

**SAGI RAMA KRISHNAM RAJU ENGINEERING COLLEGE
(AUTONOMOUS)
DEPARTMENT OF INFORMATION TECHNOLOGY**

***ANDROID APPLICATION DEVELOPMENT LAB
RECORD***

SAGI RAMA KRISHNAM RAJU ENGINEERING COLLEGE
(AUTONOMOUS)
DEPARTMENT OF INFORMATION TECHNOLOGY

SNo	EXPERIMENT	PAGE Nos
1.	Create an Android application that shows Hello + name of the user and run it on an emulator	
2.	Create an application that takes the name from a text box and shows hello message along with the name entered in text box, when the user clicks the OK button.	
3.	Create a screen that has input boxes for User Name, Password, Address, Gender (radio buttons for male and female), Age (numeric), Date of Birth (Date Picket), State (Spinner) and a Submit button. On clicking the submit button, print all the data below the Submit Button. Use a. Linear Layout, b. Relative Layout and c. Grid Layout or Table Layout	
4.	Develop an application that shows names as a list and on selecting a name it should show the details of the candidate on the next screen with a "Back" button. If the screen is rotated to landscape mode (width greater than height), then the screen should show list on left fragment and details on right fragment instead of second screen with back button. Use Fragment transactions and Rotation event listener.	
5.	Develop an application that uses a menu with 3 options for dialing a number, opening a website and to send an SMS. On selecting an Option, the appropriate action should be invoked using intents.	
6.	Develop an application that inserts some notifications into Notification area and whenever a notification is inserted, it should show a toast with details of the notification.	
7.	Create an application that uses a text file to store user names and passwords (tab separated fields and one record per line). When the user submits a login name and password through a screen, the details should be verified with the text file data and if they match, show a dialog saying that login is successful. Otherwise, show the dialog with Login Failed message	
8.	Create a user registration application that stores the user details in a database table.	
9.	Create a database and a user table where the details of login names and passwords are stored. Insert some names and passwords initially. Now the login details entered by the user should be verified with the database and an appropriate dialog should be shown to the user.	

Course Objectives: This Course will enable students to

1.	To learn how to develop Applications in android environment.
2.	To learn how to develop user interface applications.
3.	To learn how to develop URL related applications.

Introduction

User Interface (UI) Applications:

Adobe XD:

Description: Adobe XD is a vector-based design tool for creating user interfaces and user experiences for web and mobile applications.

Use Cases: Prototyping, wireframing, and designing UI/UX.

Sketch:

Description: Sketch is a design tool specifically for UI and UX design, popular among macOS users.

Use Cases: Creating digital interfaces, icons, and prototypes.

Figma:

Description: Figma is a collaborative design tool that enables real-time collaboration between team members on UI/UX projects.

Use Cases: Designing, prototyping, and collaborating on user interfaces.

InVision:

Description: InVision is a prototyping tool that helps designers create interactive and animated prototypes for web and mobile applications.

Use Cases: Prototyping, user testing, and collaboration.

URL-Related Applications:

Web Browsers (e.g., Google Chrome, Mozilla Firefox):

Description: Web browsers are software applications that allow users to access and navigate the World Wide Web.

Use Cases: Browsing websites, accessing web applications.

URL Shorteners (e.g., Bitly, TinyURL):

Description: URL shorteners are tools that reduce the length of long URLs, making them easier to share, especially in character-limited contexts.

Use Cases: Sharing links on social media, in emails, or in situations with character limits.

Download Managers (e.g., Internet Download Manager, JDownloader):

Description: Download managers help in organizing and accelerating the download process of files from the internet.

Use Cases: Managing and accelerating downloads from URLs.

Bookmark Managers (e.g., Pocket, Raindrop.io):

Description: Bookmark managers help users organize and save URLs for quick access later.

Use Cases: Saving and organizing bookmarks for websites of interest.

MIT App Inventor

MIT App Inventor is a web-based platform developed by MIT (Massachusetts Institute of Technology) that allows users to create mobile applications for Android devices without needing to know how to code. It provides a visual interface where users can drag and drop components to design the user interface and specify the functionality of their app through a blocks-based programming environment.

Here's a brief overview of how it works:

Visual Interface Design: Users design the layout of their app's user interface by dragging and dropping components such as buttons, text boxes, images, and more onto a design canvas. This process is similar to creating a PowerPoint presentation or designing a webpage.

Blocks Programming: Users define the behavior of their app by using a blocks-based programming language. Instead of writing lines of code, users assemble blocks that represent different actions, events, and functions. These blocks snap together like puzzle pieces to create the logic of the app.

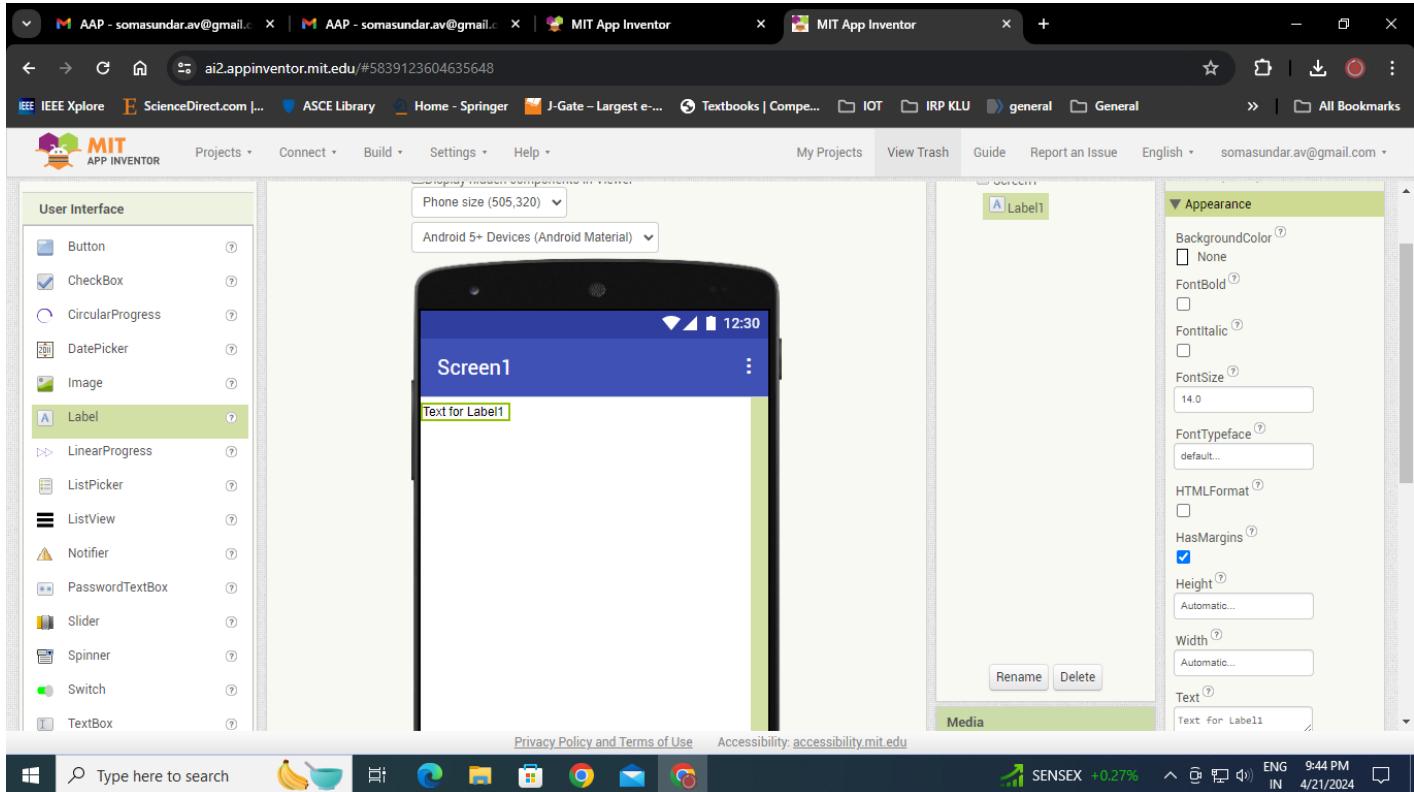
Real-Time Testing: As users build their app, they can test it instantly on their connected Android device or emulator. This allows for rapid iteration and debugging, as users can see how their app behaves in real-time as they make changes.

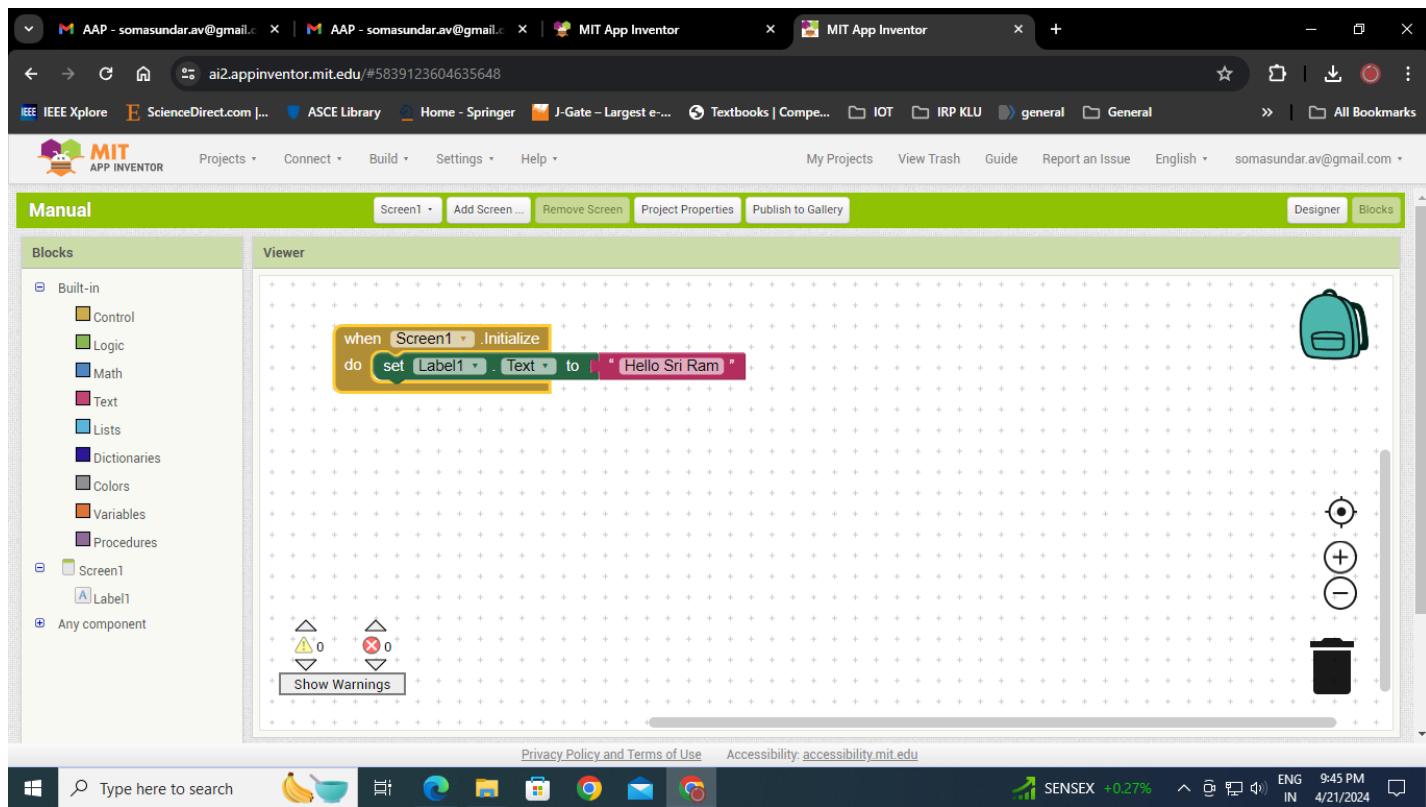
Access to Device Features: MIT App Inventor provides access to various device features and sensors, such as the camera, GPS, accelerometer, and more. Users can incorporate these features into their apps to create a wide range of functionalities.

Experiment 1

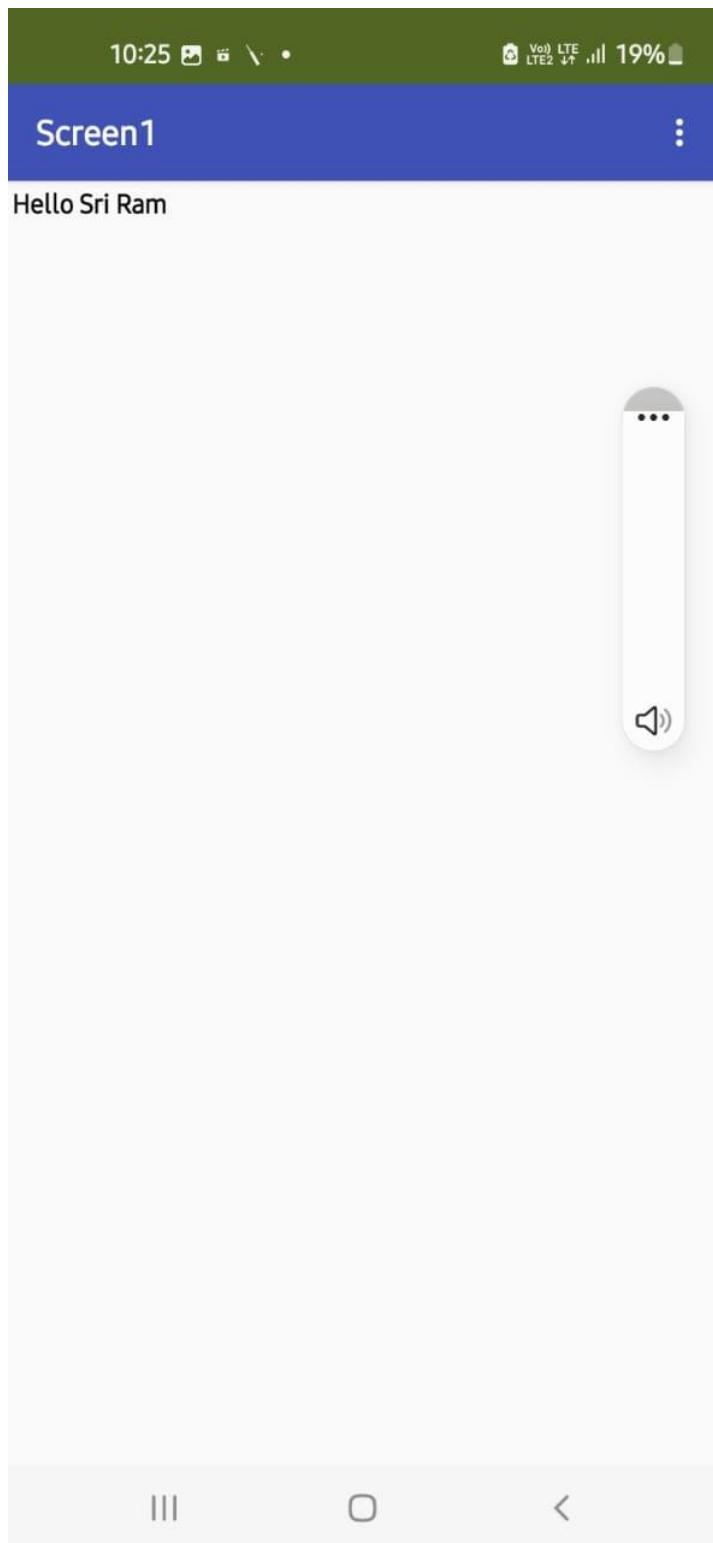
AIM: Create an Android application that shows Hello + name of the user and run it on an emulator.

Program:





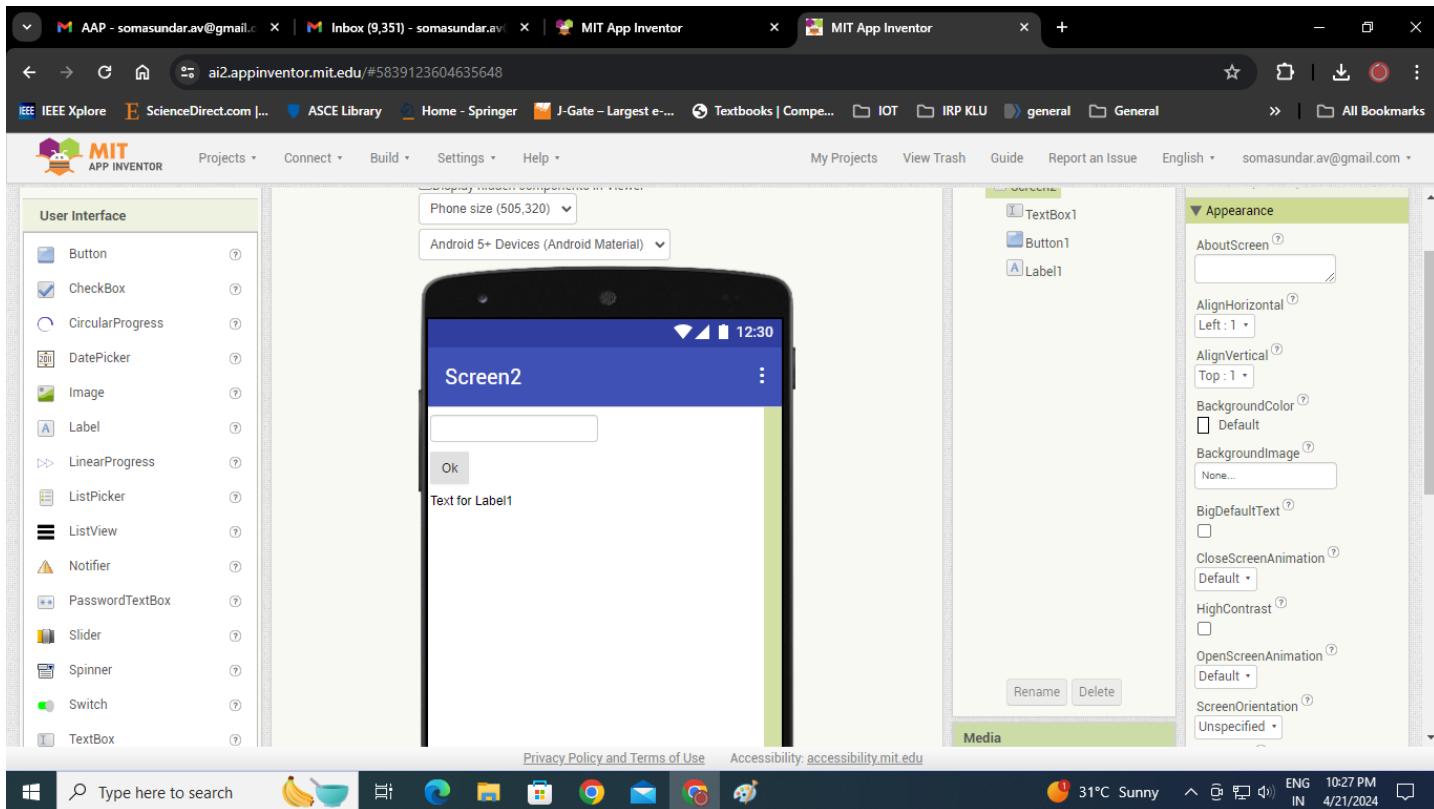
Output:

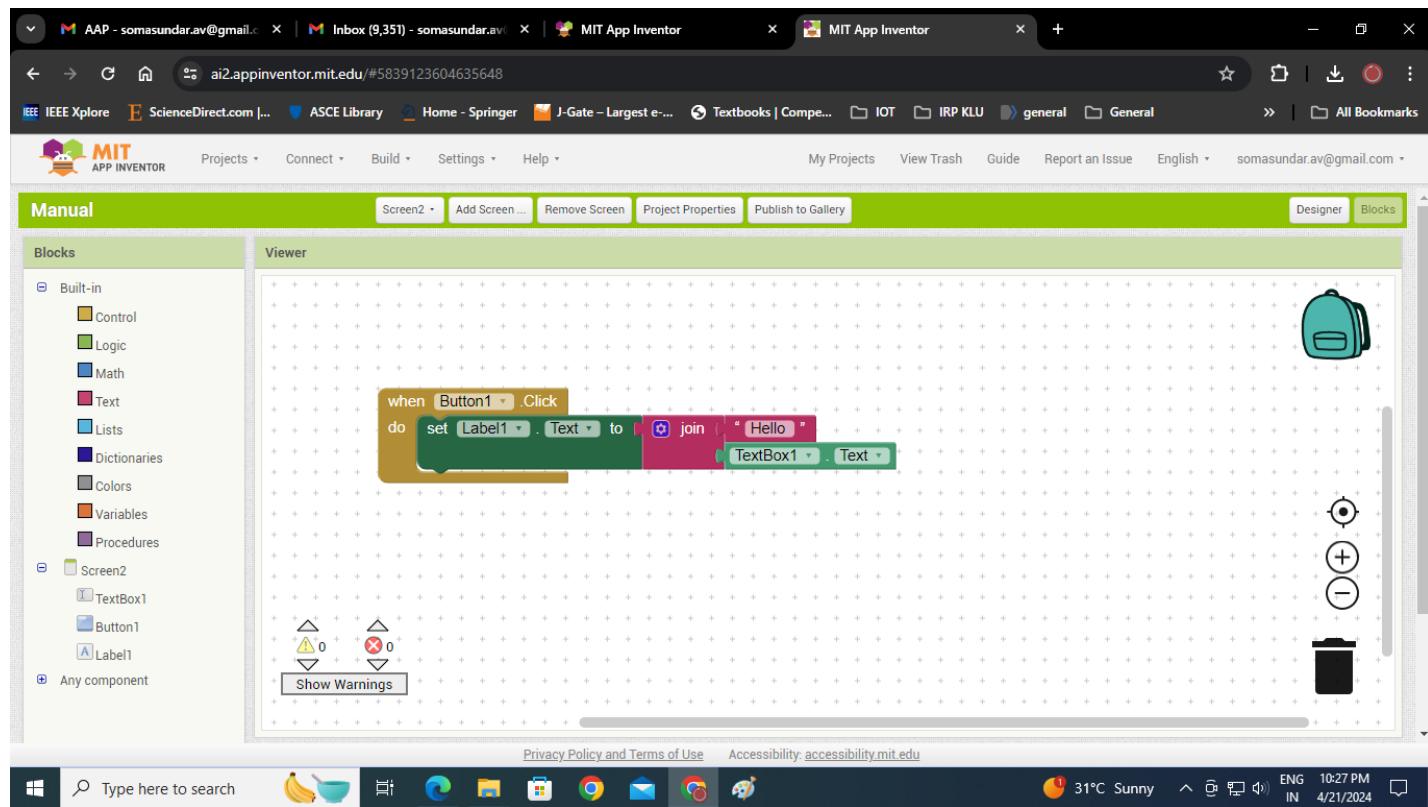


Experiment:2

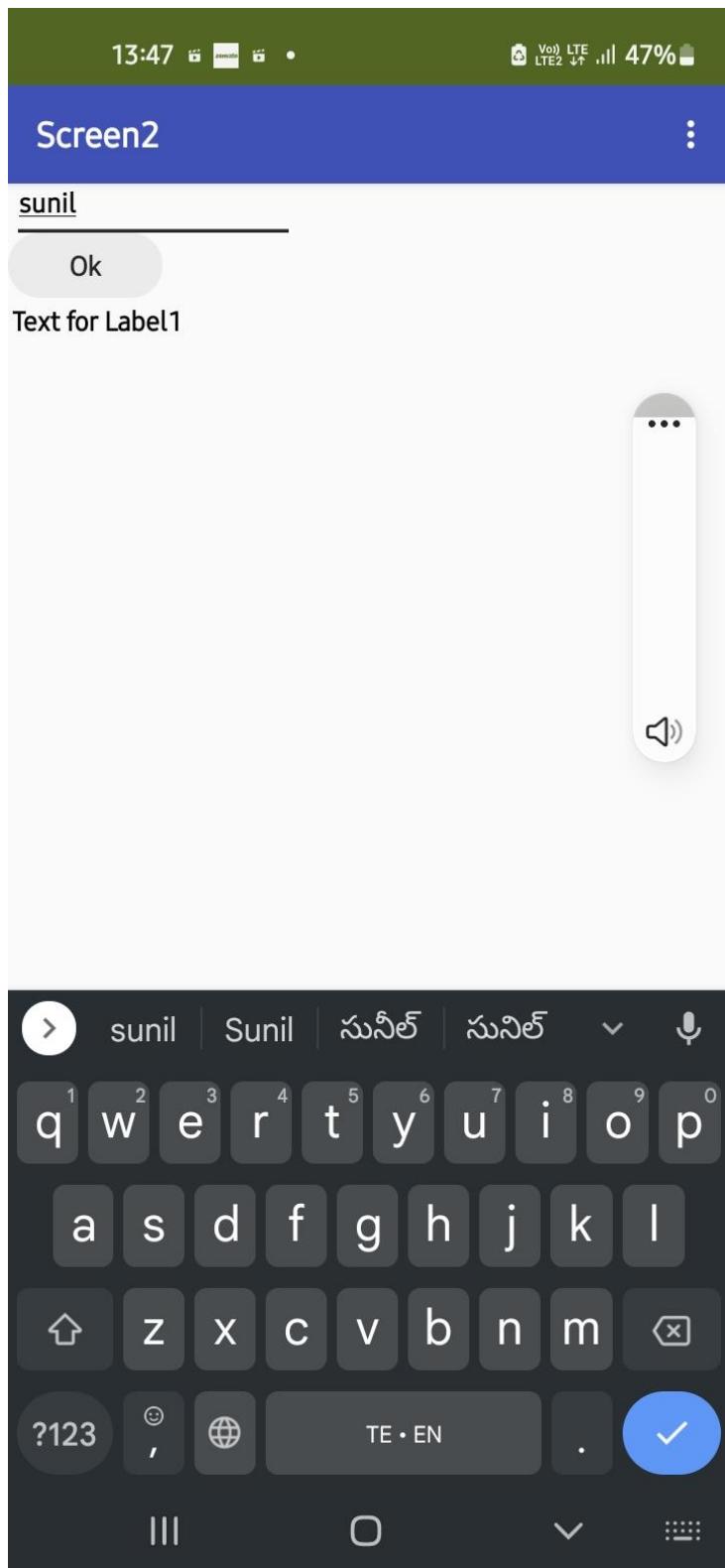
AIM: Create an application that takes the name from a text box and shows hello message along with the name entered in text box, when the user clicks the OK button.

Program:





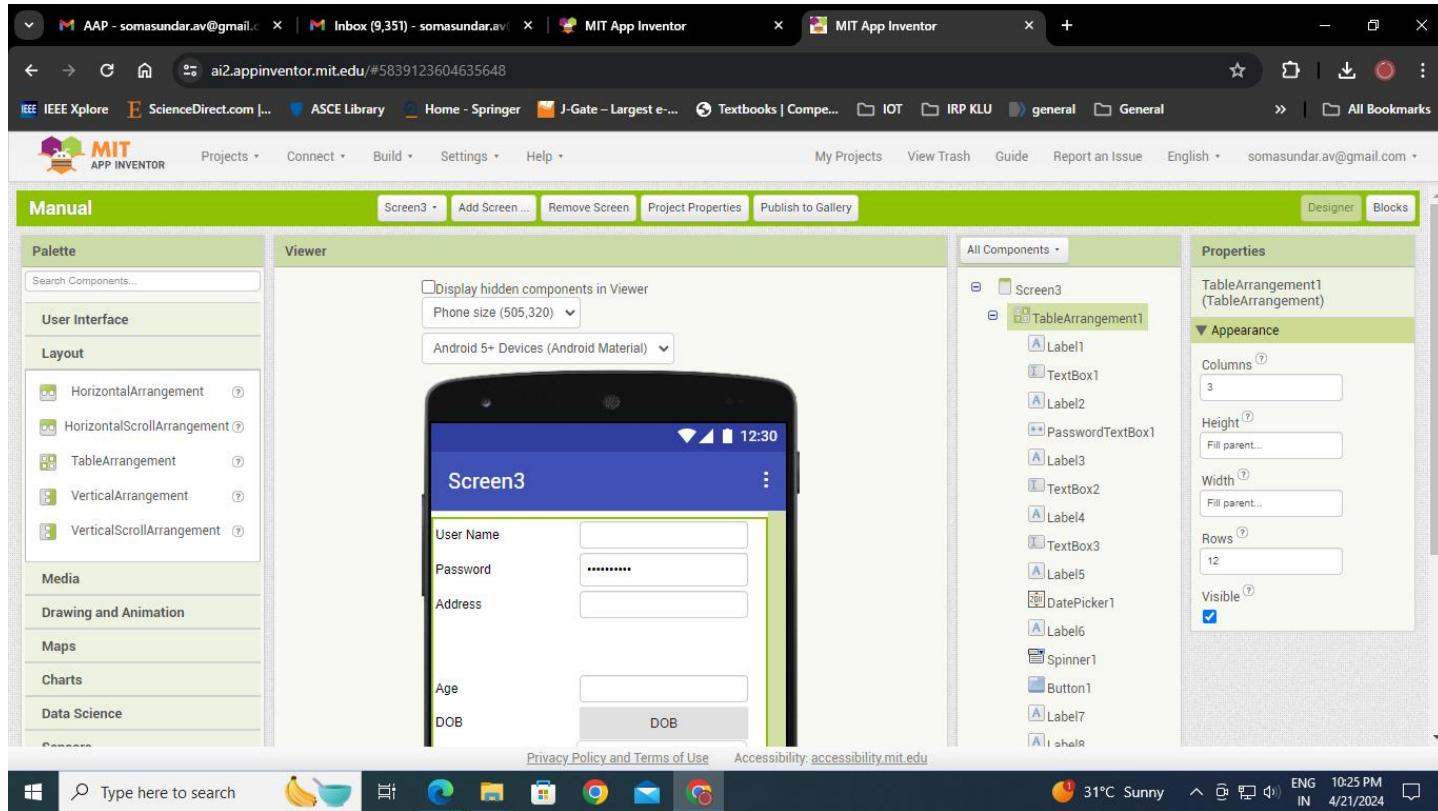
Output:

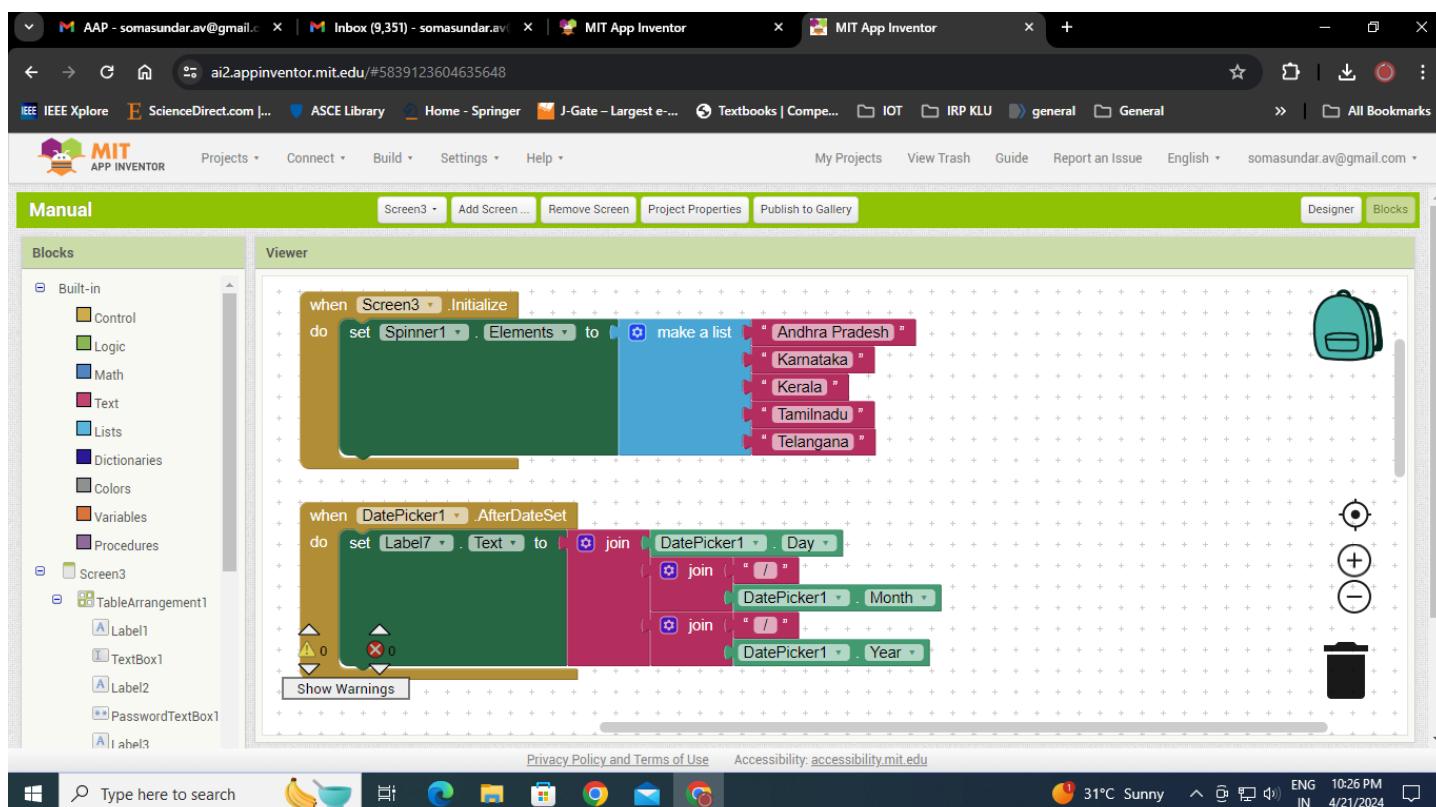
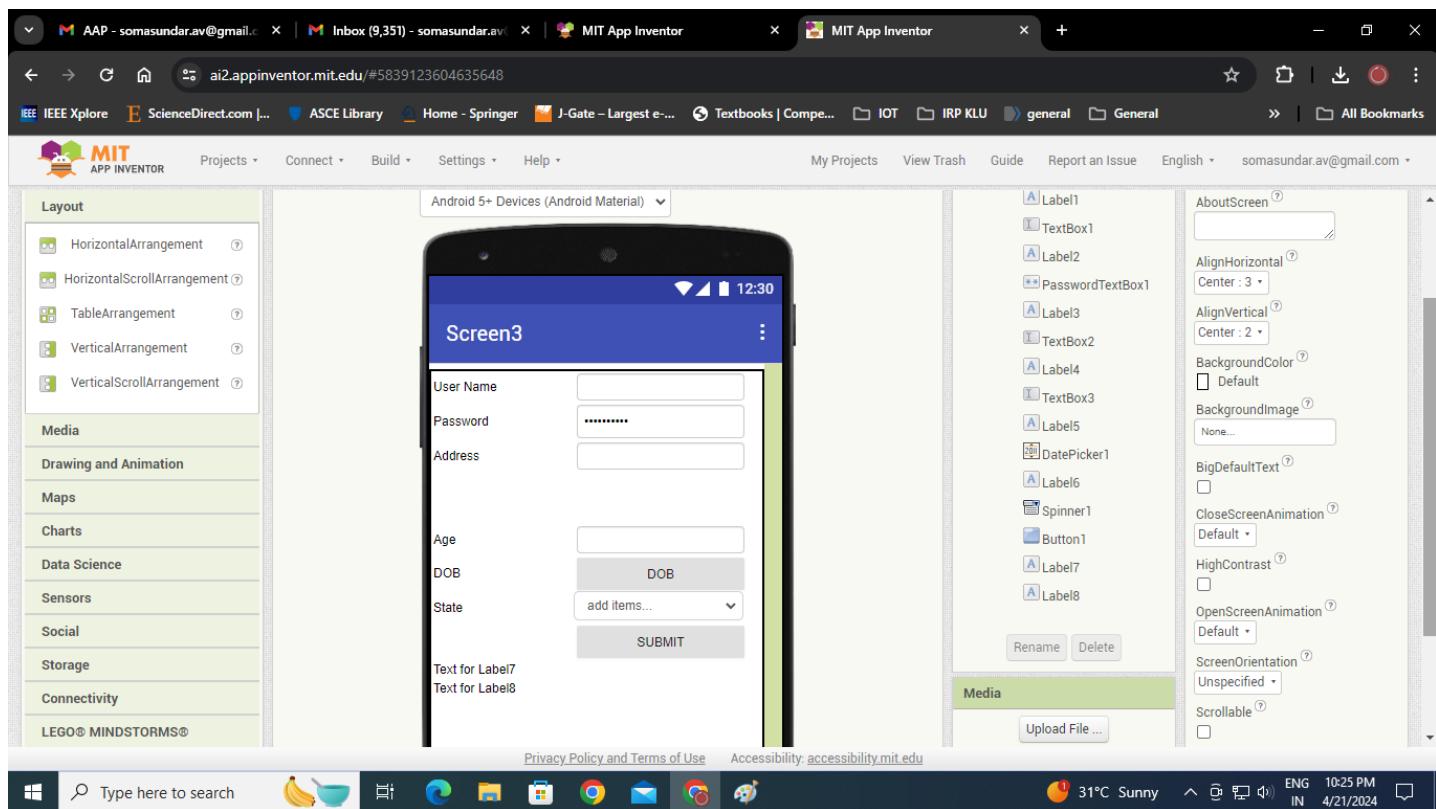


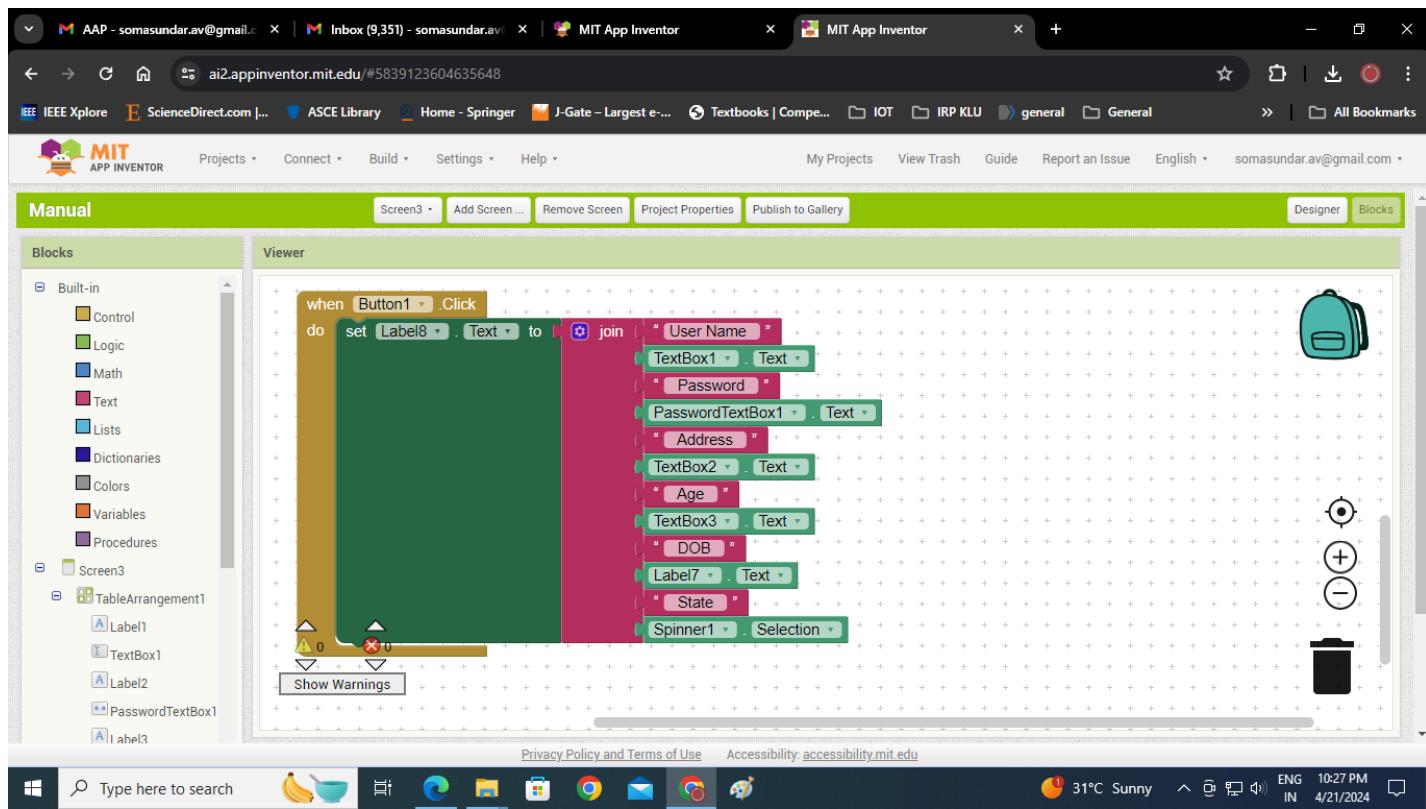
Experiment 3

AIM: Create a screen that has input boxes for User Name, Password, Address, Gender (radio buttons for male and female), Age (numeric), Date of Birth (Date Picket), State (Spinner) and a Submit button. On clicking the submit button, print all the data below the Submit Button. Use a. Linear Layout, b. Relative Layout and c. Grid Layout or Table Layout

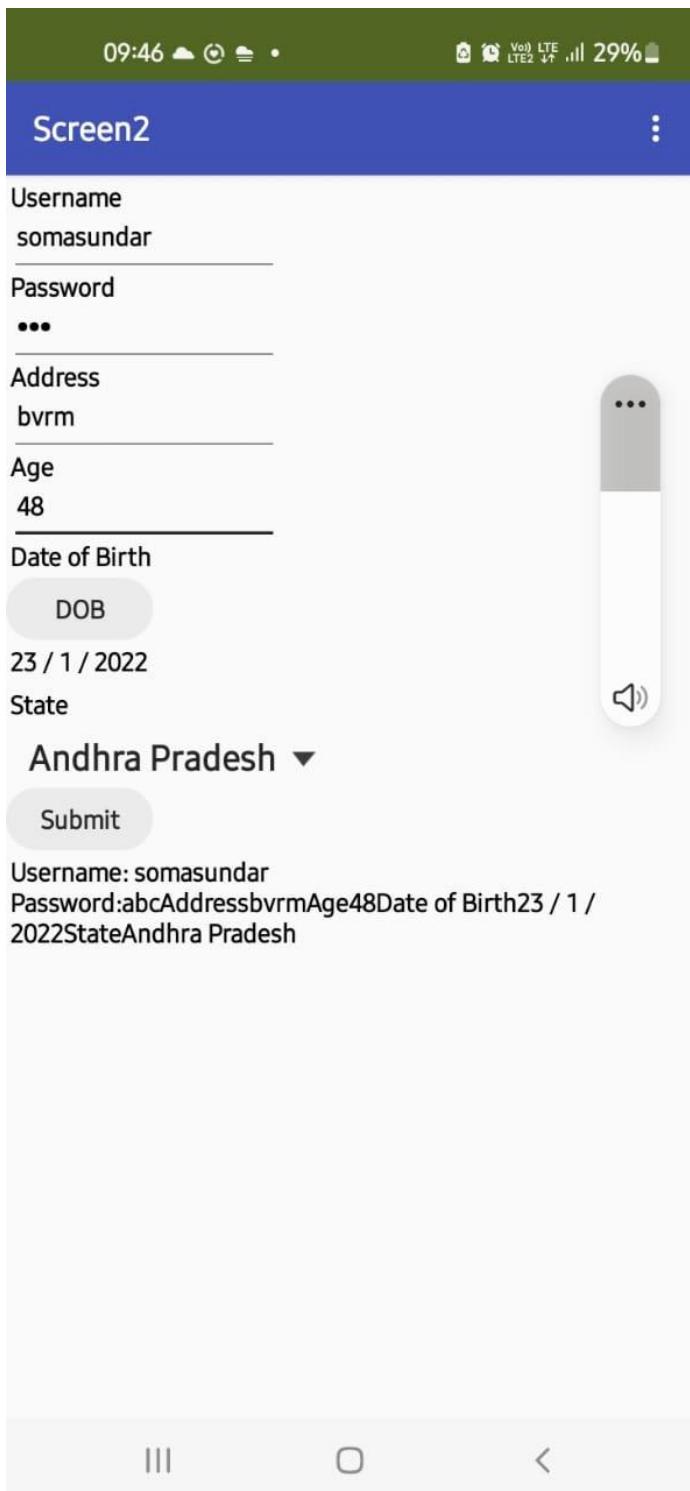
Program:







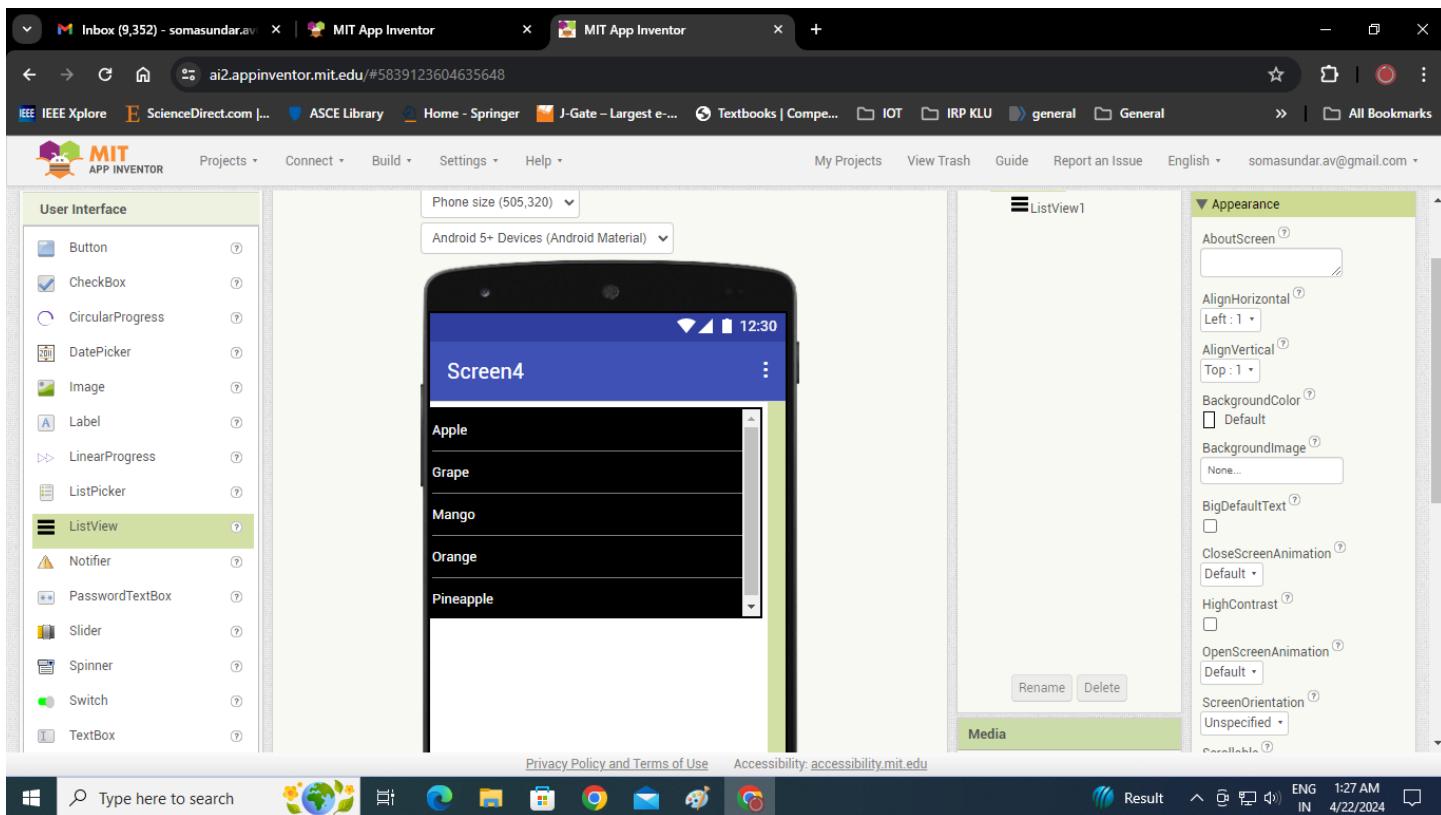
Output:

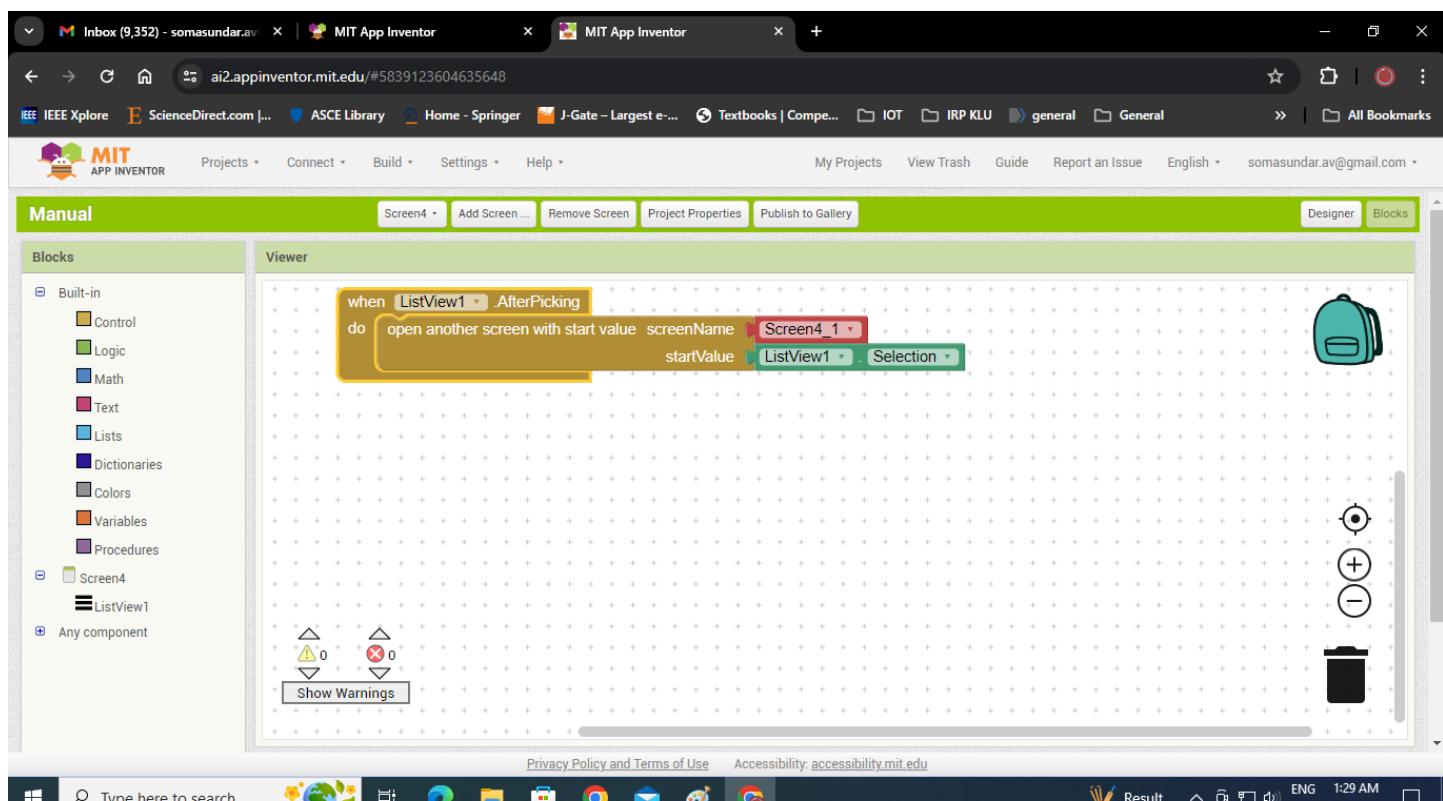
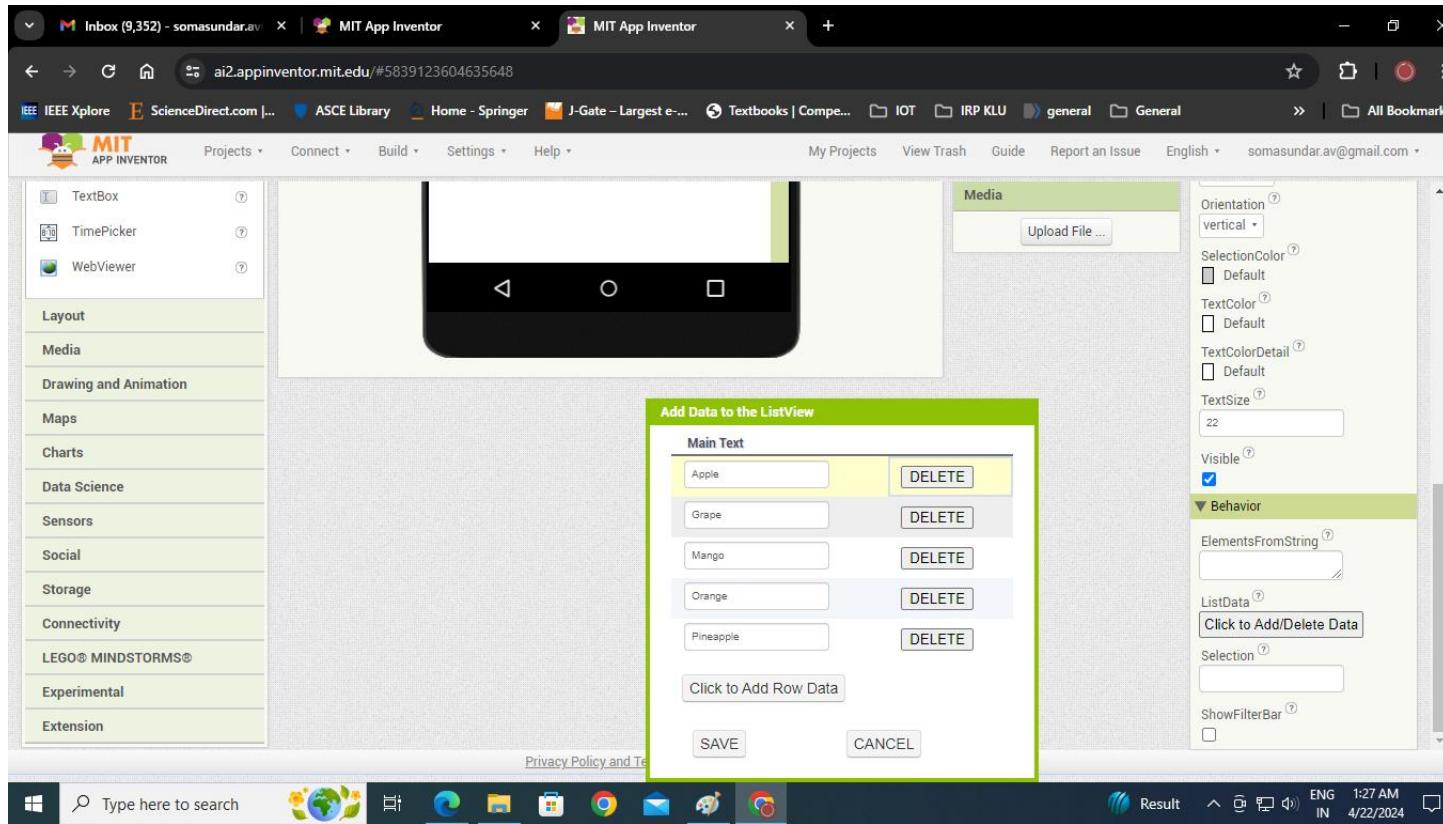


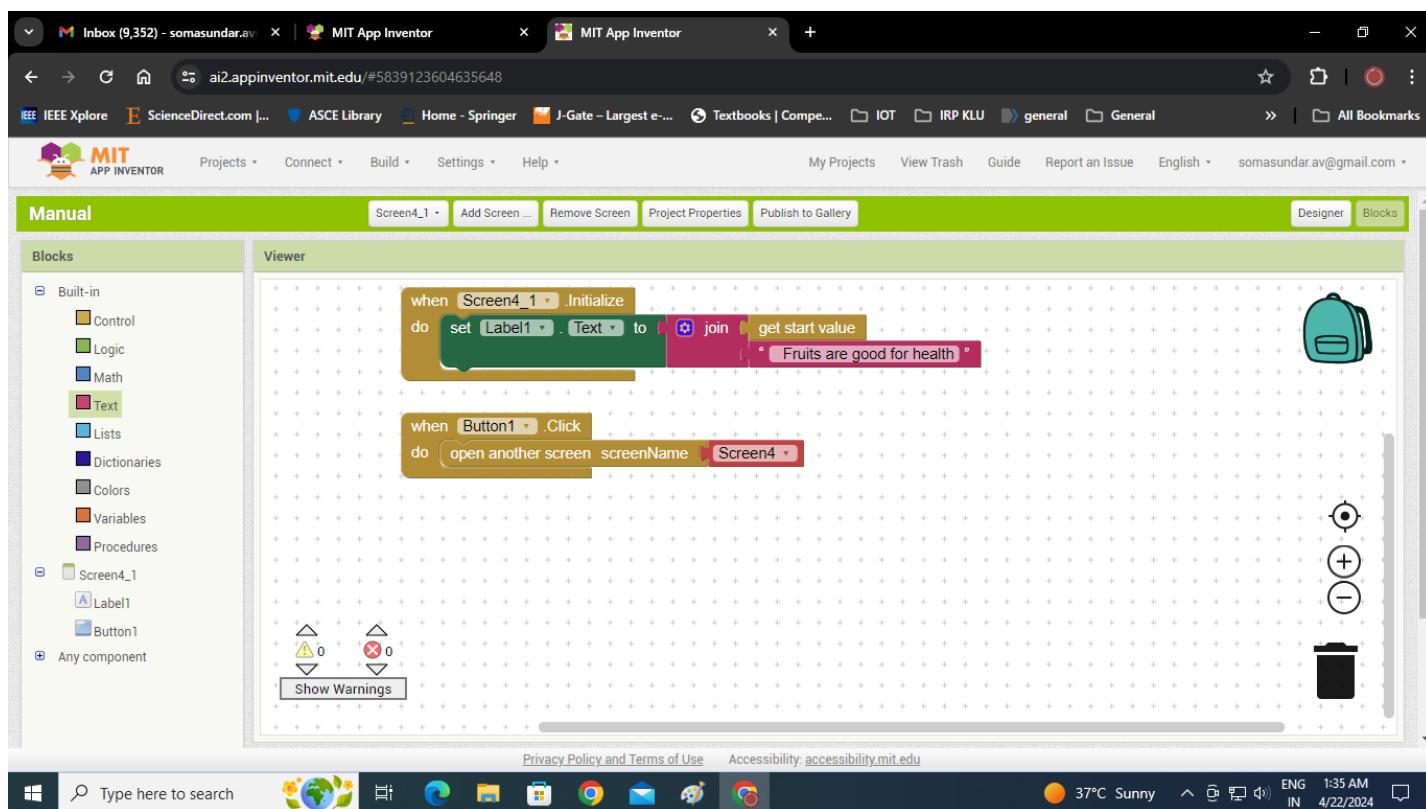
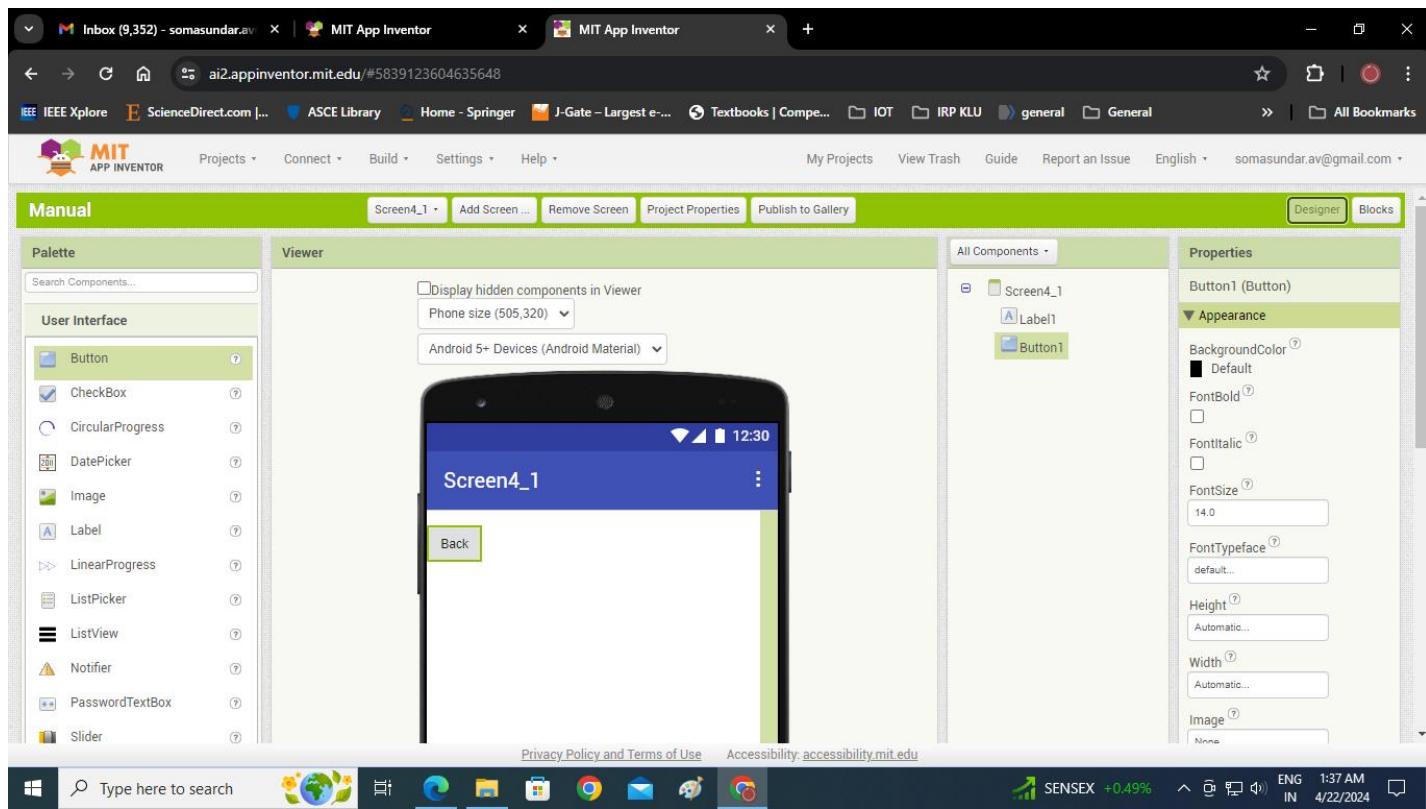
Experiment 4

AIM: Develop an application that shows names as a list and on selecting a name it should show the details of the candidate on the next screen with a “Back” button. If the screen is rotated to landscape mode (width greater than height), then the screen should show list on left fragment and details on right fragment instead of second screen with back button. Use Fragment transactions and Rotation event listener.

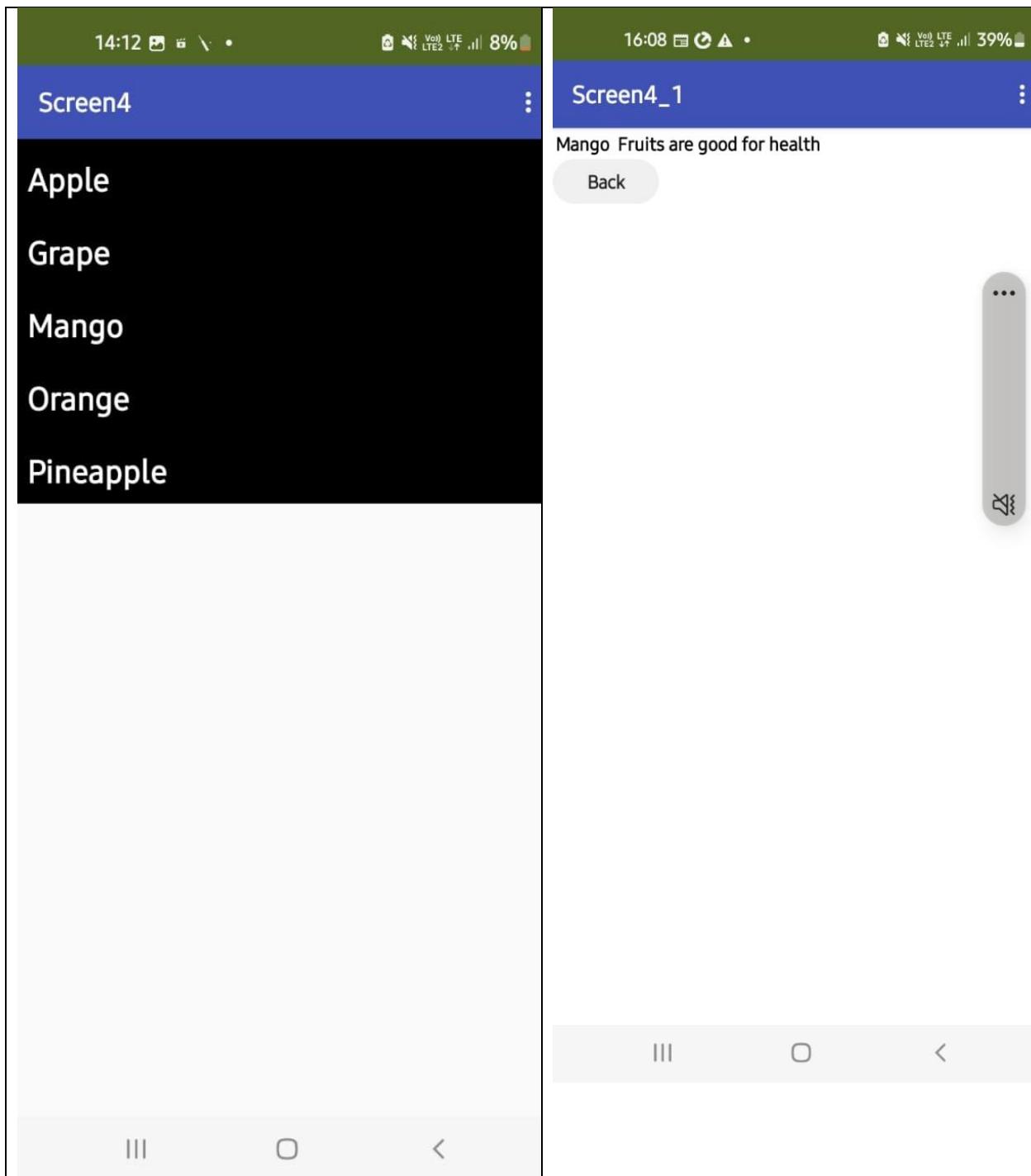
Program:







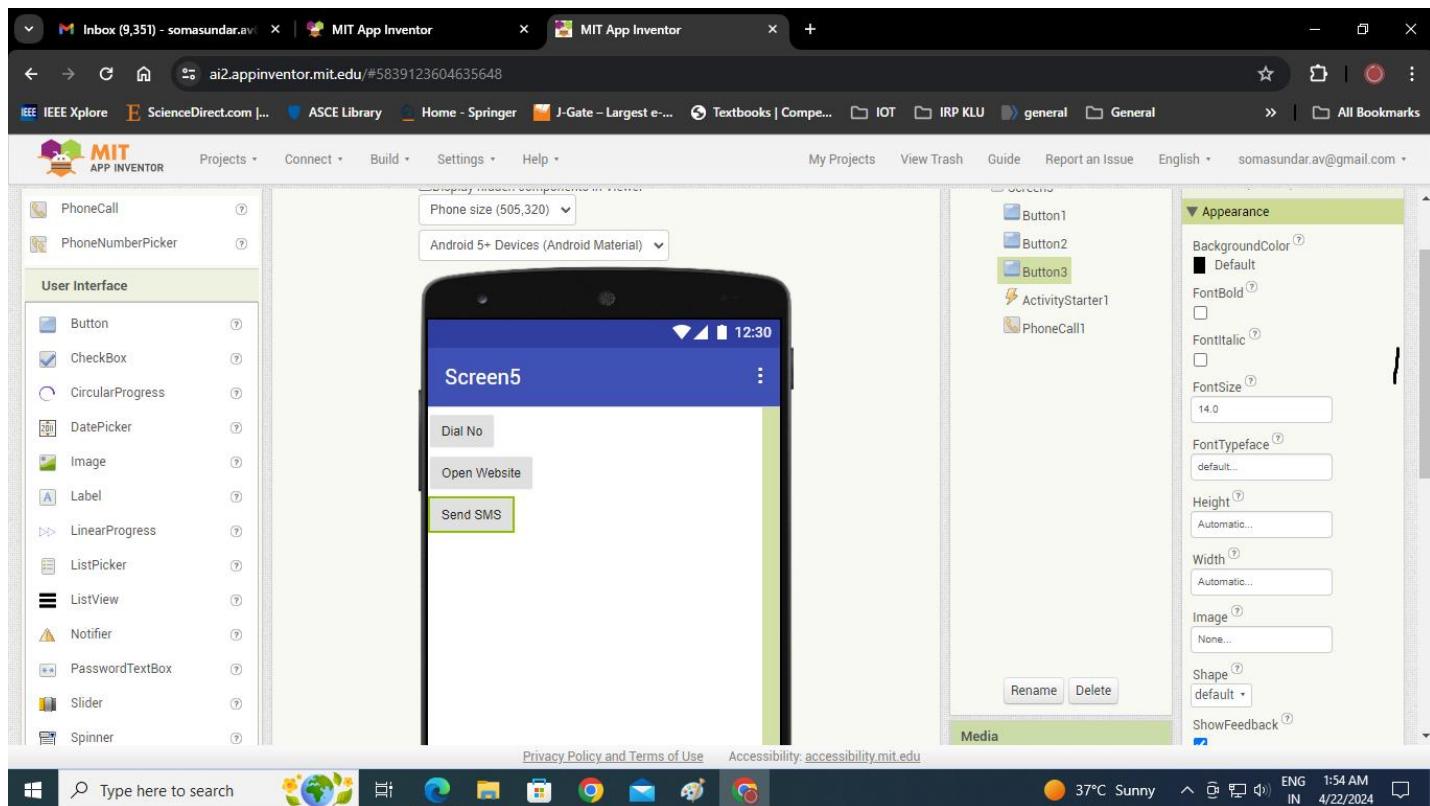
Output:



Experiment 5

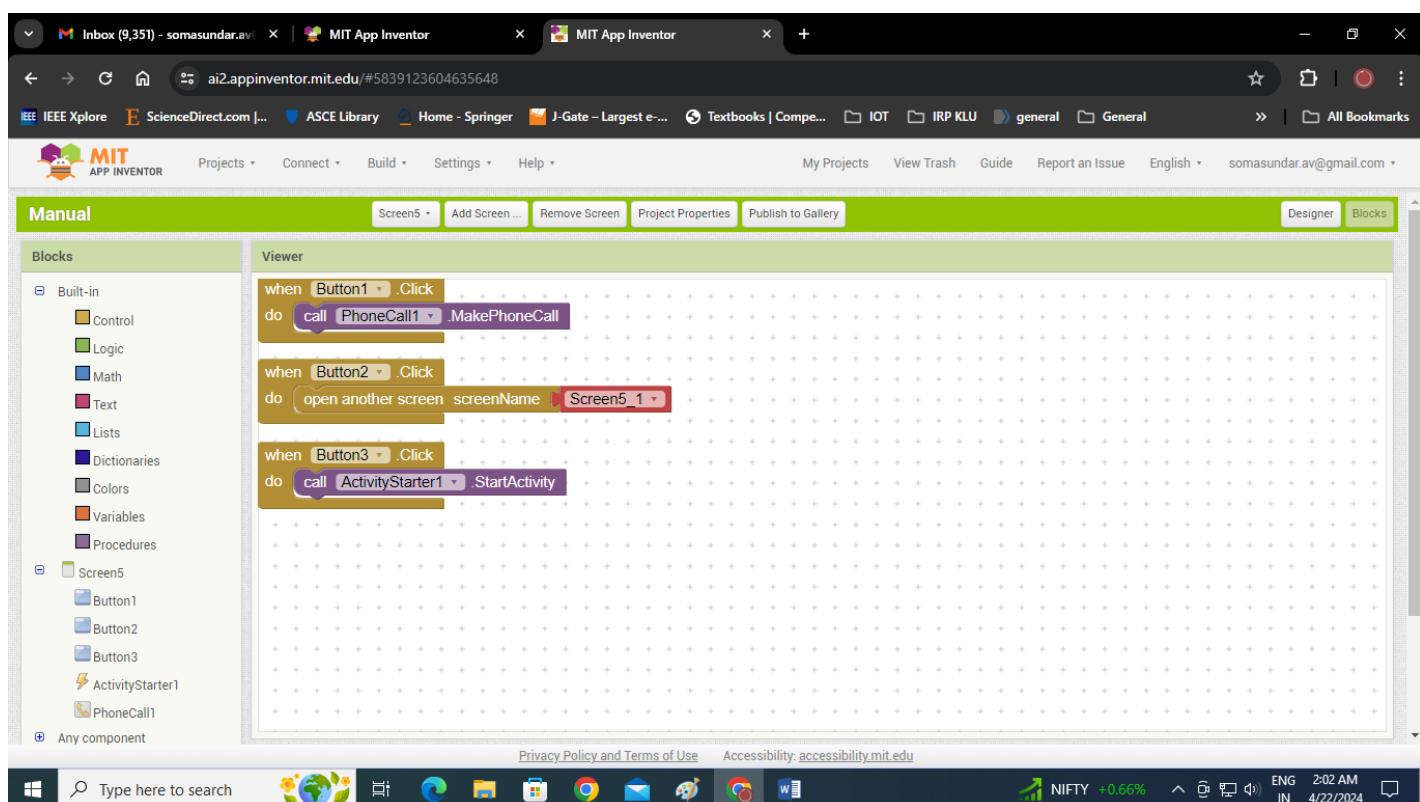
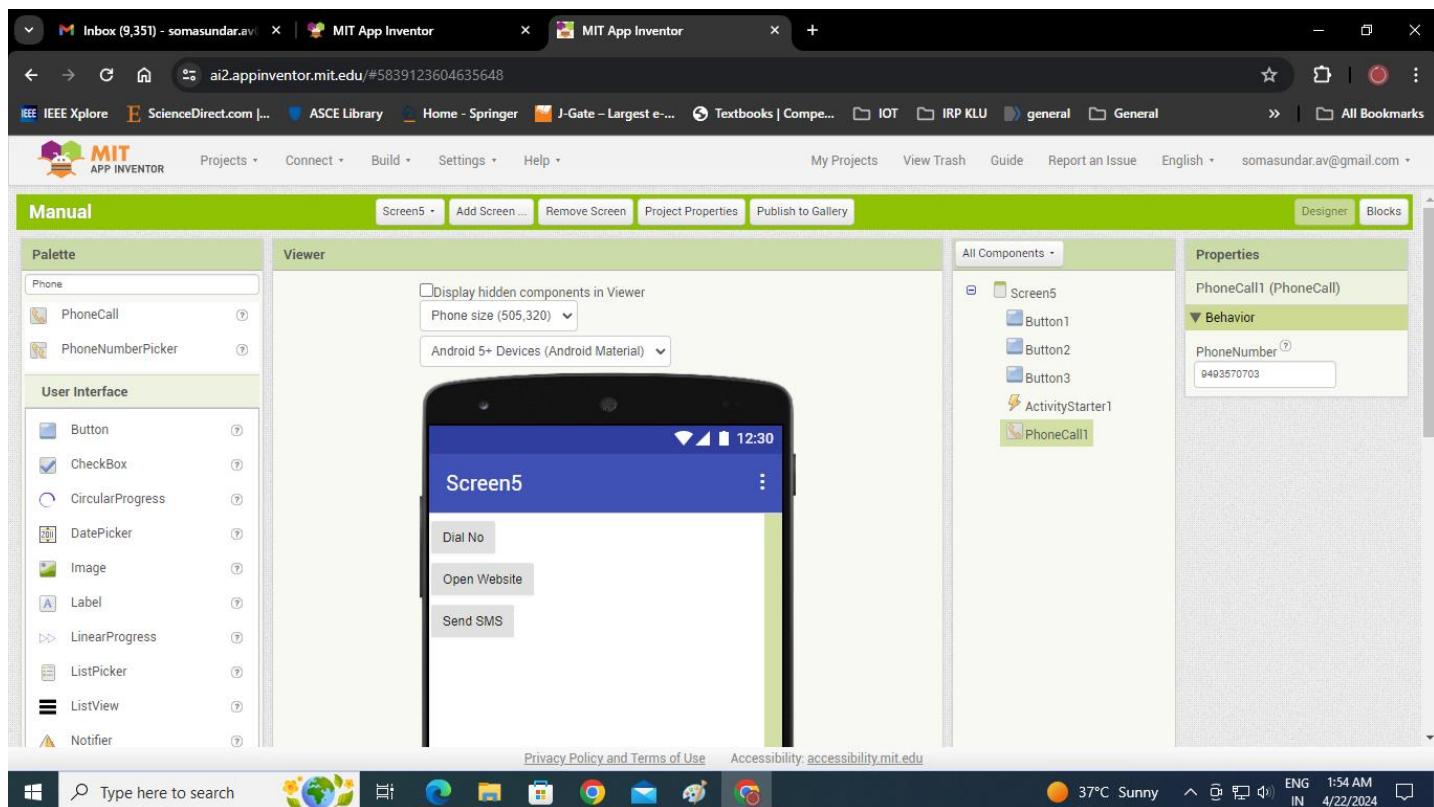
AIM: Develop an application that uses a menu with 3 options for dialing a number, opening a website and to send an SMS. On selecting an option, the appropriate action should be invoked using intents.

Program:

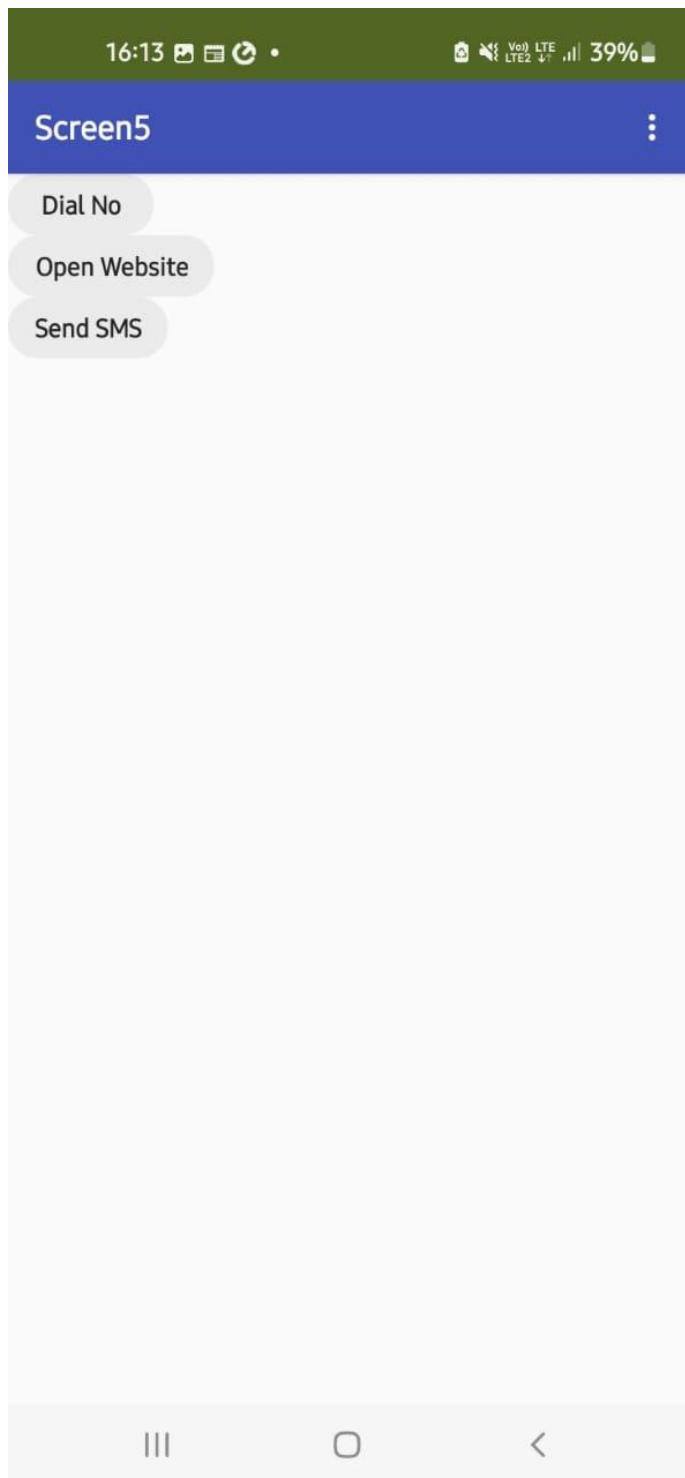


Activity Starter Behavior Properties:

Action android.intent.action.SENDTO
DataUri smssto:9493570703
ExtraKey sms_body
ExtraValue Hello, this is a test SMS!123



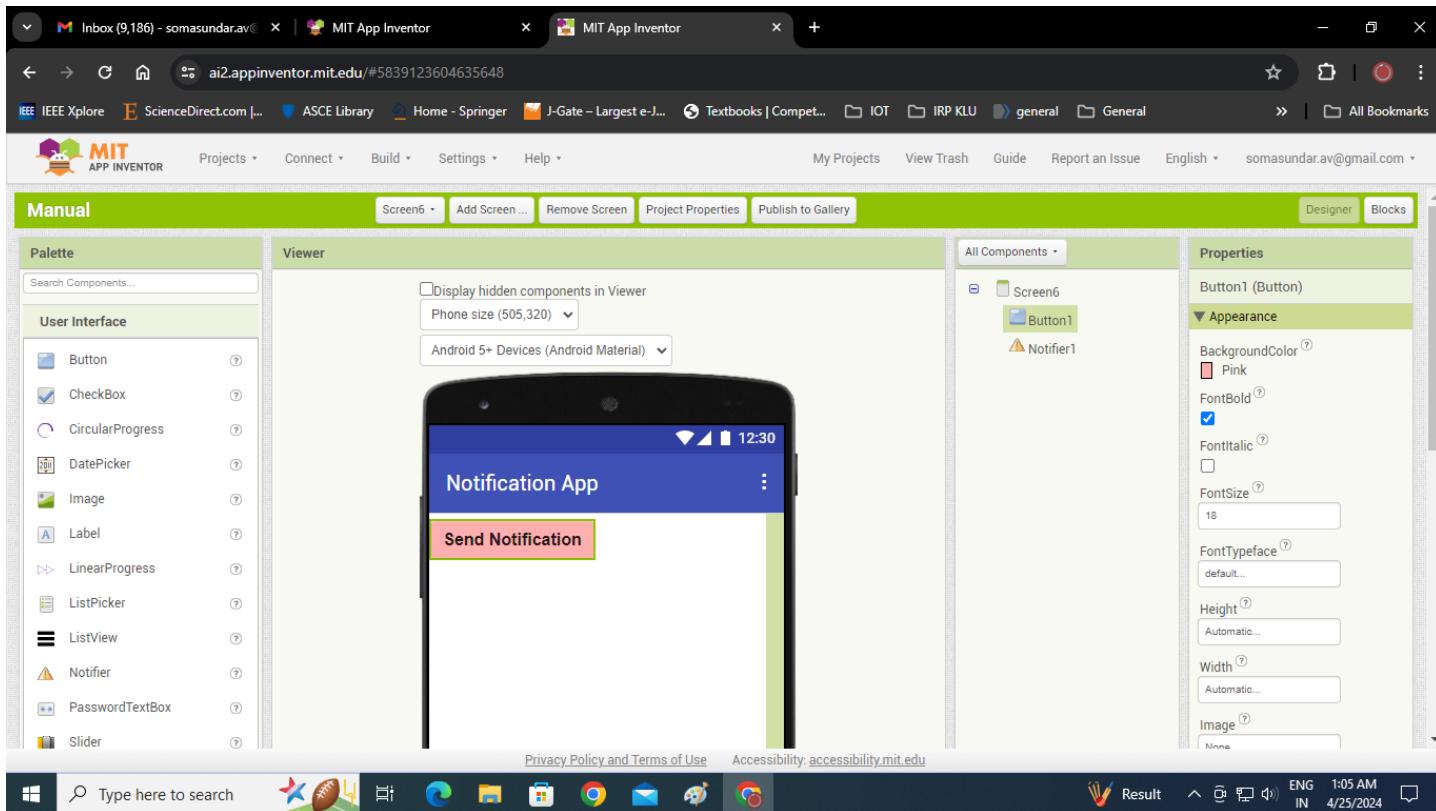
Output:



Experiment 6

AIM: Develop an application that inserts some notifications into Notification area and whenever a notification is inserted, it should show a toast with details of the notification.

Program:

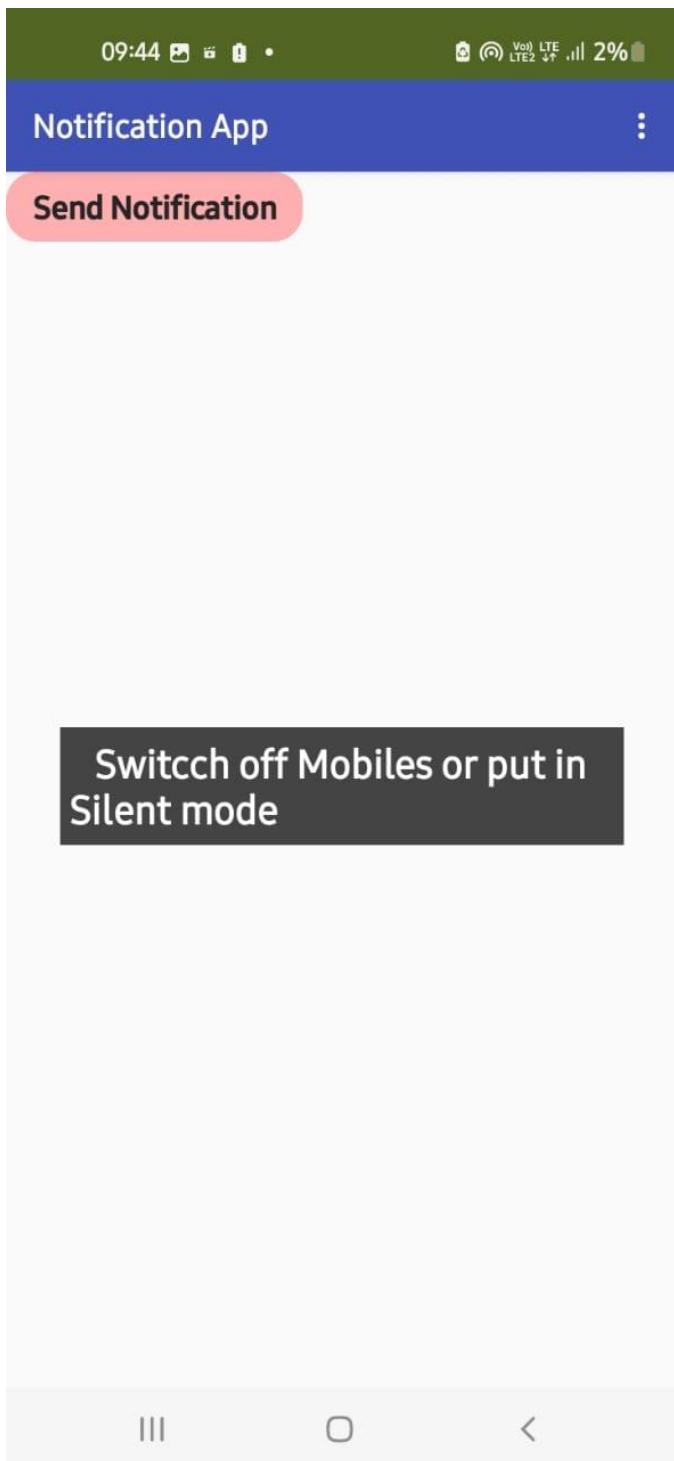


The screenshot shows the MIT App Inventor Designer interface. On the left, the 'Blocks' palette is open, displaying categories like Built-in (Control, Logic, Math, Text, Lists, Dictionaries, Colors, Variables, Procedures), Screen6, Button1, Notifier1, and Any component. In the center, the 'Viewer' window displays a script for 'Button1 .Click':

```
when Button1 .Click
do call Notifier1 .ShowAlert
    notice "Switch off Mobiles or put in Silent mode"
```

Below the script, there are two warning icons (triangle and circle) and a 'Show Warnings' button. On the right side of the screen, there is a small icon of a backpack and some control buttons (+, -, trash). At the bottom, the Windows taskbar shows various pinned apps and the system clock.

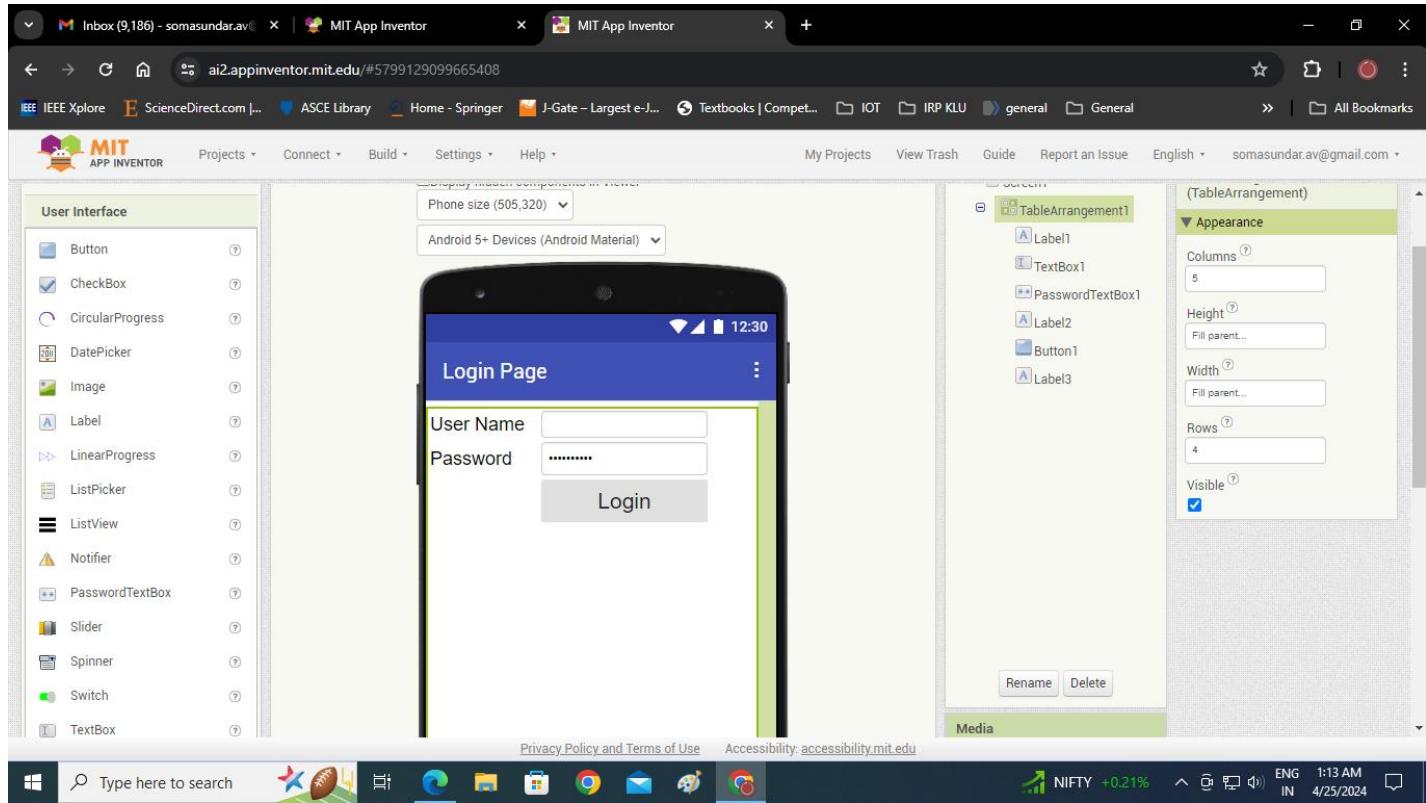
Output:

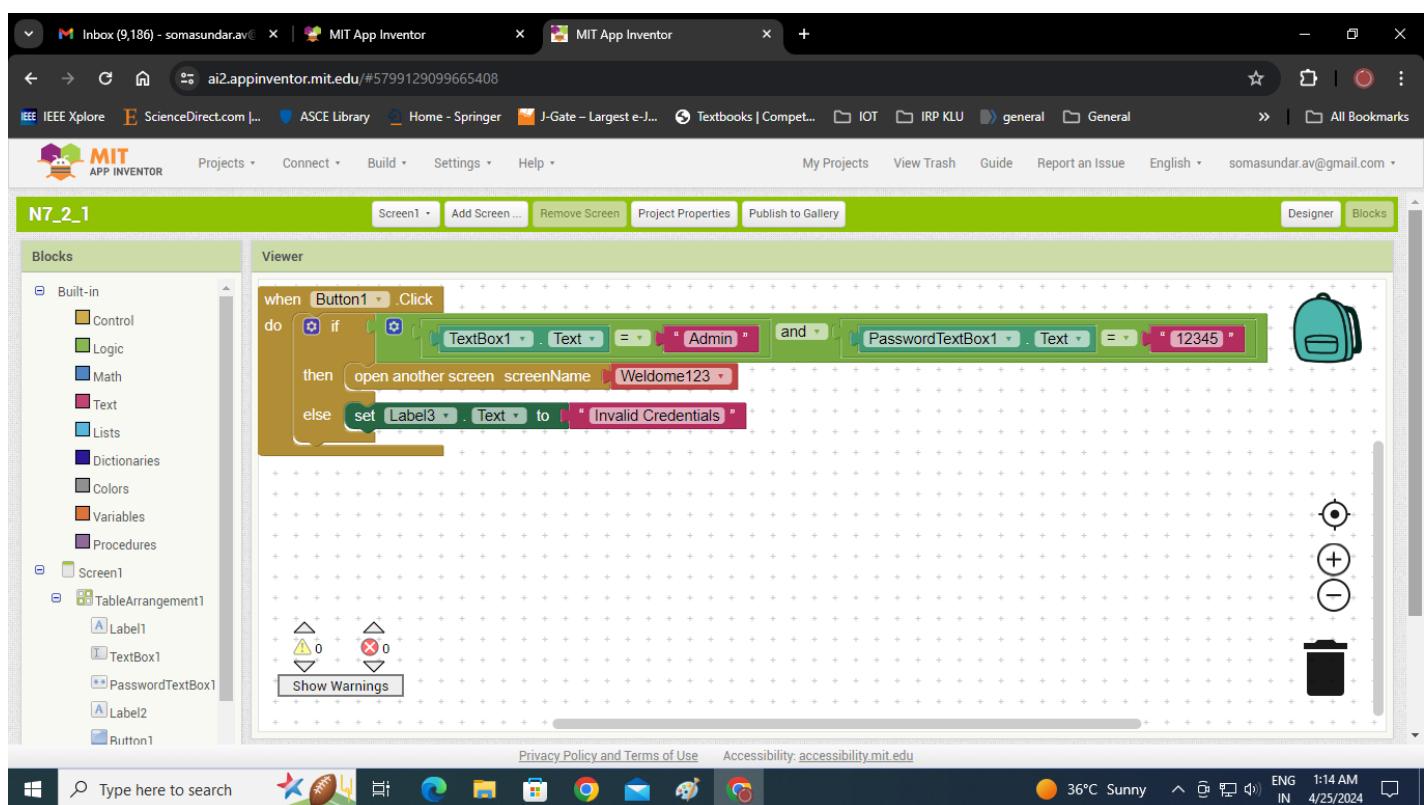
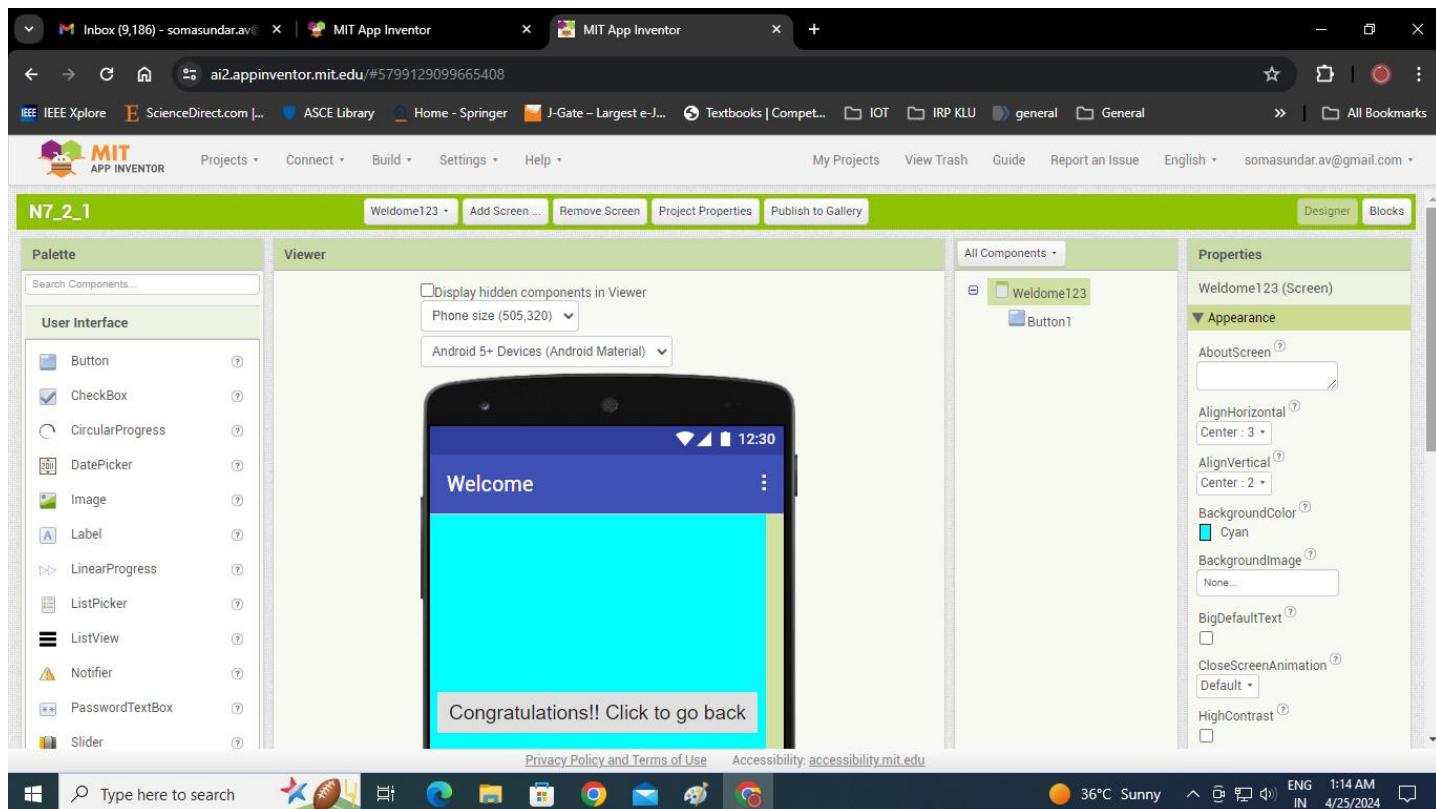


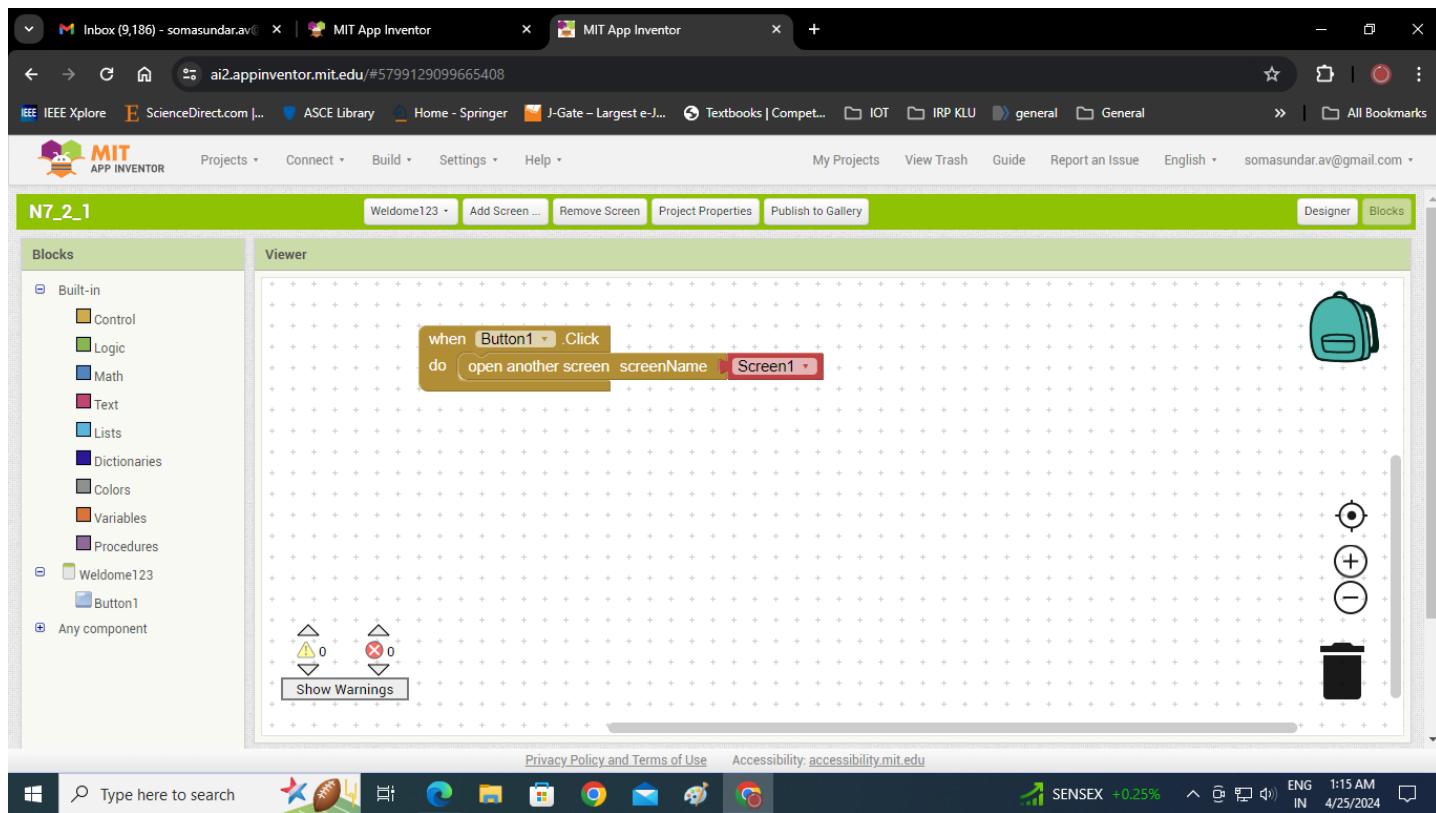
Experiment: 7

AIM: Create an application that uses a text file to store user names and passwords (tab separated fields and one record per line). When the user submits a login name and password through a screen, the details should be verified with the text file data and if they match, show a dialog saying that login is successful. Otherwise, show the dialog with Login Failed message

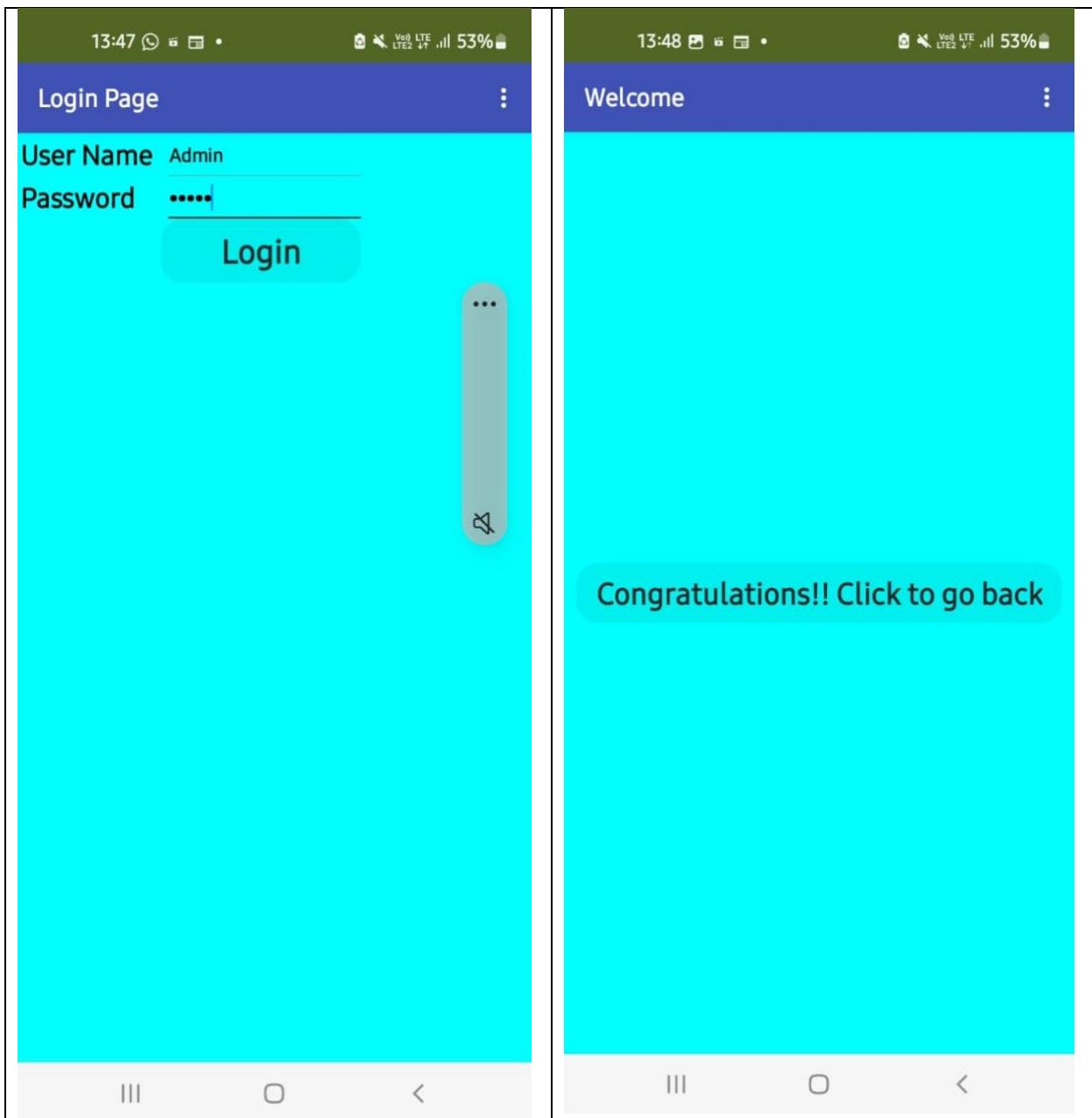
Program:







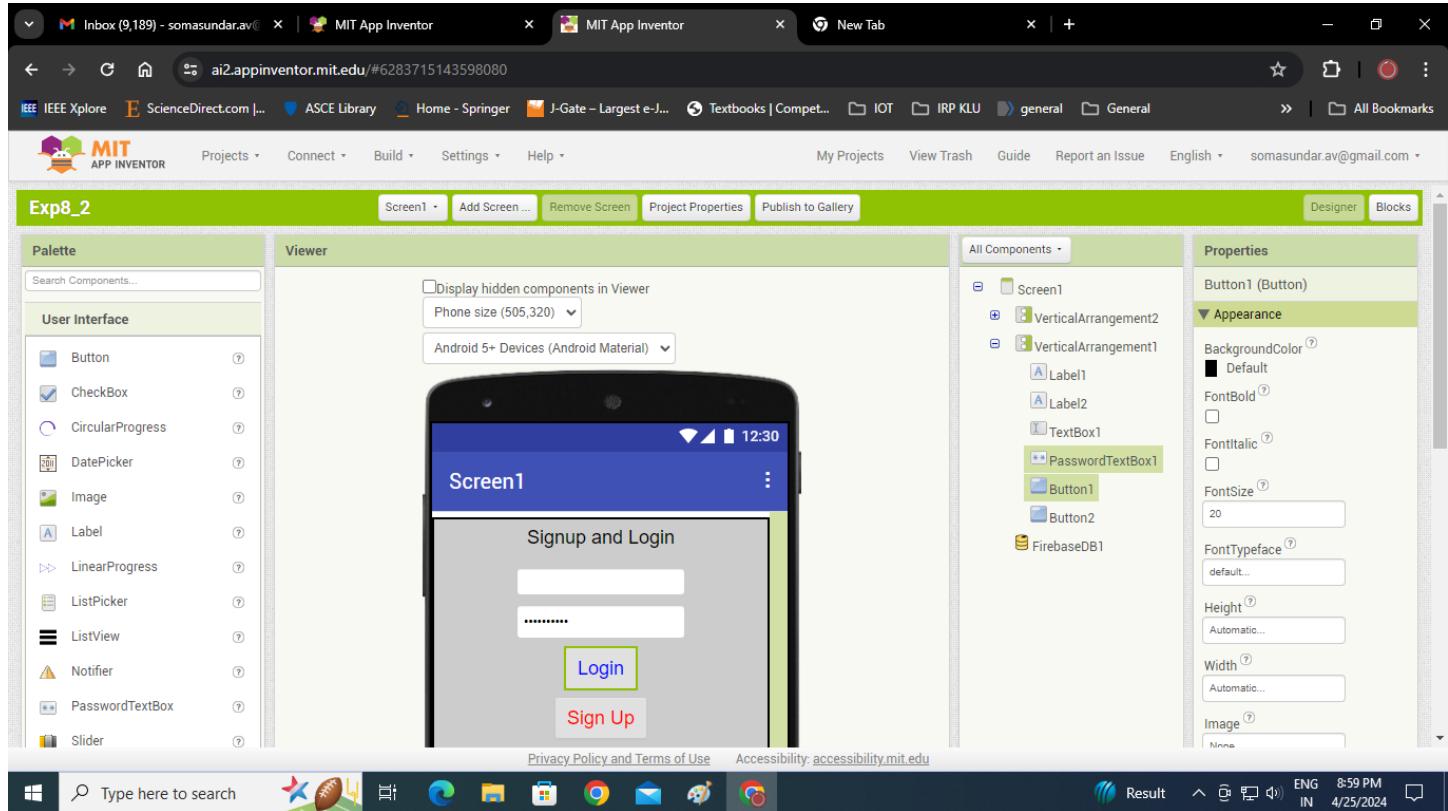
Output:

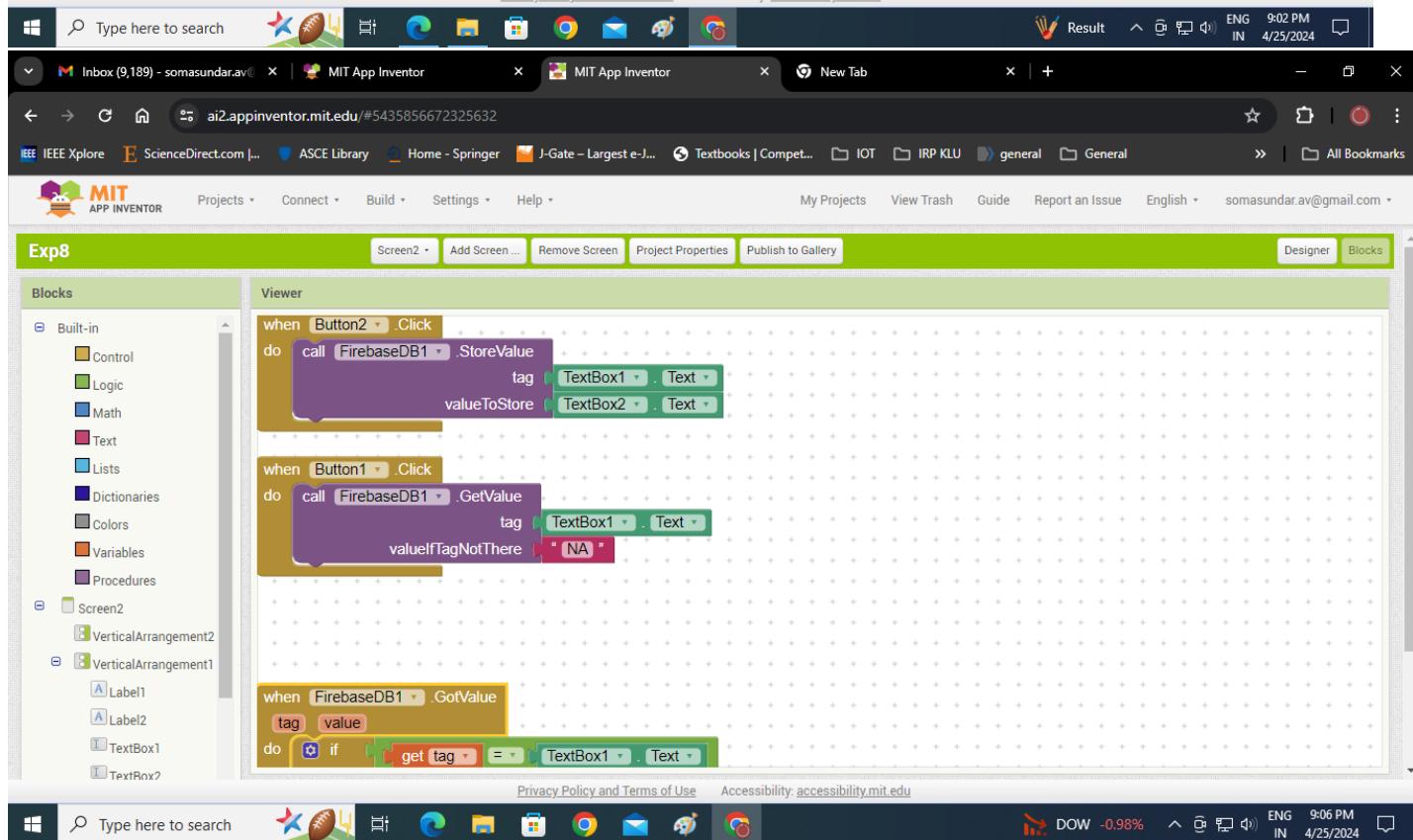
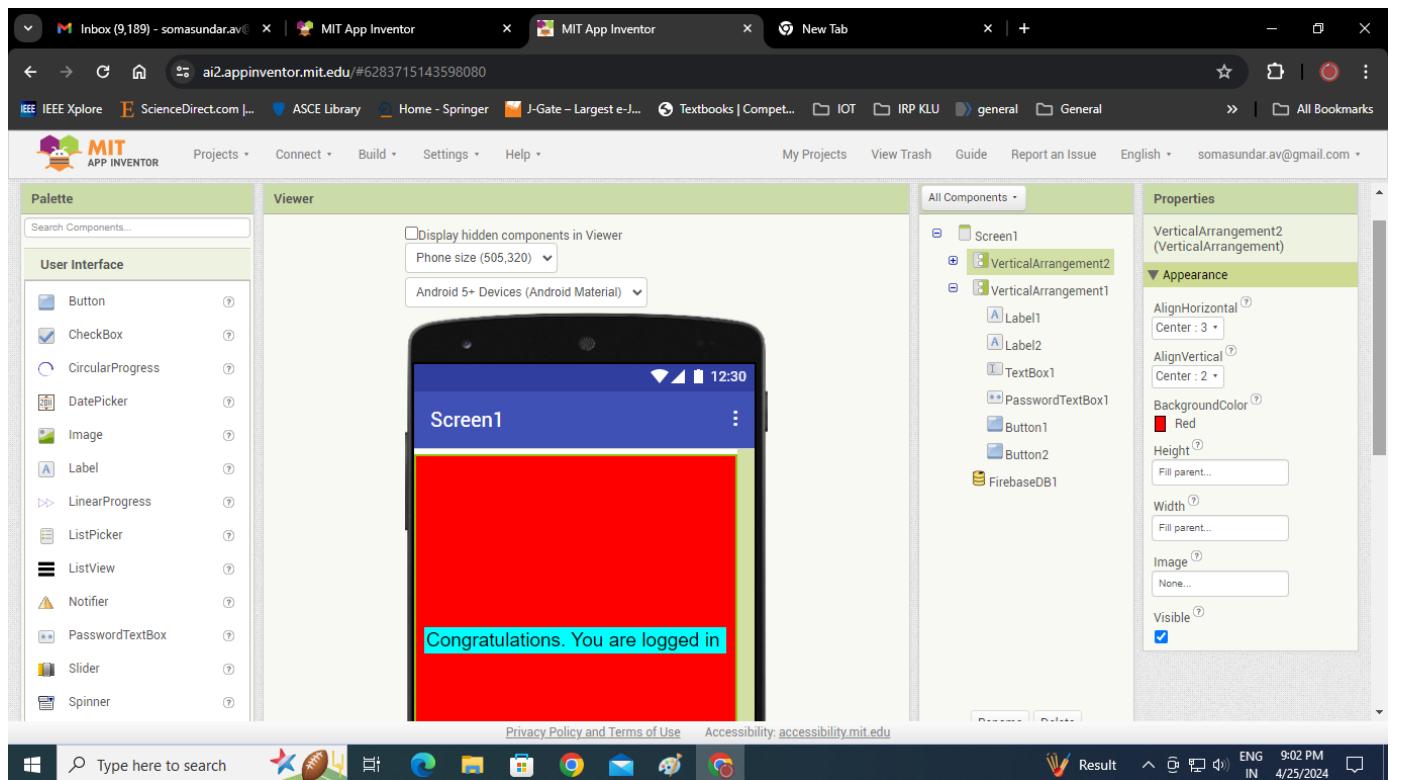


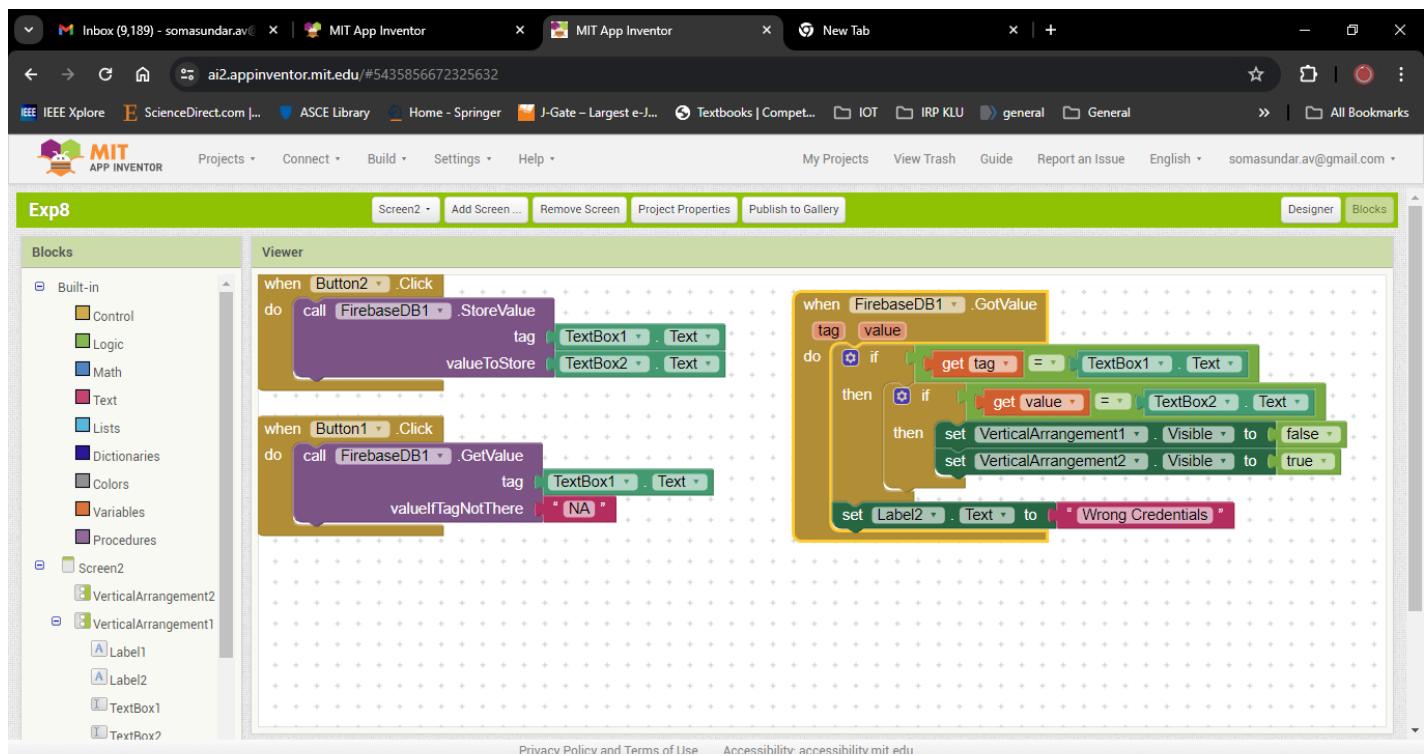
Experiment: 8

AIM: Create a database and a user table where the details of login names and passwords are stored. Insert some names and passwords initially. Now the login details entered by the user should be verified with the database and an appropriate dialog should be shown to the user.

Program:







The screenshot shows the Firebase Realtime Database Rules playground. The rules are defined as follows:

```
{
  "rules": {
    ".read": "now < 1713423600000", // 2024-4-18
    ".write": "now > 1713423600000", // 2024-4-18
  }
}
```

The screenshot displays two windows side-by-side, illustrating the integration of Firebase Realtime Database with MIT App Inventor.

Top Window (Browser - Firebase Realtime Database):

- URL:** console.firebaseio.google.com/project/aad2db/database/aad2db-default-rtbd/data
- Project Overview:** AAD2DB
- Realtime Database:** Data tab selected. Shows a tree structure of data under the root:
 - Exp8
 - abc: "123"
 - abcd: "12345"
 - Exp8_2
 - anu: "123"
 - shalini: "1234"
- Bottom Status Bar:** NIFTY +0.22%, ENG IN 9:11 PM 4/25/2024

Bottom Window (MIT App Inventor):

- Title:** AI2.appinventor.mit.edu/#6283715143598080
- Toolbar:** Projects, Connect, Build, Settings, Help, My Projects, View Trash, Guide, Report an Issue, English, somasundar.av@gmail.com
- Designer View:**
 - Palette:** User Interface components: Button, CheckBox, CircularProgress, DatePicker, Image, Label, LinearProgress, ListPicker, ListView, Notifier, PasswordTextBox, Slider.
 - Viewer:** Displays a smartphone screen titled "Screen1" showing a "Signup and Login" form with two text boxes, a "Login" button, and a "Sign Up" button.
 - All Components:** A tree view of components:
 - Screen1
 - VerticalArrangement2
 - VerticalArrangement1
 - Label1
 - Label2
 - TextBox1
 - PasswordTextBox1
 - Button1
 - Button2
 - FirebaseDB1
 - Properties Panel:** Properties for the FirebaseDB1 component:
 - Behavior:**
 - FirebaseToken: eyJ0eXAiOiJKV1QiLCJhbG...
 - FirebaseURL: https://aad2db-default-rtbd...
 - Use Default
 - Persist
 - ProjectBucket:** Exp8_2
- Bottom Status Bar:** 29°C Sunny, ENG IN 9:13 PM 4/25/2024

Copy url from Firebase Real time database into MIT App Inventor, FirebaseDB1 component's Behavior property -FirebaseURL

Output:

