

APPENDIX A

(SOURCE CODE)

```
import java.awt.*; im-
port java.awt.event.*;
import java.util.ArrayList;

public class AlumniManagementSystem extends Frame {

    // Components
    Label titleLabel, nameLabel, batchLabel, emailLabel, yearLabel, eventLabel, messageLabel;
    TextField nameField, batchField, emailField, yearField, eventField;
    Button registerAlumniButton, viewAlumniButton, searchAlumniButton, createEventButton,
viewEventsButton, exitButton;
    TextArea outputArea;

    // Data Storage
    ArrayList<String> alumniList = new ArrayList<>(); Ar-
rayList<String> eventList = new ArrayList<>();

    public AlumniManagementSystem() {
        // Frame settings
        setTitle("Alumni Management System"); set-
        Size(600, 600);
        setLayout(null); setBackground-
        ground(Color.LIGHT_GRAY);

        // Title
        titleLabel = new Label("Alumni Management System");
        titleLabel.setBounds(200, 30, 300, 30); titleLa-
        bel.setFont(new Font("Arial", Font.BOLD, 18)); add(ti-
        tleLabel);

        // Labels and Input Fields
```

```
nameLabel = new Label("Name:"); nameLabel.set-  
Bounds(50, 80, 100, 20); add(name-  
Label);
```

```
nameField = new TextField(); nameField.set-  
Bounds(150, 80, 200, 20); add(nameField);
```

```
batchLabel = new Label("Batch:"); batchLa-  
bel.setBounds(50, 110, 100, 20); add(batchLa-  
bel);
```

```
batchField = new TextField(); batchField.set-  
Bounds(150, 110, 200, 20); add(batchField);
```

```
emailLabel = new Label("Email:"); email-  
Label.setBounds(50, 140, 100, 20); add(email-  
Label);
```

```
emailField = new TextField(); emailField.set-  
Bounds(150, 140, 200, 20); add(emailField);
```

```
yearLabel = new Label("Year of Graduation:");  
yearLabel.setBounds(50, 170, 150, 20);  
add(yearLabel);
```

```
yearField = new TextField(); yearField.set-  
Bounds(200, 170, 150, 20); add(yearField);
```

```
eventLabel = new Label("Event:");  
eventLabel.setBounds(50, 200, 100, 20);  
add(eventLabel);
```

```

eventField = new TextField(); eventField.set-
Bounds(150, 200, 200, 20); add(eventField);

// Buttons
registerAlumniButton = new Button("Register Alumni");
registerAlumniButton.setBounds(50, 240, 150, 30);
add(registerAlumniButton);

viewAlumniButton = new Button("View All Alumni");
viewAlumniButton.setBounds(210, 240, 150, 30);
add(viewAlumniButton);

searchAlumniButton = new Button("Search Alumni by Year");
searchAlumniButton.setBounds(370, 240, 170, 30);
add(searchAlumniButton);

createEventButton = new Button("Create Event"); cre-
ateEventButton.setBounds(50, 280, 150, 30); add(cre-
ateEventButton);

viewEventsButton = new Button("View All Events");
viewEventsButton.setBounds(210, 280, 150, 30);
add(viewEventsButton);

exitButton = new Button("Exit"); exitBut-
ton.setBounds(370, 280, 150, 30); add(exitBut-
ton);

// Output Area
outputArea = new TextArea(); outputA-
rea.setBounds(50, 330, 500, 200); outputA-
rea.setEditable(false); add(outputArea);

```

```

// Message Label messageLa-
bel = new Label("");
messageLabel.setBounds(50, 540, 500, 20); add(mes-
sageLabel);

// Event Listeners
registerAlumniButton.addActionListener(e -> registerAlumni()); viewAlumniButton.ad-
dActionListener(e -> viewAllAlumni()); searchAlumniButton.addActionListener(e ->
searchAlumniByYear()); createEventButton.addActionListener(e -> createEvent());
viewEventsButton.addActionListener(e -> viewAllEvents()); exitButton.addAction-
Listener(e -> System.exit(0));

// Window Listener to Close addWin-
dowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e) { dis-
        pose();
    }
});
}

// Register Alumni
private void registerAlumni() {
    String name = nameField.getText();
    String batch = batchField.getText();
    String email = emailField.getText();
    String year = yearField.getText();

    if (!name.isEmpty() && !batch.isEmpty() && !email.isEmpty() && !year.isEmpty()) {
        alumniList.add("Name: " + name + ", Batch: " + batch + ", Email: " + email + ", Year: " +
year);
        messageLabel.setText("Alumni registered successfully!"); clear-
Fields();
    } else {

```

```

        messageLabel.setText("All fields are required for alumni registration!");
    }
}

// View All Alumni
private void viewAllAlumni() { outputArea.setText("Registered Alumni:\n"); for
    (String alumni : alumniList) {
        outputArea.append(alumni + "\n");
    }
}

// Search Alumni by Year of Graduation private void searchAlumniByYear() {
    String year = yearField.getText();
    outputArea.setText("Alumni Graduated in Year " + year + ":\n");
    for (String alumni : alumniList) {
        if (alumni.contains("Year: " + year)) { outputArea.append(alumni + "\n");
        }
    }
}

// Create Event
private void createEvent() {
    String event = eventField.getText();
    if (!event.isEmpty()) {
        eventList.add(event);
        messageLabel.setText("Event created successfully!");
        eventField.setText("");
    } else {
        messageLabel.setText("Event field cannot be empty!");
    }
}
}

```

```

// View All Events
private void viewAllEvents() { outputA-
    rea.setText("Upcoming Events:\n"); for
    (String event : eventList) {
        outputArea.append(event + "\n");
    }
}

// Clear Input Fields private
void clearFields() {
    nameField.setText("");
    batchField.setText("");
    emailField.setText("");
    yearField.setText("");
}

// Main Method
public static void main(String[] args) {
    new AlumniManagementSystem().setVisible(true);
}
}

```