function: ReceivedCodewordEditField
128     function ReceivedCodewordEditFieldValueChanged(app, event)
129         value = app.ReceivedCodewordEditField.Value;
130
131     end
132
133     % Button pushed function: CheckButton
134     function CheckButtonPushed(app, event)
135         crc_matlab_receiver(app)
136     end
137
138     % Value changed function: ErrorStatusEditField
139     function ErrorStatusEditFieldValueChanged(app, event)
140         value = app.ErrorStatusEditField.Value;
141
142     end
143 end

```

```

144
145 % Component initialization
146 methods (Access = private)
147
148 % Create UIFigure and components
149 function createComponents(app)
150
151 % Create UIFigure and hide until all components are created
152 app.UIFigure = uifigure('Visible', 'off');
153 app.UIFigure.Position = [100 100 640 480];
154 app.UIFigure.Name = 'MATLAB App';
155
156 % Create InputBitStingEditFieldLabel
157 app.InputBitStingEditFieldLabel = uilabel(app.UIFigure);
158 app.InputBitStingEditFieldLabel.HorizontalAlignment = 'right';
159 app.InputBitStingEditFieldLabel.FontSize = 14;
160 app.InputBitStingEditFieldLabel.FontWeight = 'bold';
161 app.InputBitStingEditFieldLabel.Position = [29 366 101 22];
162 app.InputBitStingEditFieldLabel.Text = 'Input Bit Sting';
163
164 % Create Image
165 app.Image = uimage(app.UIFigure);
166 app.Image.Position = [-166 -76 979 647];
167 app.Image.ImageSource = 'bg.jpg';
168
169 % Create InputBitStingEditField
170 app.InputBitStingEditField = uieditfield(app.UIFigure, 'text');
171 app.InputBitStingEditField.ValueChangedFcn = createCallbackFcn(app,
    ↳ @InputBitStingEditFieldValueChanged, true);
172 app.InputBitStingEditField.FontSize = 14;
173 app.InputBitStingEditField.FontWeight = 'bold';
174 app.InputBitStingEditField.Placeholder = '1001010';
175 app.InputBitStingEditField.Position = [145 366 127 22];
176
177 % Create InputPolynomialinBinaryEditFieldLabel
178 app.InputPolynomialinBinaryEditFieldLabel = uilabel(app.UIFigure);
179 app.InputPolynomialinBinaryEditFieldLabel.HorizontalAlignment = '
    ↳ right';
180 app.InputPolynomialinBinaryEditFieldLabel.FontSize = 14;
181 app.InputPolynomialinBinaryEditFieldLabel.FontWeight = 'bold';
182 app.InputPolynomialinBinaryEditFieldLabel.Position = [289 366 182
    ↳ 22];
183 app.InputPolynomialinBinaryEditFieldLabel.Text = 'Input Polynomial in
    ↳ Binary';
184
185 % Create InputPolynomialinBinaryEditField
186 app.InputPolynomialinBinaryEditField = uieditfield(app.UIFigure, '
    ↳ text');
187 app.InputPolynomialinBinaryEditField.ValueChangedFcn =
    ↳ createCallbackFcn(app,
    ↳ @InputPolynomialinBinaryEditFieldValueChanged, true);
188 app.InputPolynomialinBinaryEditField.FontSize = 14;
189 app.InputPolynomialinBinaryEditField.FontWeight = 'bold';
190 app.InputPolynomialinBinaryEditField.Placeholder = '1011';
191 app.InputPolynomialinBinaryEditField.Position = [486 366 127 22];
192
193 % Create TransmittingCodewordLabel
194 app.TransmittingCodewordLabel = uilabel(app.UIFigure);
195 app.TransmittingCodewordLabel.FontSize = 18;
196 app.TransmittingCodewordLabel.FontWeight = 'bold';
197 app.TransmittingCodewordLabel.Position = [38 227 113 55];
198 app.TransmittingCodewordLabel.Text = {'Transmitting'; 'Codeword'};
199
200 % Create TransmittingCodewordEditField
201 app.TransmittingCodewordEditField = uieditfield(app.UIFigure, 'text')
    ↳ ;
202 app.TransmittingCodewordEditField.ValueChangedFcn = createCallbackFcn
    ↳ (app, @TransmittingCodewordEditFieldValueChanged, true);
203 app.TransmittingCodewordEditField.Eitable = 'off';
204 app.TransmittingCodewordEditField.FontSize = 18;
205 app.TransmittingCodewordEditField.FontWeight = 'bold';
206 app.TransmittingCodewordEditField.Placeholder = 'Shows Transmitting
    ↳ Codeword';

```



```

207     app.TransmittingCodewordEditField.Position = [166 227 321 55];
208
209     % Create GenerateTransmittingCodewordButton
210     app.GenerateTransmittingCodewordButton = uibutton(app.UIFigure, 'push
    ↳ ');
211     app.GenerateTransmittingCodewordButton.ButtonPushedFcn =
    ↳ createCallbackFcn(app,
    ↳ @GenerateTransmittingCodewordButtonPushed, true);
212     app.GenerateTransmittingCodewordButton.BackgroundColor = [0.4784 0.8
    ↳ 0.2902];
213     app.GenerateTransmittingCodewordButton.FontSize = 14;
214     app.GenerateTransmittingCodewordButton.FontWeight = 'bold';
215     app.GenerateTransmittingCodewordButton.Position = [180 314 238 25];
216     app.GenerateTransmittingCodewordButton.Text = 'Generate Transmitting
    ↳ Codeword';
217
218     % Create ReceivedCodewordEditFieldLabel
219     app.ReceivedCodewordEditFieldLabel = uilabel(app.UIFigure);
220     app.ReceivedCodewordEditFieldLabel.HorizontalAlignment = 'right';
221     app.ReceivedCodewordEditFieldLabel.FontSize = 14;
222     app.ReceivedCodewordEditFieldLabel.FontWeight = 'bold';
223     app.ReceivedCodewordEditFieldLabel.Position = [37 153 139 22];
224     app.ReceivedCodewordEditFieldLabel.Text = 'Received Codeword';
225
226     % Create ReceivedCodewordEditField
227     app.ReceivedCodewordEditField = uieditfield(app.UIFigure, 'text');
228     app.ReceivedCodewordEditField.ValueChangedFcn = createCallbackFcn(app
    ↳ , @ReceivedCodewordEditFieldValueChanged, true);
229     app.ReceivedCodewordEditField.FontSize = 14;
230     app.ReceivedCodewordEditField.FontWeight = 'bold';
231     app.ReceivedCodewordEditField.Placeholder = '1001010111';
232     app.ReceivedCodewordEditField.Position = [191 153 214 22];
233
234     % Create CheckButton
235     app.CheckButton = uibutton(app.UIFigure, 'push');
236     app.CheckButton.ButtonPushedFcn = createCallbackFcn(app,
    ↳ @CheckButtonPushed, true);
237     app.CheckButton.BackgroundColor = [0.4784 0.8 0.2902];
238     app.CheckButton.FontSize = 14;
239     app.CheckButton.FontWeight = 'bold';
240     app.CheckButton.Position = [444 152 100 25];
241     app.CheckButton.Text = 'Check';
242
243     % Create ErrorStatusEditFieldLabel
244     app.ErrorStatusEditFieldLabel = uilabel(app.UIFigure);
245     app.ErrorStatusEditFieldLabel.HorizontalAlignment = 'center';
246     app.ErrorStatusEditFieldLabel.FontSize = 18;
247     app.ErrorStatusEditFieldLabel.FontWeight = 'bold';
248     app.ErrorStatusEditFieldLabel.Position = [38 70 113 55];
249     app.ErrorStatusEditFieldLabel.Text = 'Error Status';
250
251     % Create ErrorStatusEditField
252     app.ErrorStatusEditField = uieditfield(app.UIFigure, 'text');
253     app.ErrorStatusEditField.ValueChangedFcn = createCallbackFcn(app,
    ↳ @ErrorStatusEditFieldValueChanged, true);
254     app.ErrorStatusEditField.Editable = 'off';
255     app.ErrorStatusEditField.HorizontalAlignment = 'center';
256     app.ErrorStatusEditField.FontSize = 18;
257     app.ErrorStatusEditField.FontWeight = 'bold';
258     app.ErrorStatusEditField.Placeholder = 'Shows Error Status';
259     app.ErrorStatusEditField.Position = [166 70 321 55];
260
261     % Create Lamp
262     app.Lamp = uilamp(app.UIFigure);
263     app.Lamp.Position = [505 78 42 42];
264     app.Lamp.Color = [1 0 0];
265
266     % Create CyclicRedundancyCheckCRCLabel
267     app.CyclicRedundancyCheckCRCLabel = uilabel(app.UIFigure);
268     app.CyclicRedundancyCheckCRCLabel.HorizontalAlignment = 'center';
269     app.CyclicRedundancyCheckCRCLabel.WordWrap = 'on';
270     app.CyclicRedundancyCheckCRCLabel.FontSize = 14;
271     app.CyclicRedundancyCheckCRCLabel.FontWeight = 'bold';

```

```

272     app.CyclicRedundancyCheckCRCLabel.Position = [140 433 362 22];
273     app.CyclicRedundancyCheckCRCLabel.Text = 'Cyclic Redundancy Check (
        ↳ CRC)';

274
275     % Create Image2
276     app.Image2 = uiimage(app.UIFigure);
277     app.Image2.Position = [33 394 71 80];
278     app.Image2.ImageSource = 'images__1_-removebg-preview.png';
279
280     % Create Image3
281     app.Image3 = uiimage(app.UIFigure);
282     app.Image3.Position = [538 394 93 80];
283     app.Image3.ImageSource = 'Logo-PNG-1.png';
284
285     % Create
        ↳ DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label
286     app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label =
        ↳ uilabel(app.UIFigure);
287     app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
        ↳ HorizontalAlignment = 'center';
288     app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
        ↳ FontWeight = 'bold';
289     app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
        ↳ Position = [60 3 521 22];
290     app.DEVELOPEDBYMDSAIFULISLAMRIMON213002039NAZMUNNAHAR191000000Label.
        ↳ Text = 'DEVELOPED BY: MD SAIFUL ISLAM RIMON (213002039) &
        ↳ NAZMUN NAHAR (191000000)';

291
292     % Create Label
293     app.Label = uilabel(app.UIFigure);
294     app.Label.HorizontalAlignment = 'center';
295     app.Label.FontWeight = 'bold';
296     app.Label.Position = [68 28 499 22];
297     app.Label.Text = 'PROJECT SUPERVISOR: RUSMITA HALIM CHAITY, LECTURER
        ↳ , DEPT. OF CSE, GUB';

298
299     % Create InputBitStringLabel
300     app.InputBitStringLabel = uilabel(app.UIFigure);
301     app.InputBitStringLabel.FontSize = 14;
302     app.InputBitStringLabel.FontWeight = 'bold';
303     app.InputBitStringLabel.Position = [37 366 106 22];
304     app.InputBitStringLabel.Text = 'Input Bit String';
305
306     % Show the figure after all components are created
307     app.UIFigure.Visible = 'on';
308
309     end
310
311     % App creation and deletion
312     methods (Access = public)
313
314         % Construct app
315         function app = crcapp
316
317             % Create UIFigure and components
318             createComponents(app)
319
320             % Register the app with App Designer
321             registerApp(app, app.UIFigure)
322
323             if nargin == 0
324                 clear app
325             end
326         end
327
328         % Code that executes before app deletion
329         function delete(app)
330
331             % Delete UIFigure when app is deleted
332             delete(app.UIFigure)
333         end
334     end
335 end

```

## 2.4 Algorithms

---

**Algorithm 1: DATA TRANSMISSION SIMULATOR (MATLAB APP)**

---

Start the MATLAB environment and initialize the main application window.

Load any necessary resources or libraries.

**Create Home Page Interface:**

Create a menu-driven interface named "Home Page."

Add buttons for each feature:

- Line Coding

- Hamming Encoding & Decoding

- Analog to Digital Signal Conversion

- Character Stuffing & Destuffing

- IPv4 Conversion

- Bit Stuffing & Destuffing

- CRC Error Detection

**Feature Selection Handling:**

Implement a callback function for each menu item.

Each callback invokes the corresponding feature module.

**Implement Feature Modules: for *each feature module* do**

- Define functionality for each module.

Create functions for the specific operations.

**User Input and Output:**

Design input boxes for user data entry.

Design output boxes for displaying results.

Implement a graphical demonstration system for visualizing data.

**Execution of Operations: for *each operation* do**

- Retrieve data from input boxes.

Process data using the respective algorithm.

Display results in output boxes or graphically.

**Validation and Error Handling:**

Include input validation for data integrity.

Implement error handling and user feedback mechanisms.

**Cleanup and Close Application:**

Allow users to exit to Home Page or close the application.

Release any allocated resources.

---

# Chapter 3

## Performance Evaluation

### 3.1 Simulation Environment/ Simulation Procedure

1. **Platform:** The code is written in Matlab language, suggesting it is intended to run on an Matlab App Designer. The environment should include all Matlab Packages and App Designer for running the compiled code.
2. **Code Compilation:** Use Matlab App Designer to compile all parts of codes and files first. Otherwise, the program will generate an error like: a file missing.
3. **Execution:** After running once all the files, Now run the homepage only and click every functionality without clicking on Run for every file. Just enjoy!

## 3.2 Results Analysis/Testing

### 3.2.1 Home Page

After running the project, at first, we will have graphical interface like this. Here we have to click which transmission technique we are going to use and it will open a new window automatically.

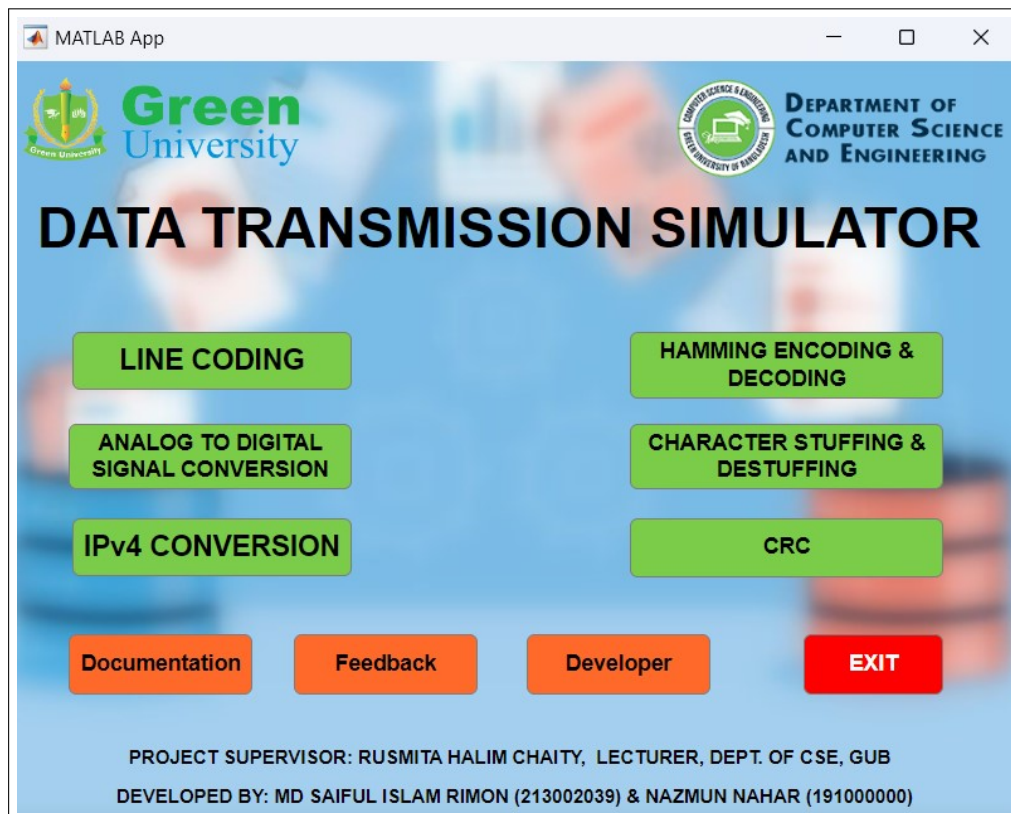


Figure 3.1: Graphical User Interface: Home Page

### 3.2.2 Line Coding

After clicking on Line Coding button on the home page, we will see a graphical user interface like the picture below. Here, we can input the bit-stream and can try different line coding techniques just by clicking on the buttons for Line Coding techniques.

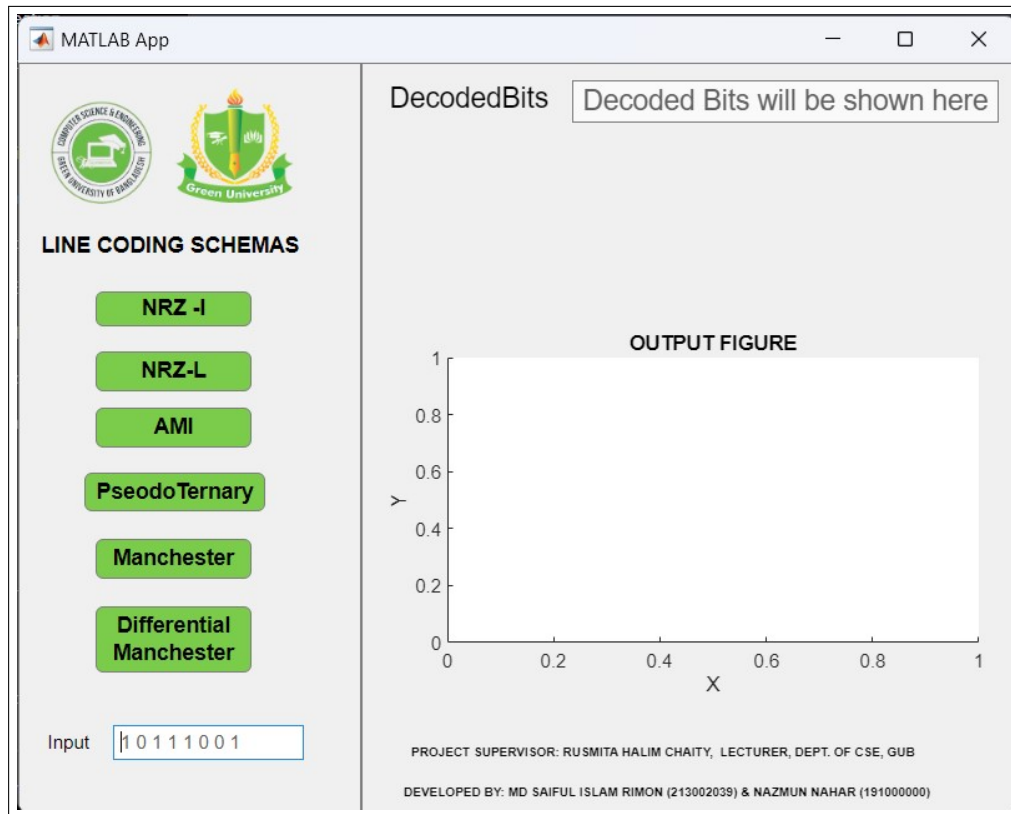


Figure 3.2: Graphical User Interface: Line Coding

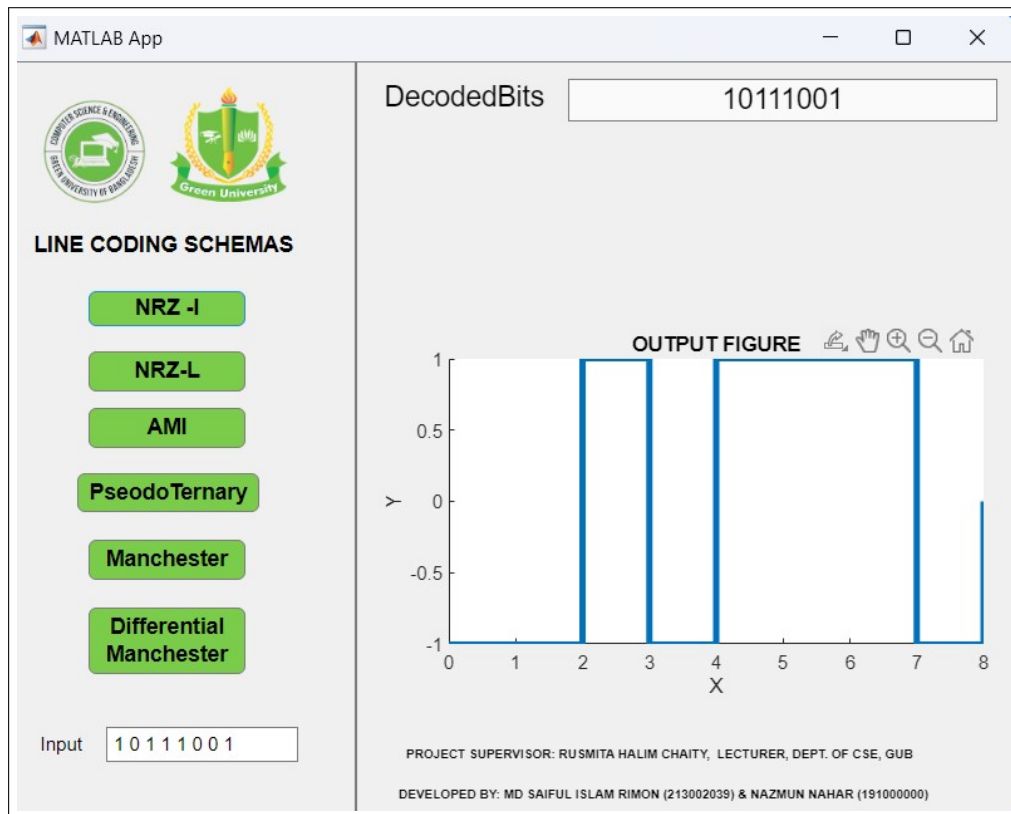


Figure 3.3: Graphical User Interface: Line Coding: NRZ - I

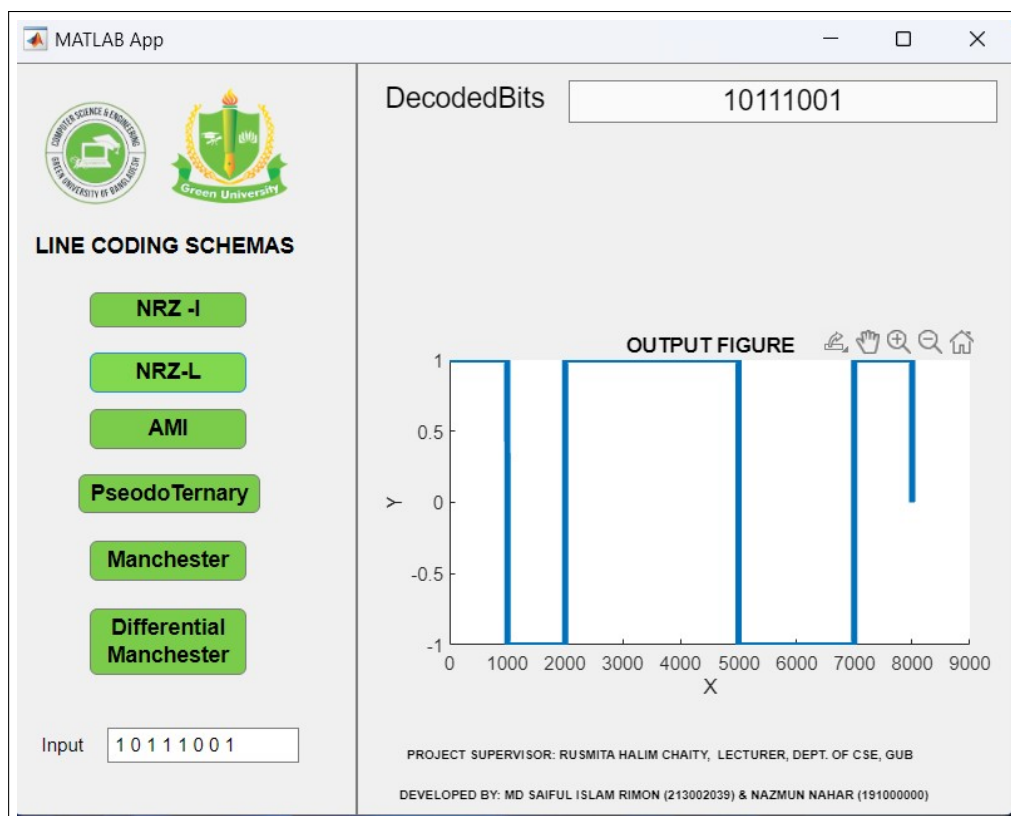


Figure 3.4: Graphical User Interface: Line Coding: NRZ - L

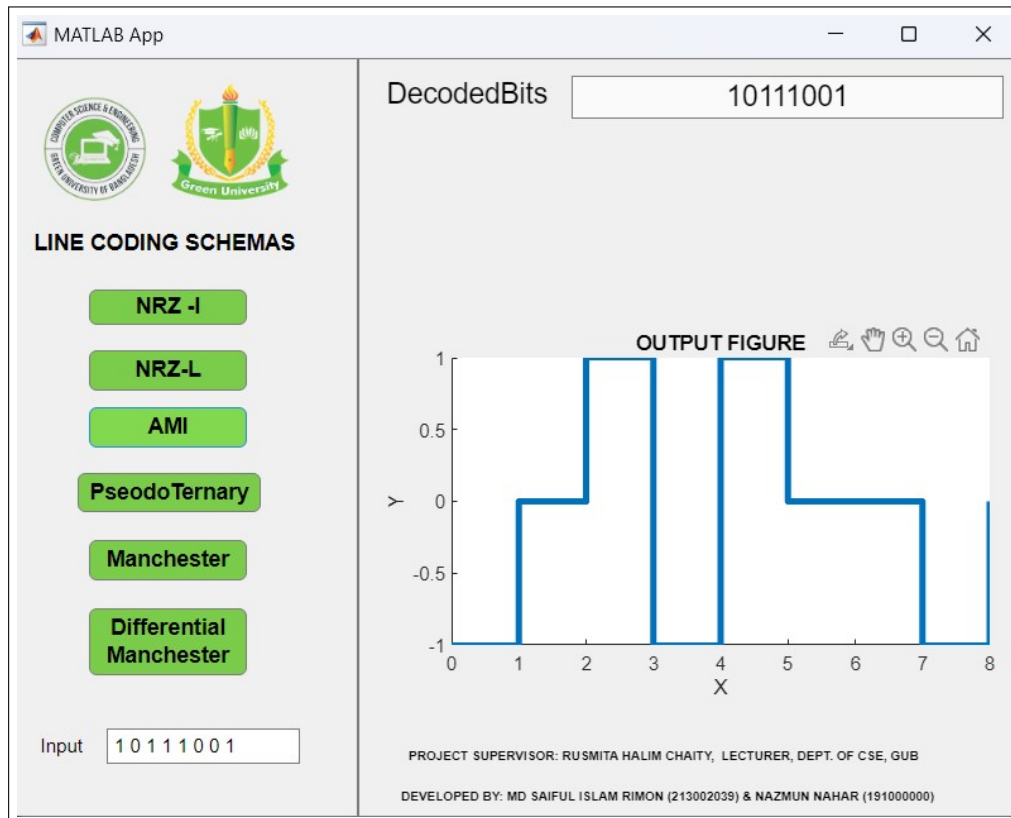


Figure 3.5: Graphical User Interface: Line Coding: AMI

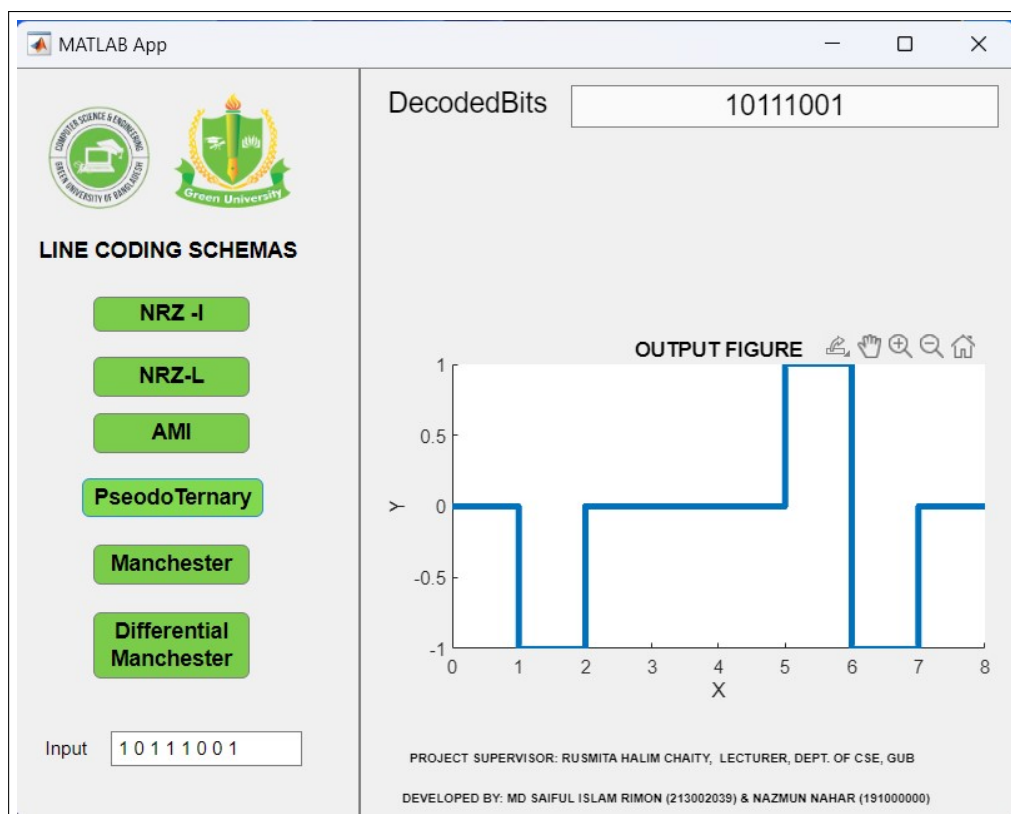


Figure 3.6: Graphical User Interface: Line Coding: PSEUDO TERNARY



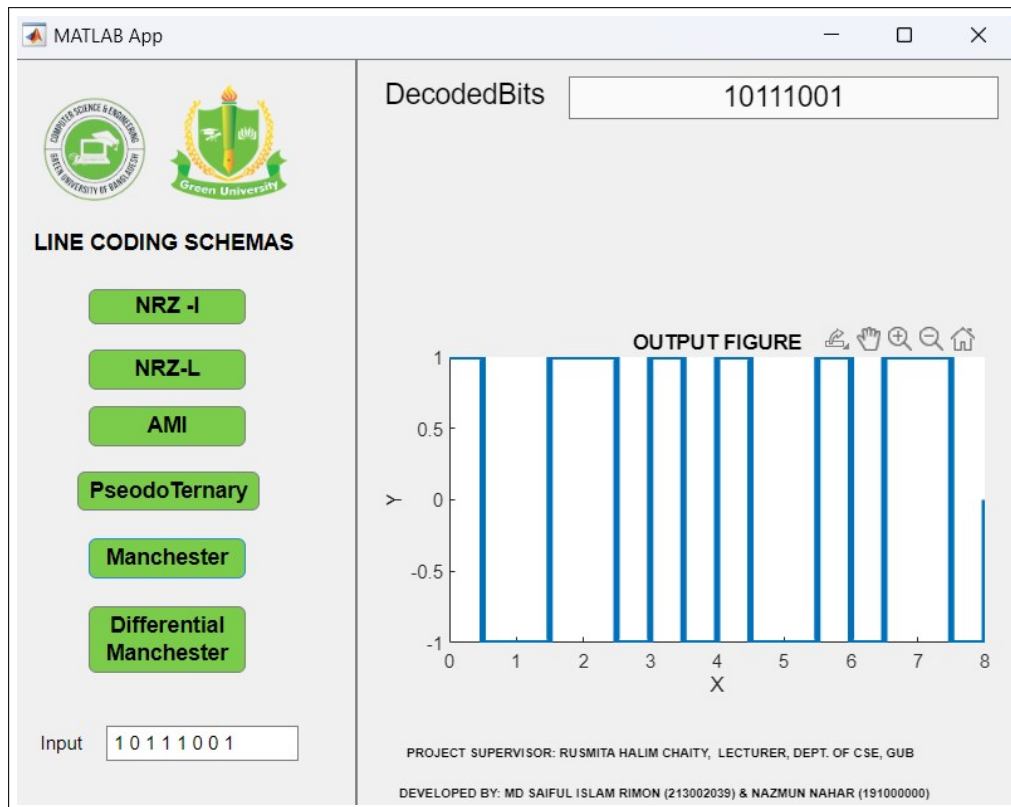


Figure 3.7: Graphical User Interface: Line Coding: MANCHESTER

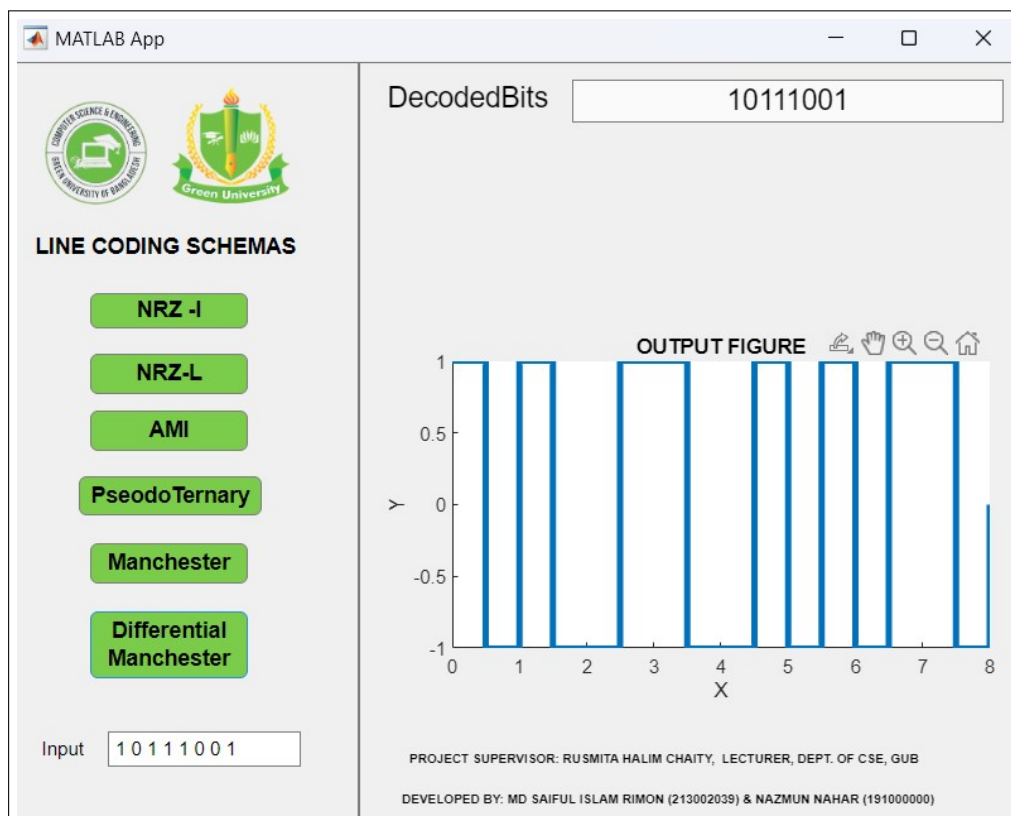


Figure 3.8: Graphical User Interface: Line Coding: DIFFERENTIAL MANCHESTER

### 3.2.3 Hamming Encoding & Decoding

After clicking on Hamming Encoding & Decoding button on the home page, we will see a graphical user interface like the picture below. Here, we can input the bit-stream and can see encoding and its parity bits. Then, if the receiver's bit-stream is different, then it will show the position of error and correct the bit-stream. It is the, single bit error detection and correction technique.

The image shows a MATLAB App window titled "MATLAB App" with a standard window control bar. The main interface has a blue background and contains the following elements:

- Logos:** On the top left is the logo of "COMPUTER SCIENCE & ENGINEERING, GREEN UNIVERSITY OF BANGLADESH". On the top right is the logo of "Green University".
- Title:** "4 BIT HAMMING CODE PARITY BITS AND ERROR DETECTION" is centered at the top.
- Encoding Section:**
  - A text input field contains "1 0 1 0".
  - A green button labeled "GENERATE SENDING BIT STRING" is to the right.
  - Below this, three small input fields for parity bits are labeled P1 (value 1), P2 (value 0), and P4 (value 1).
  - To the right, two more input fields are labeled "NUMBER OF PARITY BITS" (value 3) and "LENGTH OF HAMMING CODE" (value 7).
  - A text input field labeled "SENDED BIT STRING" contains "1 0 1 1 0 1 0".
- Error Detection Section:**
  - A green button labeled "ERROR DETECTION AND CORRECTION" is positioned above the received bit string input.
  - A text input field labeled "RECEIVED BIT STRING" contains "1 0 1 1 0 1 0".
  - Below this, three small input fields for parity bits are labeled P1 (value 1), P2 (value 1), and P4 (value 0).
  - To the right, two more input fields are labeled "POSITION OF ERROR" (value 3) and "CORRECTED BIT STRING" (value "1 0 1 1 0 1 0").
- Footer:** At the bottom, it says "PROJECT SUPERVISOR: RUSMITA HALIM CHAITY, LECTURER, DEPT. OF CSE, GUB" and "DEVELOPED BY: MD SAIFUL ISLAM RIMON (213002039) & NAZMUN NAHAR (191000000)".

Figure 3.9: Graphical User Interface: Hamming Encoding & Decoding Page

### 3.2.4 Analog to Digital Signal Conversion

After clicking on the Analog to Digital Signal Conversion button on the home page, we will see a graphical user interface like the picture below. Here, we can input the frequency and the number of bits and it will show both the Analog signal and the converted digital signal.

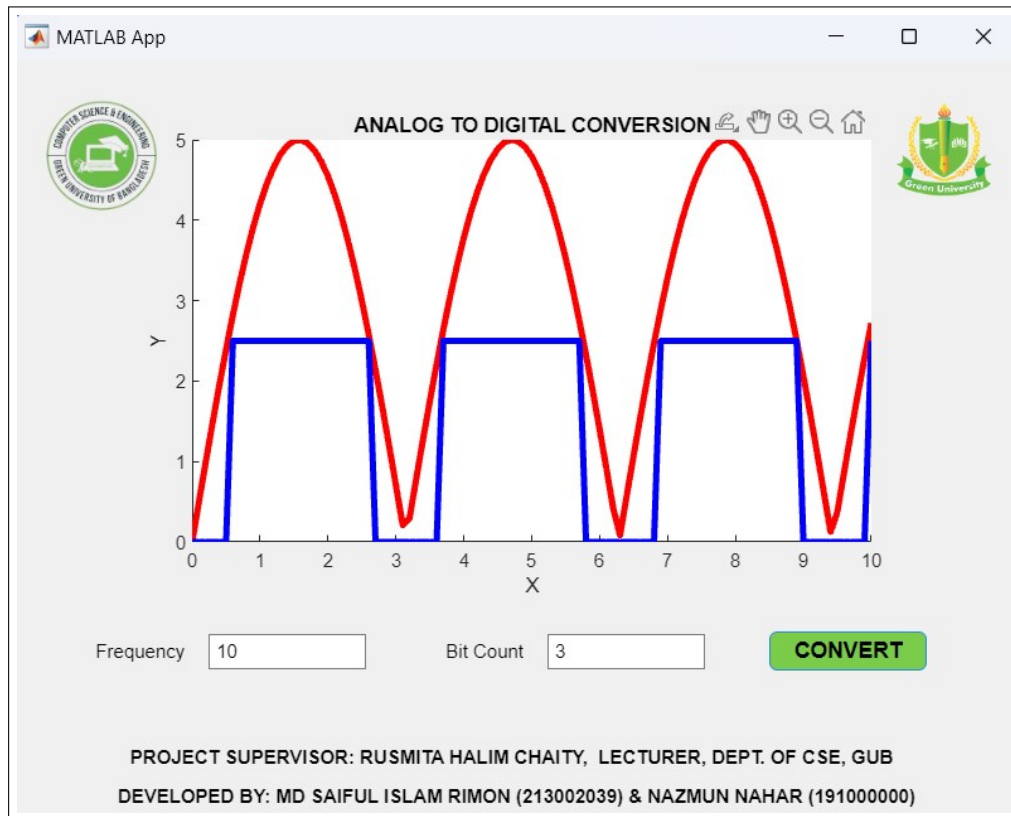


Figure 3.10: Graphical User Interface: Analog to Digital Signal Conversion Page

### 3.2.5 Character Stuffing & Destuffing

After clicking on the Character Stuffing & Destuffing button on the home page, we will see a graphical user interface like the picture below. Here, we can input the string. After clicking the button Stuff, then it show the stuffed string and then if we input the stuffed string and if we click Destuff button, then it will show the destuffed string.

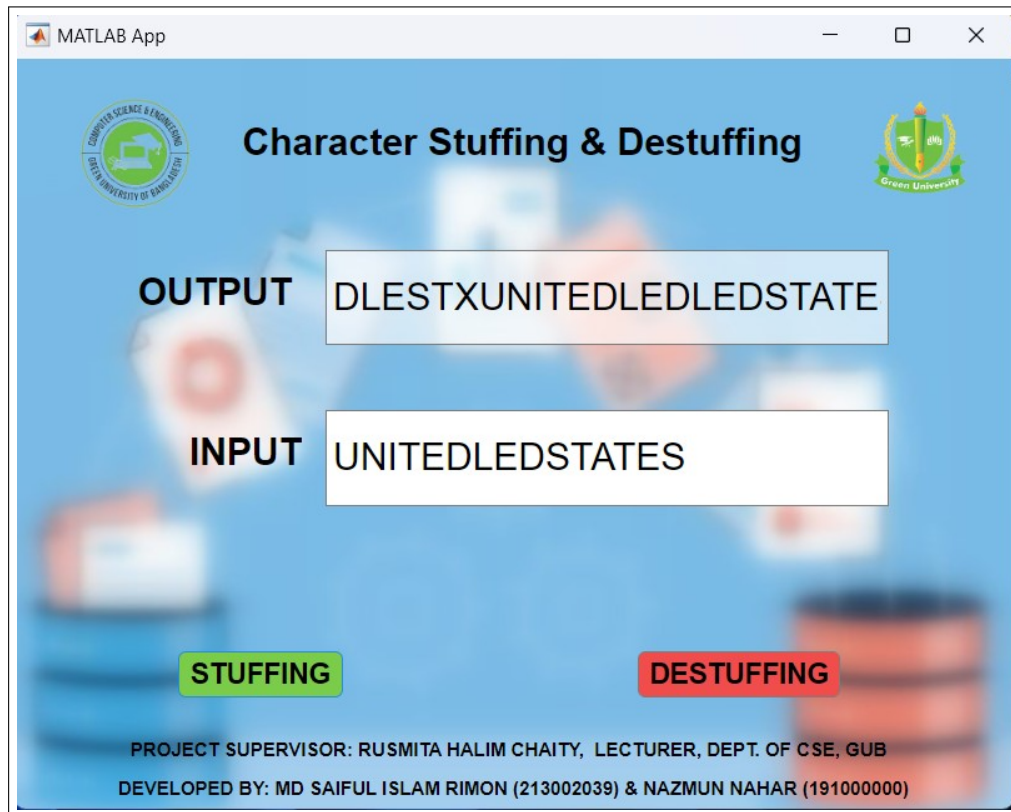


Figure 3.11: Graphical User Interface: Character Stuffing & Destuffing Page: Stuffed String

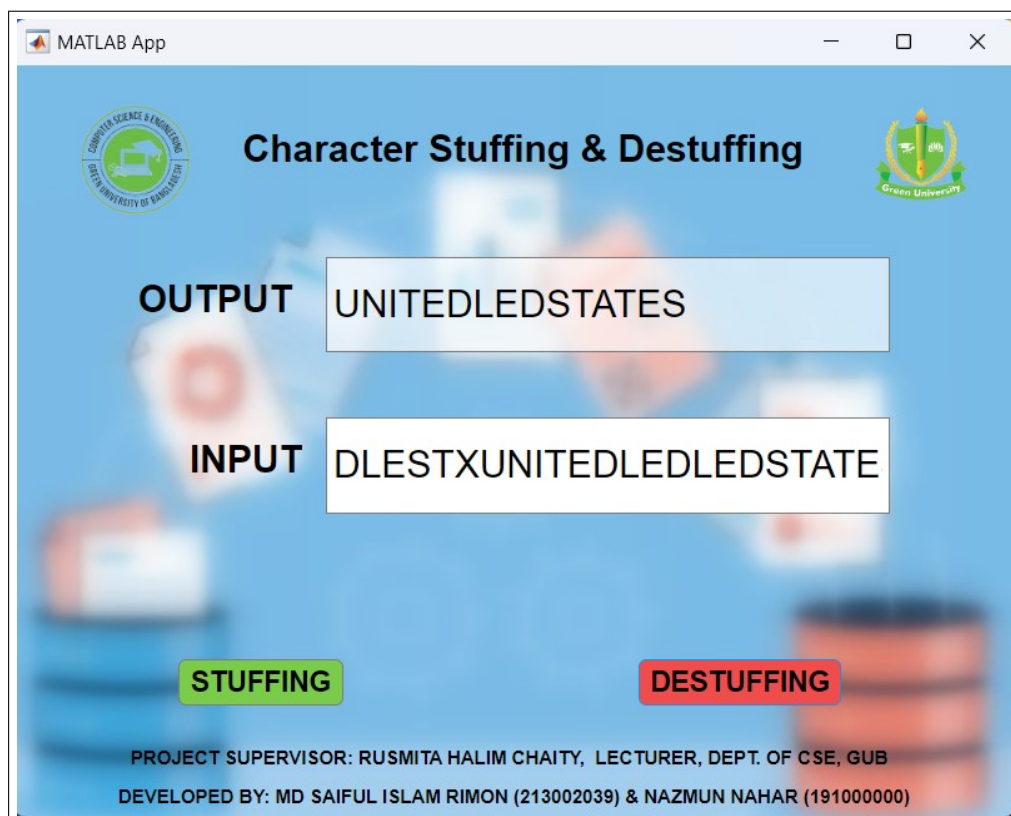


Figure 3.12: Graphical User Interface: Character Stuffing & Destuffing Page: Destuffed String

### 3.2.6 IPv4 Conversion

After clicking on the IPv4 Conversion button on the home page, we will see a graphical user interface like the picture below. Here, we can input the decimal IP, and convert it into binary IP and vice versa.

**IPv4 CONVERSION**

**Binary IP** 11000000 10101000 00001010 00001010

**Decimal IP**

Input 192.168.10.10

Decimal IP To Binary

Binary IP To Decimal

PROJECT SUPERVISOR: RUSMITA HALIM CHAITY, LECTURER, DEPT. OF CSE, GUB  
DEVELOPED BY: MD SAIFUL ISLAM RIMON (213002039) & NAZMUN NAHAR (191000000)

Figure 3.13: Graphical User Interface: IPv4 Conversion Page: Decimal to Binary

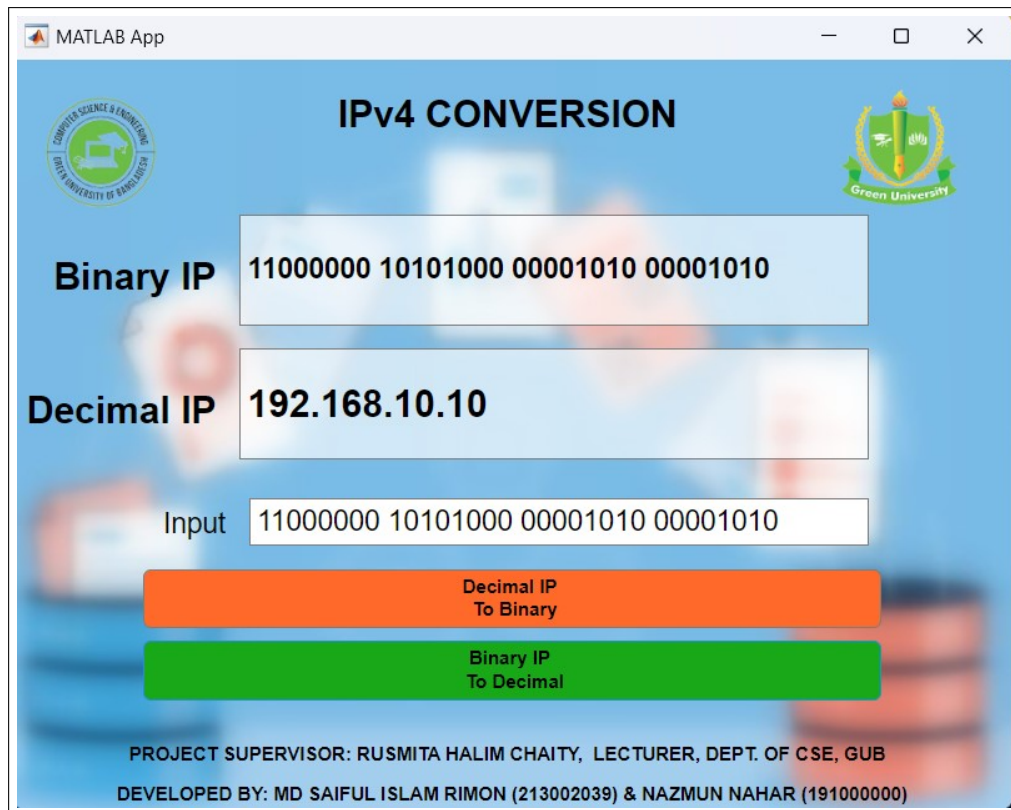


Figure 3.14: Graphical User Interface: IPv4 Conversion Page: Binary to Decimal



### 3.2.7 Cyclic Redundancy Check (CRC)

After clicking on the Cyclic Redundancy Check (CRC) button on the home page, we will see a graphical user interface like the picture below. Here, we can input the bit-string and the polynomial in binary, then it will generate the transmitted codeword. If the receiver codeword is same with the transmitted codeword, then will show "No Error Detected" and a provide a green light signal. If it is not same, then an error will found and provide a red light signal.

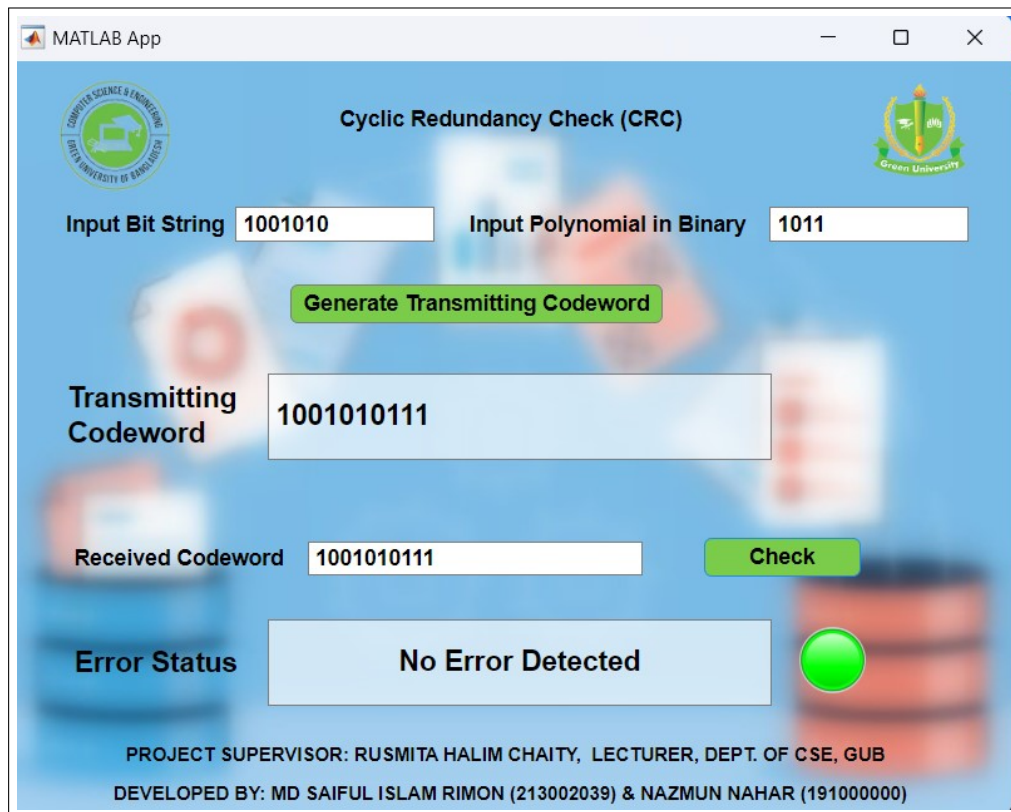


Figure 3.15: Graphical User Interface: Cyclic Redundancy Check (CRC): No Error Detected



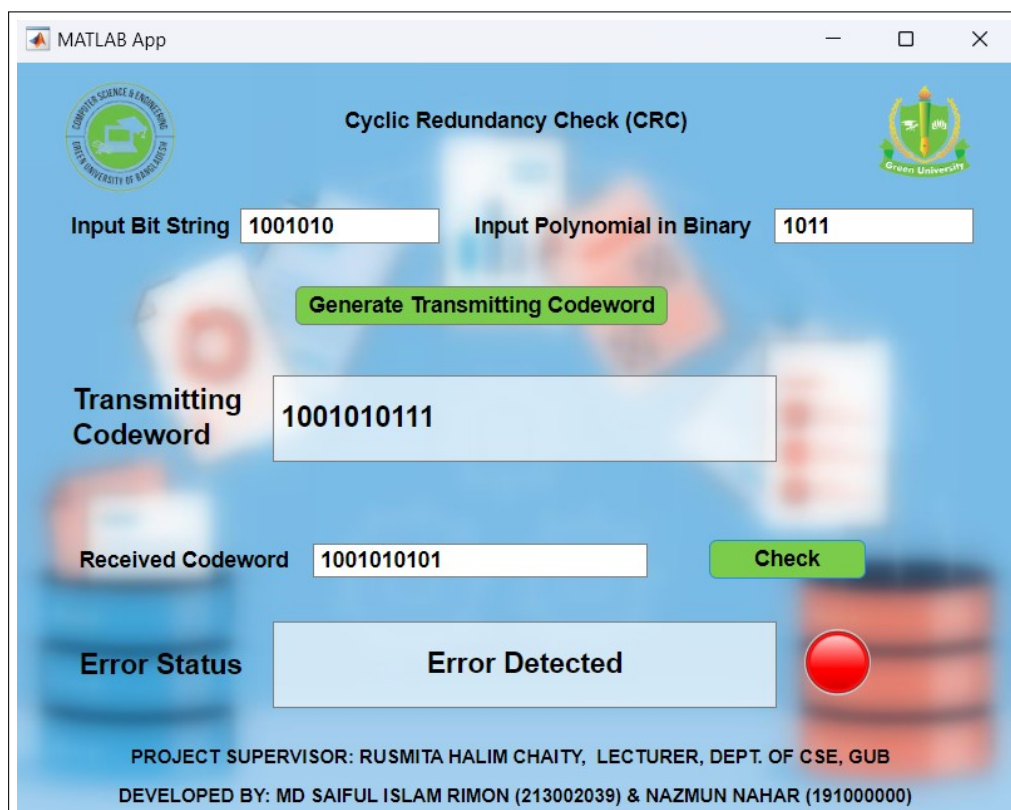


Figure 3.16: Graphical User Interface: Cyclic Redundancy Check (CRC): Error Detected

### 3.3 Results Overall Discussion

The project successfully delivers a comprehensive and user-friendly Data Transmission Simulator system. Its intuitive interface, combined with essential features like Line Coding, Hamming Encoding-Decoding, Analog to Digital Signal Conversion, Character Stuffing & Destuffing, IPv4 Conversion, Cyclic Redundancy Check (CRC)- demonstrates a well-thought-out design and implementation. The project stands as a robust solution to streamline and simplify data transmission tasks.

#### 3.3.1 Complex Engineering Problem Discussion

The complex engineering problem presented in the project involves developing a Data Transmission Simulator using Matlab language. This task is intricate due to several factors discussed below:

##### 1. Depth of Knowledge Required:

- **Understanding of Digital Communication Concepts:** The project encompasses various elements of digital communication, such as line coding, signal conversion, and error detection. A solid foundation in these topics is necessary to comprehend the underlying principles and objectives of each module.
- **Proficiency in MATLAB Programming:** Implementing these features requires an in-depth understanding of MATLAB's programming environment, including its GUI development tools (such as App Designer or GUIDE), built-in functions, and debugging techniques.
- **Algorithm Development Skills:** The project developer must be capable of translating communication theory into practical algorithms, which requires knowledge of data structures, control flow, and MATLAB's numerical capabilities.

##### 2. Depth of Analysis Required:

- **Algorithmic Efficiency:** Each feature, like Hamming Encoding & Decoding or CRC Error Detection, must be analyzed for algorithmic efficiency. This involves choosing the most appropriate algorithms that balance time complexity and resource usage.
- **User Interface Design:** An analysis of the user interface is required to ensure that the application is intuitive and user-friendly. This includes the layout of input and output boxes, as well as the graphical demonstration system for signals.
- **Error Handling and Data Validation:** The developer needs to analyze the potential points of failure and implement robust error handling and data validation to ensure the application behaves correctly with unexpected or erroneous input.

### 3. Extent of Applicable Codes:

- **Modular Programming:** The code for each feature (a-g) must be modular, allowing for easy updates and maintenance. The extent of the code will include separate functions or classes for each feature, following MATLAB's best practices for code organization.
- **GUI Component Integration:** The extent of applicable codes also includes the integration of GUI components such as buttons, input boxes, and display panels. It requires understanding of callback functions and event-driven programming in MATLAB.
- **Data Processing and Visualization:** For the graphical demonstration system, the code must include data processing algorithms that can convert raw input into a visual format, using MATLAB's plotting functions and graphics system.
- **Compliance with Standards:** Where applicable, the code must adhere to industry standards for digital communication, particularly for encoding, decoding, and error detection methods.

Addressing these complex problems involves a multi-disciplinary approach that combines technical knowledge with analytical skills. The success of such a project lies in the ability to translate theoretical concepts into practical, working solutions that are both efficient and user-friendly.

# Chapter 4

## Conclusion

### 4.1 Discussion

This MATLAB project presents an effective user interface for exploring key digital communication techniques such as Line Coding, Hamming Encoding Decoding, and Analog to Digital Signal Conversion. The inclusion of error detection methods like Hamming codes and CRC is particularly beneficial for understanding data integrity in communication systems. Additionally, the tools for Character and Bit Stuffing Destuffing provide practical insights into data framing, crucial for network communications. The IPv4 Conversion feature aligns with current networking practices, enhancing the software's professional applicability.

The software's design, featuring clear input and output boxes and graphical signal demonstrations, ensures ease of use, making it accessible for learners at various levels. By combining educational value with user-friendly functionality, the program serves as a comprehensive tool for both students and professionals in digital communications.

Future iterations could expand on advanced topics and user feedback integration to further enrich learning experiences and practical application.

### 4.2 Limitations

1. **Bugs in Hamming Encoding & Decoding:** The current Hamming Encoding & Decoding used in this application has several bugs on the receiver side's parity bit and checking the position of error and correction of the bit-stream.
2. **Character Stuffing & Destuffing:** The starting flag and ending flag is fixed here. That's mean, the user cannot set it. And the escaped code is also fixed here, which user cannot change. Also, there are some bugs in destuffing the string.
3. **Error Handling and Validation:** The program has limited error handling capabilities, particularly in dealing with invalid user inputs or overflow scenarios, which could lead to system crashes or incorrect processing.

4. **Hardware and Software Dependency:** Being written in Matlab language, the application is highly dependent on specific software environments, limiting its portability.

## 4.3 Scope of Future Work

1. **Development of the existing Graphical User Interface (GUI):** Enhance the currently running GUI would greatly enhance user experience and accessibility, making the system more appealing and easier to navigate.
2. **Fixing the algorithmic bugs:** Fixing the algorithmic bugs of Hamming Encoding & Decoding, Character Stuffing & Destuffing and others.
3. **Robust Error Handling and Input Validation:** Developing more sophisticated error handling and validation mechanisms to prevent crashes and ensure system stability, especially under erroneous or unexpected inputs.
4. **Incorporating Advanced Reporting and Analytics:** Adding functionality for detailed reporting and analytics would enable better tracking of parking patterns, financial management, and strategic planning.
5. **Cross-Platform Compatibility:** Rewriting or adapting the code to be compatible with multiple platforms and architectures would significantly increase the system's applicability and reach.
6. **Integration with Additional Services:** Future developments could include integration online version, or mobile version to enhance functionality and user experience.

## References