PROJECT ARTEFACT DECLARATION

SEM 1 2023/2024 I BITP 3453 MOBILE APPLICATION DEVELOPMENT

Group No:	2
Matric No	Name
B032110301	ANG WEI KANG
B032110251	LUM FU YUAN
B032110376	SIM WENG JIN

FAKULTI TEKNOLOGI MAKLUMAT DAN KOMUNIKASI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DESCRIPTION OF THE PROJECT

JobForge is a comprehensive application designed to streamline the job search process for users. Leveraging a chatbot assistant powered by Natural Language Processing (NLP), JobForge offers an intuitive interface for users to interact and seek guidance during their job search journey. The chatbot provides personalized assistance, making the job exploration experience more dynamic and user-friendly.

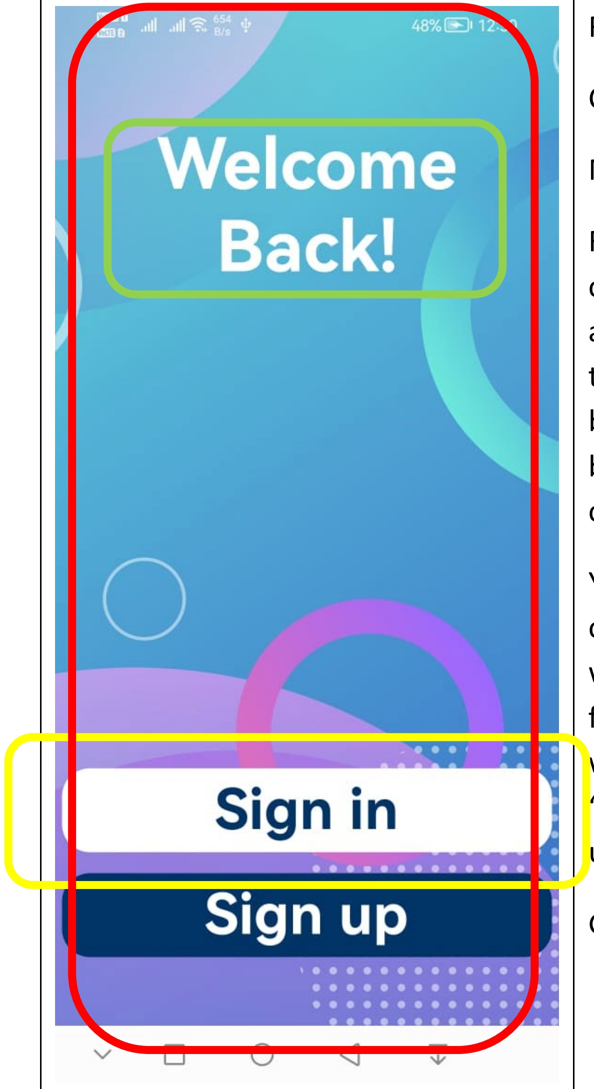
The app allows users to efficiently filter job opportunities based on their preferences and requirements. Through a sophisticated filtering system, users can narrow down job listings to find positions that align with their skills, experience, and career objectives. JobForge simplifies the application process by enabling users to apply for jobs directly through the platform, connecting job seekers with potential employers seamlessly.

Moreover, JobForge empowers users to update their personal information, including the ability to upload and modify their resumes. This ensures that users can present the most accurate and compelling professional profiles to prospective employers, enhancing their chances of securing relevant job opportunities. With its user-centric features and innovative chatbot assistant, JobForge is poised to revolutionize the job search experience, offering a holistic solution for individuals navigating the competitive employment landscape.

YouTube demo link:

https://youtu.be/mh9sOV2HwLc?si=vhomnzUDg8fbwB9q

THE APPLICATION FIRST SCREEN



File name: customBackground.dart, first_screen.dart, custom_button.dart

Class: firstPage, customBackground ,ButtonWidget

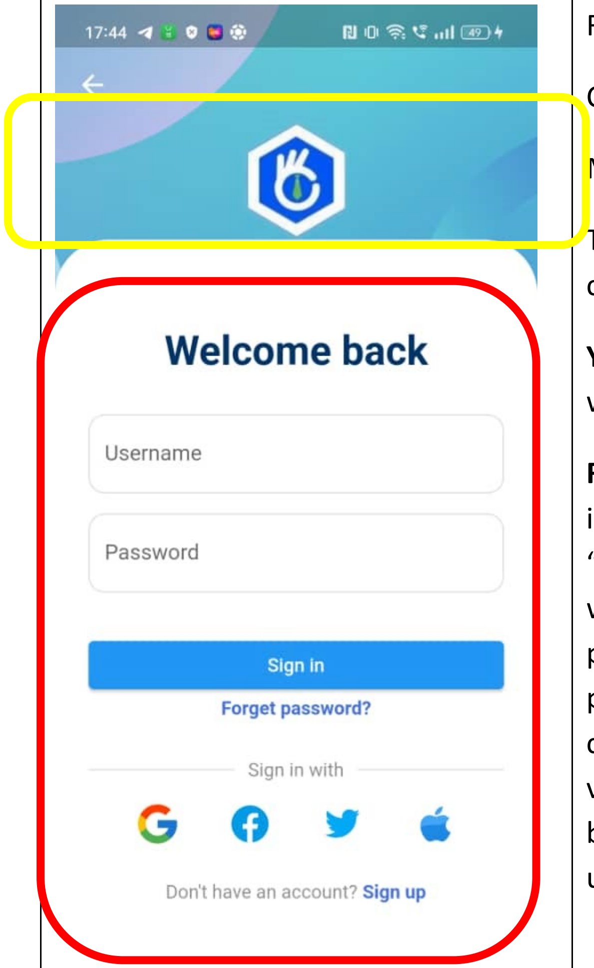
Method:

Red colour box is background image where it defined as customBackground class and located in customBackground.dart file. This class is superclass using inheritance concept/method as this class also need to be used by another class. This file return a “Scaffold” and using “Stack” as its body, inside the body, using “image.asset” to display the image and the following is the subclass widget. The benefit using inheritance method is ease designer, for example, when we want to modify the background image, we just go to customBackground.dart file to modify and the screen that use this class will change simultaneously.

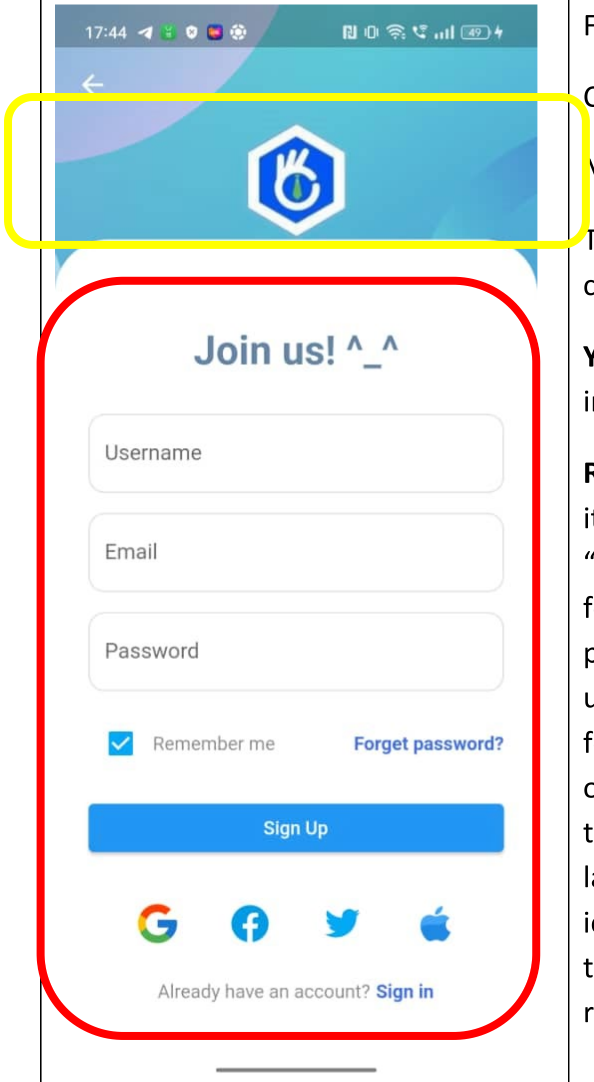
Yellow colour box is “GestureDetector “where it defined as ButtonWidget class and located in custom_button.dart . This class also is a superclass , thus the “Sign in” and “Sign up” is same class but with different parameter. ButtonWidget class return a “GestureDetector”. Inside “GestureDetector”, for ontap , colour, text, text colour have defined with a variable and make requirement for them. So when calling to this class, there are four requirement need to be filled. For “Sign in” “GestureDetector” will go to the Sign in screen while the “Sign up” “GestureDetector” will go to Sign up screen.

Green colour box is using “Richtext” to display the text where it located in first_screen.dart.

THE OTHER SCREEN 1

 Welcome back Username Password Sign in Forget password? Sign in with Don't have an account? Sign up	File name: customBackground.dart, signin_screen.dart Class: customBackground , SignInScreen Method: The background image already explained at previous page, Thus in this “Sign in” page, customBackground class is used as superclass **Yellow** colour box is “Expanded” widget inside Column widget, using “Container” to display the image with asset image. **Red** colour box also is “Expanded” widget inside Column widget before the yellow colour box. Inside it, using “SingleChildScrollView to make the form scrollable. Inside “SingleChildScrollView”, create a “Column” widget in order to vertically arranges various UI elements. The element The screen starts with a welcoming text, followed by two “TextFormField” widgets for entering the username and password. These text fields include validation logic to ensure non-empty input. A clickable "Forget password?" text are present. The "Sign in" button, when pressed, currently has a navigation action commented out, which would direct users to the “main_controller” screen upon successful validation.Horizontal dividers separate the sign-in form from alternative sign-in options represented by brand icons like Google, Facebook, Twitter, and Apple. A clickable link below suggests users sign up for an account and tapping it would navigate them to the “sign-up” screen.

THE OTHER SCREEN 2



File name: customBackground.dart, signup_screen.dart

Class: customBackground , SignUpScreen

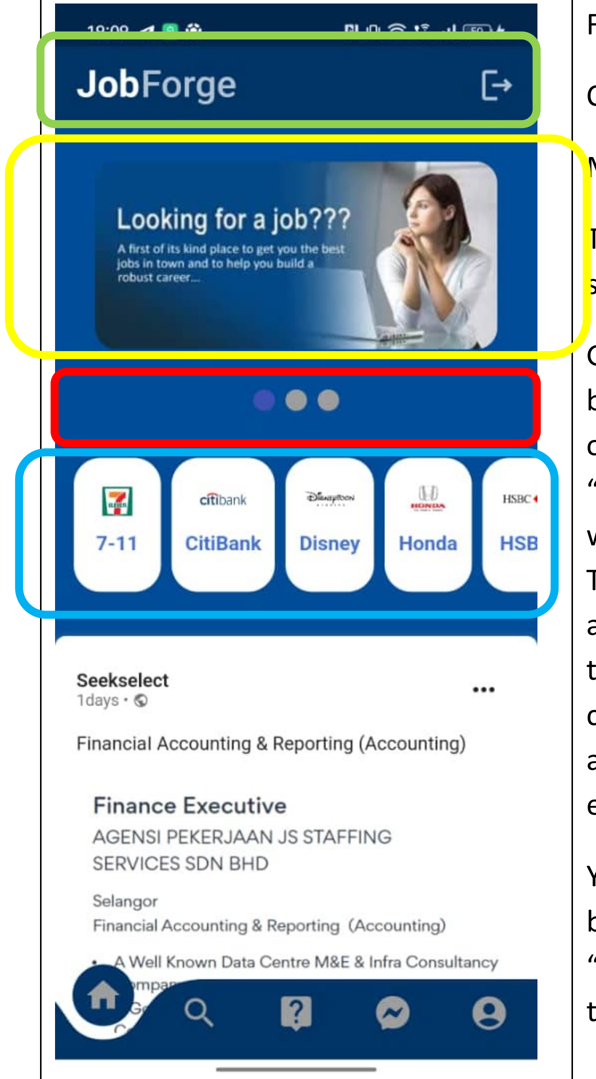
Method:

The background image already explained at first page, Thus in this "Sign up" page, customBackground class is used as superclass

Yellow colour box is "Expanded" widget inside Column widget, using "Container" to display the image with asset image.

Red colour box also is "Expanded" widget inside Column widget before the yellow colour box. Inside it, using "SingleChildScrollView to make the form scrollable. Inside "SingleChildScrollView", create a "Column" widget in order to vertically arranges various UI elements. The element starts with the form, encapsulated by a "Form" widget, holds various input fields for username, email, and password, each equipped with validation logic to ensure the required information is provided. The user interface includes a welcoming message, input fields with corresponding labels, and checkboxes for remembering the password. Additionally, a clickable "Forget password?" text provides a recovery option. The sign-in button triggers a validation check for the form's completeness and the agreement to data processing. Feedback is displayed through "SnackBar" notifications. Horizontal dividers labeled "Sign up with" separate the form from alternative sign-up options represented by brand icons such as Google, Facebook, Twitter, and Apple. A link at the bottom encourages users to switch to the sign-in screen by tapping the "Sign in" text, facilitating seamless navigation between registration and login processes.

THE OTHER SCREEN 3


JobForge
Looking for a job???
A first of its kind place to get you the best jobs in town and to help you build a robust career...
7-11
CitiBank
Disney
Honda
HSB

Seekselect
1days •

Financial Accounting & Reporting (Accounting)

Finance Executive
AGENSI PEKERJAAN JS STAFFING SERVICES SDN BHD

Selangor
Financial Accounting & Reporting (Accounting)

A Well Known Data Centre M&E & Infra Consultancy

File name: home.dart, middle_widget.dart, top_widget.dart,upper_widget.dart

Class: home_page, Upper, topper, PostContainer, Middle, news

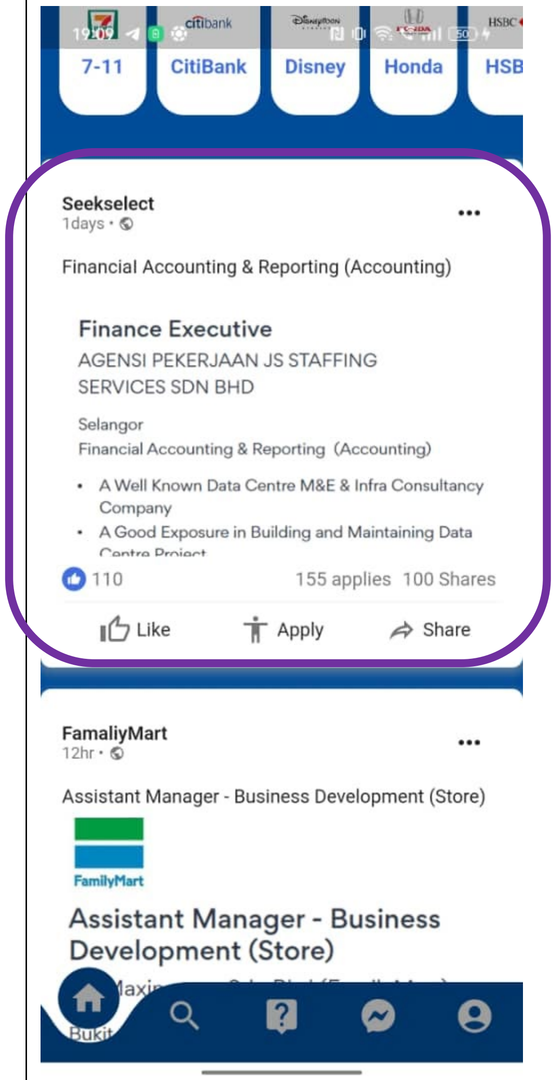
Method:

The whole page is using "CustomScrollView" widget which can create a dynamic scrollable view with slivers.

Green colour box is a "SliverAppBar" widget serving as the app bar for a navigation screen. The app bar features a gradient background specified by the "Color.fromRGBO" function, combining shades of blue. The title of the app bar is constructed as a `Row` containing two "Text" widgets. The first "Text" widget displays "Job" with a bold and larger font, coloured in white, while the second "Text" widget presents "Forge" with a slightly smaller font and a white70 colour, providing a subtle contrast. The "centerTitle" property is set to "false", allowing the title to align to the start of the app bar. The app bar is designed to be floating, which means it remains visible even when the user scrolls through the content below. On the right side of the app bar, there is an action represented by a "Container" containing an "IconButton". The button displays a logout icon. Upon pressing the logout button, the app navigates to the "firstPage" (First screen) using the `Navigator.pushAndRemoveUntil` method, ensuring that all previous routes are cleared.

Yellow colour box is "PageView" widget that configured with a horizontal scroll direction, managed by a specified controller ("_controller"). The individual pages of the "PageView" consist of custom "news" widgets, each featuring an "Image" widget displaying different news images retrieved from the asset directory. The "fit: BoxFit.cover" property ensures the images cover the allotted space.

	Red colour box is “SmoothPageIndicator” that provide a visual representation of the current page within the scrollable view. The indicator is synchronized with the “PageView” through the shared controller (“_controller”) and employs the “SwapEffect()” for a visual effect. To enhance the layout, a “SizedBox” with a height of 15 is introduced, contributing vertical spacing between the page indicator and the other components. Blue colour box isa “ListView.builder” that utilized to dynamically build the list of icons in a horizontal scrollable fashion. The builder function creates instances of the “iconBox” widget for each item in the list. The “iconBox” class, which extends “StatelessWidget”, represents each icon element within the scrollable list. It takes an index parameter to determine the position of the icon within the list. Inside the “iconBox”, a “Stack” widget is used to overlay a gesture detector on top of a container representing the individual icon. The gesture detector enables navigation to a “company_post” screen when the icon is tapped. The icon itself is represented by an image loaded from the asset directory (\`piclist[index].path\`). The image is contained within a styled container with a white background, rounded corners (via \`BorderRadius.circular(20)\`), and additional padding. The container also includes text displaying the name of the icon (\`piclist[index].name\`). The overall structure of the \`iconBox\` class ensures a visually appealing representation of each icon within the horizontal scrollable list.



7-11 CitiBank Disney Honda HSB

Seekselect
1days •

Financial Accounting & Reporting (Accounting)

Finance Executive
AGENSI PEKERJAAN JS STAFFING SERVICES SDN BHD

Selangor
Financial Accounting & Reporting (Accounting)

- A Well Known Data Centre M&E & Infra Consultancy Company
- A Good Exposure in Building and Maintaining Data Centre Project

110 155 applies 100 Shares

Like Apply Share

FamaliyMart
12hr •

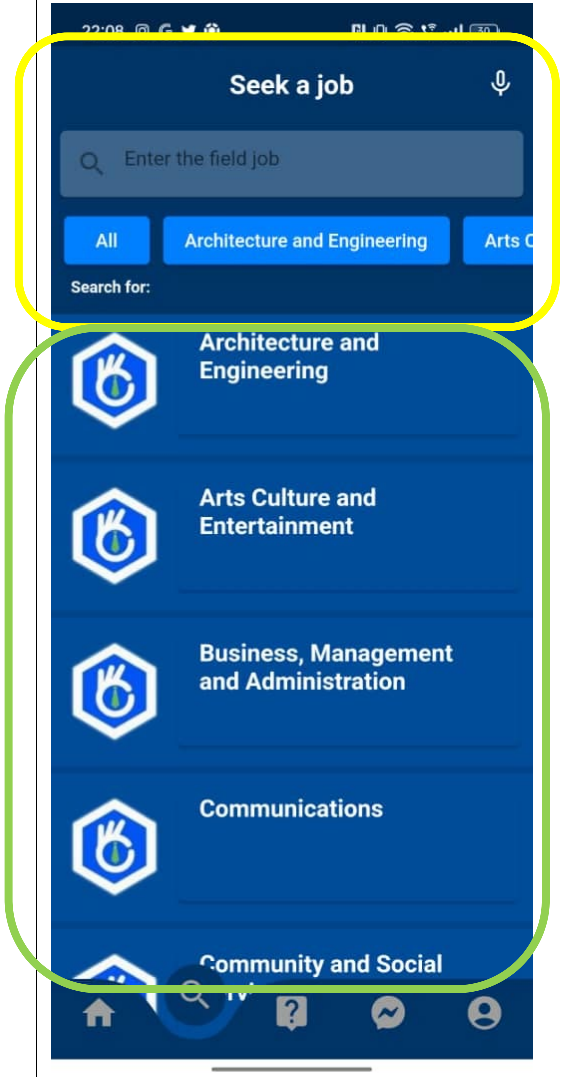
Assistant Manager - Business Development (Store)

FamilyMart

Assistant Manager - Business Development (Store)

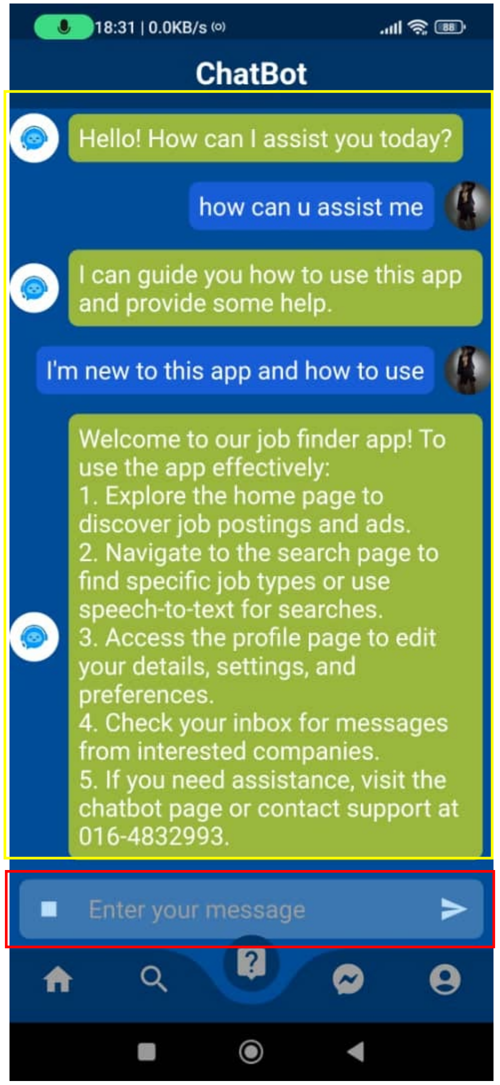
Purple colour box is a “PostContainer” widget serves as the main container for individual posts, incorporating the post's header, content, and statistics. The header, managed by the `_PostHeader` widget, includes the user's profile picture, username, and the time elapsed since the post's creation.The core content of the post, including the caption and an optional image, is organized within the `Column` widget in the `PostContainer`. The presence of an image is conditionally rendered based on the existence of a path in the post data. The `_PostStats` widget is responsible for displaying the post statistics, such as the number of likes, comments, and shares. Additionally, it features interactive buttons for liking, commenting, and sharing, each represented by the `_PostButton` widget.The `_PostButton` widget is a reusable component representing an interactive button within the post statistics section. It consists of an icon and an optional label, responding to user taps with specified `onTap` functions. The `ProfileAvatar` widget is employed within the `_PostHeader` to display the user's profile picture.

THE OTHER SCREEN 4

 22:08 Seek a job Enter the field job All Architecture and Engineering Arts C Search for: Architecture and Engineering Arts Culture and Entertainment Business, Management and Administration Communications Community and Social	File name: lowerWidget.dart, searchWidget.dart, speechToText.dart Class: lower_widget_search, search, speech_home Yellow colour box is an app bar with a title ("Seek a job"), a search input field, a microphone icon for speech-to-text functionality, and a bottom section for filtering job categories. The app bar is styled with a blue background color, and the speech-to-text functionality is implemented using the `speech_Home` page. Within the app bar, a `TextFormField` is utilized for user input, and a controller is assigned to it for managing the text input. The search input field is designed with a hint text, a search icon, and a close icon to clear the input. When users submit the search field, the `onFieldSubmitted` function is triggered, updating the `text` variable in the widget's state.The filtering options for job categories are organized horizontally beneath the search input. These options include an "All" button and various individual buttons for specific job categories. These buttons are implemented using the `_allItems` and `_singleItem` functions, which return `ElevatedButton` widgets styled with a blue background color. The bottom section of the app bar displays the current search text dynamically. It provides feedback to users, showing the active search query. The main body of the page contains the `lower_widget_search` widget, which is intended to display the search results or additional content related to the search. Green colour box is `lower_widget_search` Flutter widget is designed to serve as the lower section of a search page, presenting search results in a visually organized manner. The widget employs a `ListView.builder` to dynamically generate a scrollable list of search outcomes based on user input.

	Initially, it filters a global `ITEM_LIST` to create a `searchList` containing items whose names match the entered search text. Each entry in the `searchList` is then displayed as a `Column` within the list, ensuring a clear and structured presentation.Each item's layout comprises two main components: an Image Container and an Info Container. The Image Container is a fixed-size container exhibiting the item's image, dynamically sourced from the item's path in the `searchList`. The dimensions are set to 75x75, and the image is configured to fit within these dimensions. The Info Container, an expanded container, contains an `ElevatedButton` serving as an interactive area for each search result. The button is styled with a blue background color (`Color.fromRGBO(0, 76, 153,1)`), and inside it, a `Column` holds text elements such as the item's name (`searchList[index].name`), styled with bold font, a font size of 18, and a white color.The `ListView.builder` iterates through the `searchList`, creating a well-organized layout for each search result. To visually separate each item, a `Divider` widget is added.

THE OTHER SCREEN 5



Hello! How can I assist you today?

how can u assist me

I can guide you how to use this app and provide some help.

I'm new to this app and how to use

Welcome to our job finder app! To use the app effectively:
1. Explore the home page to discover job postings and ads.
2. Navigate to the search page to find specific job types or use speech-to-text for searches.
3. Access the profile page to edit your details, settings, and preferences.
4. Check your inbox for messages from interested companies.
5. If you need assistance, visit the chatbot page or contact support at 016-4832993.

File Name: ChatBotPage.dart, ChatBotCRTL.dart, API.dart

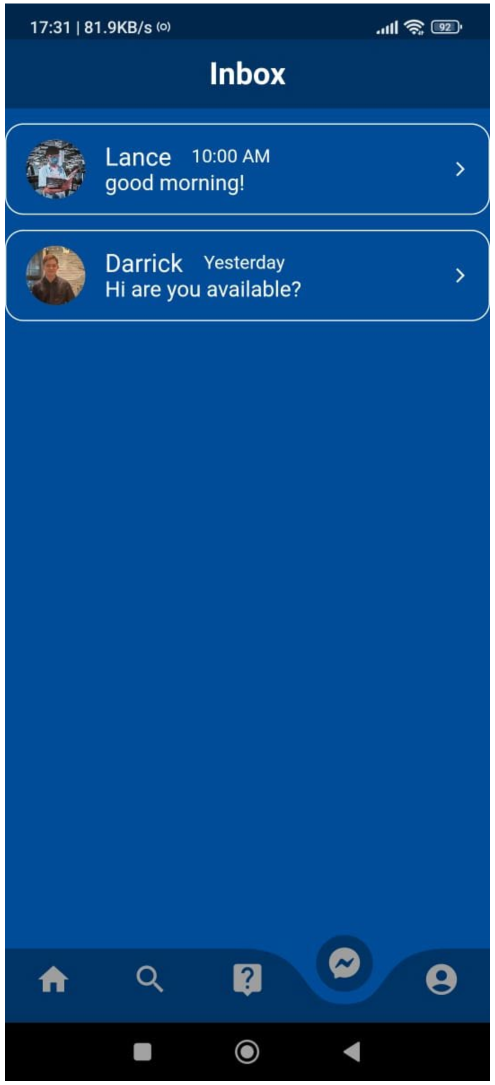
Class: ChatBotPage, ChatBotCTRL, _ChatBotCTRL

Method:

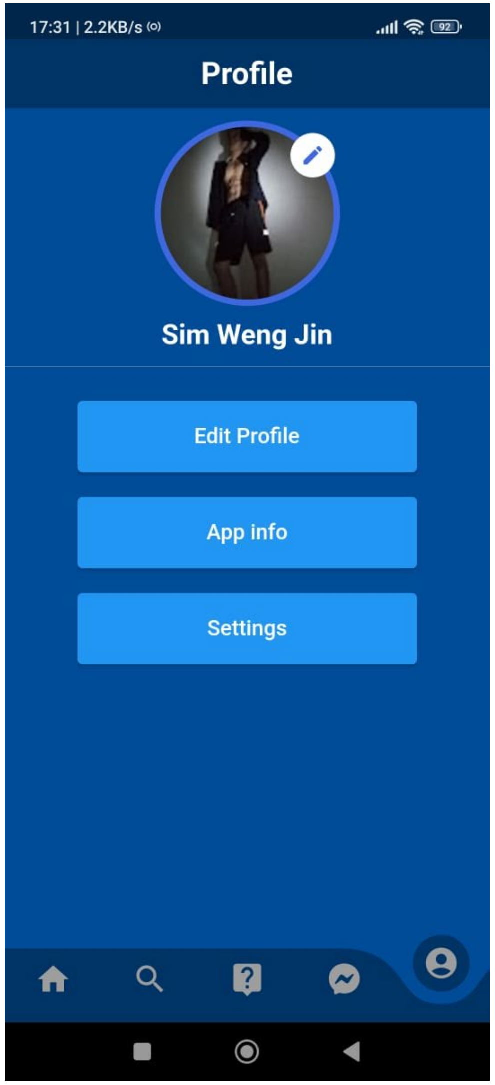
Yellow color box show the _buildMessages method creates a list of Row widgets to display chat messages, with sender-dependent alignment and avatars. Messages are styled within rounded containers, and the method is part of a chatbot that interactive with a Python FastAPI backend for sending and receiving data.

Red color box show the _buildInputArea method constructs a input area with a microphone icon for speech-to-text, a text field for user input, and a send button. It dynamically adjusts based on speech recognition state. Messages entered are sent to a Python FastAPI backend, enhancing user interaction in a chatbot.

THE OTHER SCREEN 6

 17:31 | 81.9KB/s Inbox Lance 10:00 AM good morning! Darrick Yesterday Hi are you available?	File Name: inboxPageCTRL.dart, inboxPage.dart Class: InboxPage, _InboxPageState, ChatModel Method: The InboxPage creates an inbox interface, displaying chat messages using a ListView.builder. Each message is presented in a ListTile with sender details and message content. The messages are fetched from the ChatModel class, which defines dummy data and encapsulates the structure of each chat message. The page is designed with a visually appealing layout, featuring avatars, timestamps, and rounded borders for message containers.

THE OTHER SCREEN 7



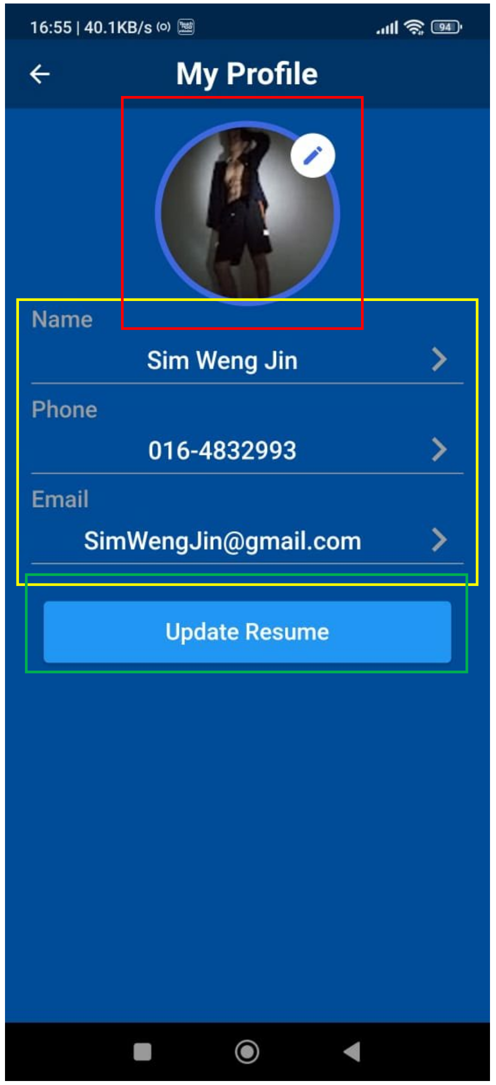
File Name: ProfilePage.dart, display_image_widget.dart

Class: ProfilePage, _ProfilePageState, DisplayImage

Method:

The ProfilePage is a user profile screen with a well-structured layout. It uses a Scaffold for the overall page structure, an AppBar for navigation, and a Column to organize the main content. The user information section includes a profile image (using display_image_widget.dart to display), name, and navigation buttons styled with ElevatedButton. These 3 ElevatedButton will navigate to different pages, such as myProfile, AboutUsPage, and SettingsPage.

THE OTHER SCREEN 8



File Name: myProfile.dart, display_image_widget.dart

Class: myProfile, _MyProfileState, DisplayImage

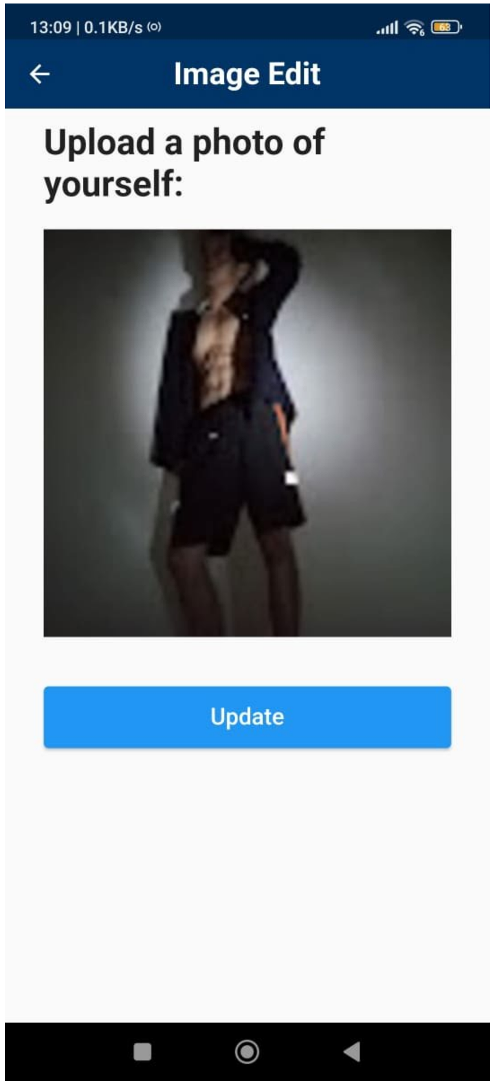
Method:

Red color box using display_image_widget.dart to display and useing InkWell for clickable and on tap will navigate to EditImagePage.

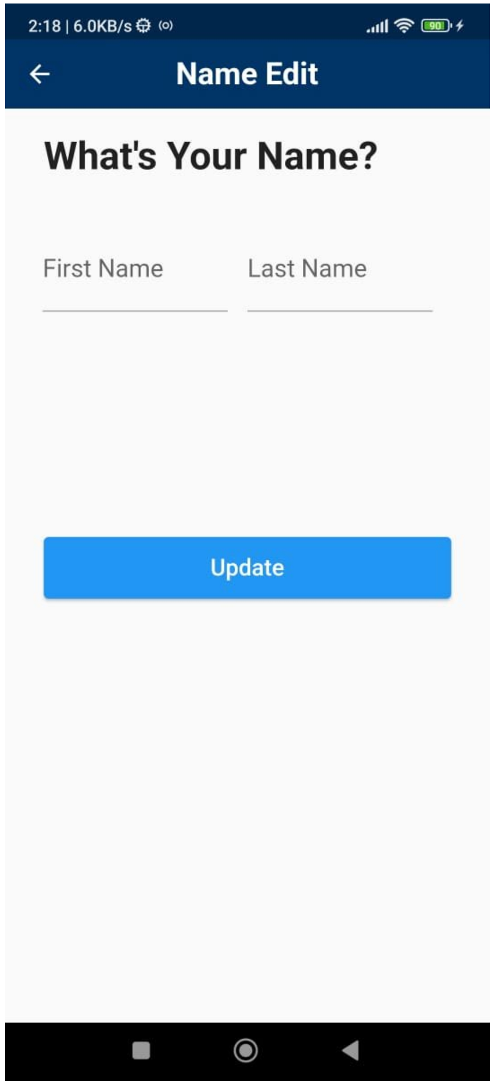
Yellow color box show the buildUserInfoDisplay(String getValue, String title, Widget editPage) method is a reusable Flutter widget-building function designed for the myProfile screen. getValue is get from user_data, title is user attribute like Name, Phone, Email, editPage is for navigate specify page such as EditNameFormPage, EditPhoneFormPage, EditEmailFormPage.

Green color box show the buildUpdateResumeButton constructs a styled button for updating the user's resume within the myProfile screen. ElevatedButton is on tap will navigate to EditResumePage

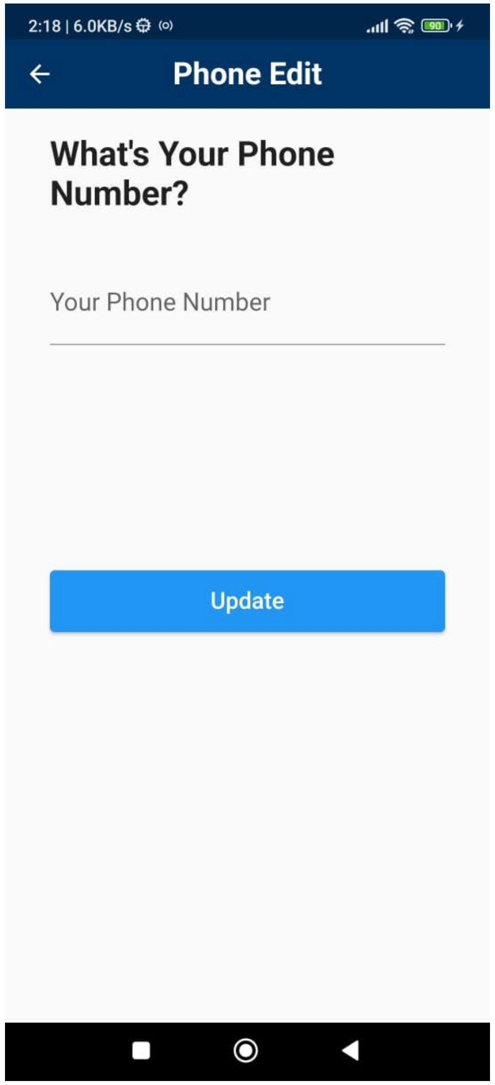
THE OTHER SCREEN 9

 13:09 | 0.1KB/s Image Edit Upload a photo of yourself: Update	File Name: edit_image.dart Class: EditImagePage, _EditImagePageState Method: The EditImagePage used GestureDetector allows users to upload a photo by tapping a image area and uses the ImagePicker to select an image from the gallery upon user interaction. The chosen image is copied to the app's directory and . ElevatedButton for update the image of User_data.

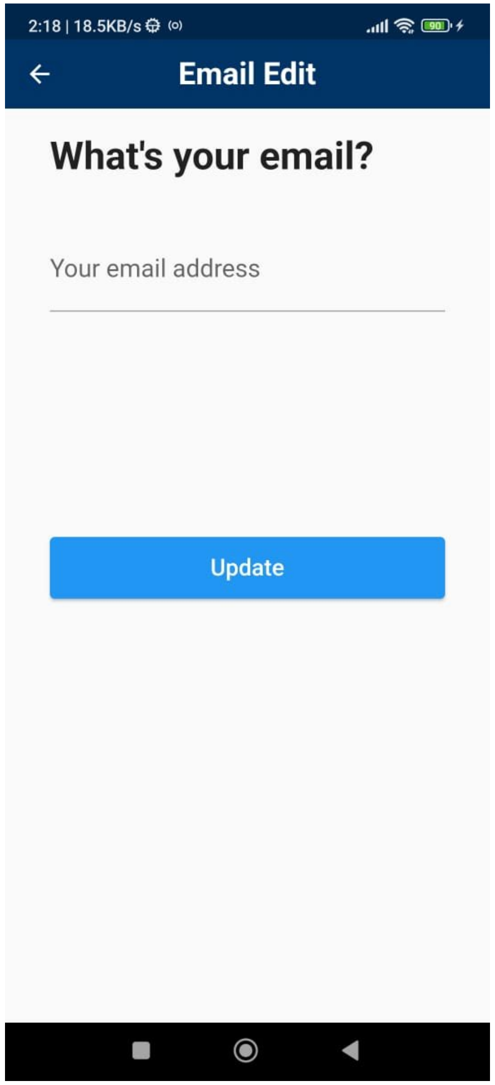
THE OTHER SCREEN 10

 2:18 | 6.0KB/s Name Edit What's Your Name? First Name Last Name Update	File Name: edit_name.dart Class: EditNameFormPage, EditNameFormPageState Method: The EditNameFormPage allow user edit the user's name. Within the body, a form is implemented with input fields for the first and last names. Form validation ensures that valid names, containing only letters, are entered. Upon validation, the "Update" button is enabled to invoke the update of the user's name with the concatenated first and last names.

THE OTHER SCREEN 11

 2:18 | 6.0KB/s Phone Edit What's Your Phone Number? Your Phone Number Update	File Name: edit_phone.dart Class: EditPhoneFormPage, EditPhoneFormPageState Method: The EditPhoneFormPage for user to edit phone number in valid format. Used TextFormField prompting the user to input their phone number, and a form field for data entry. The form used validator checks for empty, non-numeric, and insufficient-length inputs. Upon successful validation, the user's phone number is formatted, and the information is updated. The page also features an "Update" button that triggers the validation process and, if successful, navigates back to the previous screen.

THE OTHER SCREEN 12



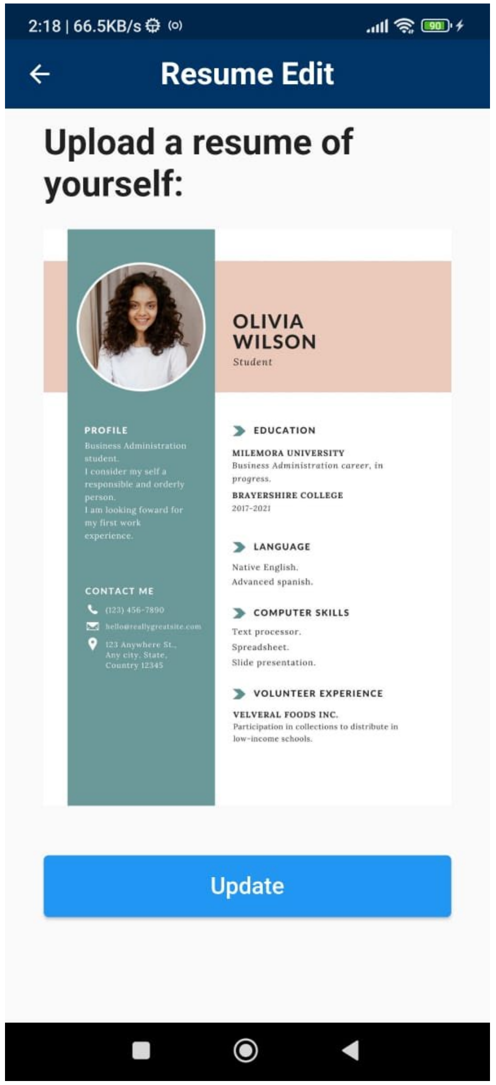
File Name: edit_email.dart

Class: EditEmailFormPage, EditEmailFormPageState

Method:

The EditEmailFormPage allow modification of the user's email address. The form includes a validation check to ensure that the email field is not left empty and used EmailValidator ensure that is the valid format input. Upon successful validation, the email is updated, and the user is navigated back to the previous screen. The page also includes an "Update" ElevatedButton, triggering the validation process and facilitating a seamless experience for users to edit their email information.

THE OTHER SCREEN 13



2:18 | 66.5KB/s

Resume Edit

Upload a resume of yourself:

OLIVIA WILSON

Student

PROFILE

Business Administration student.
I consider my self a responsible and orderly person.
I am looking foward for my first work experience.

CONTACT ME

(123) 456-7890
hello@reallygreatsite.com
123 Anywhere St., Any city, State, Country 12345

EDUCATION

MILEMORA UNIVERSITY
Business Administration career, in progress.
BRAYERSHIRE COLLEGE
2017-2021

LANGUAGE

Native English.
Advanced spanish.

COMPUTER SKILLS

Text processor.
Spreadsheet.
Slide presentation.

VOLUNTEER EXPERIENCE

VELVERAL FOODS INC.
Participation in collections to distribute in low-income schools.

Update

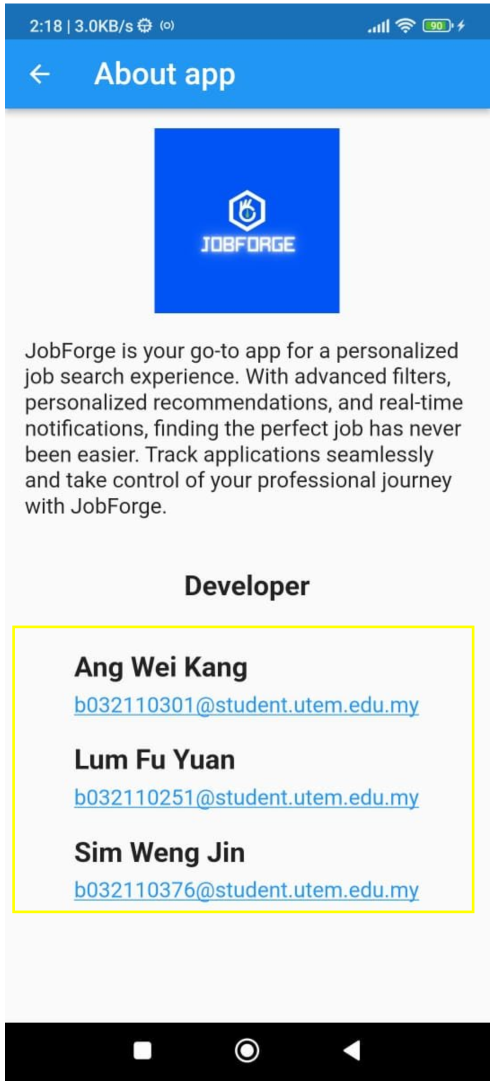
File Name: edit_resume.dart

Class: EditResumePage, _EditResumePageState

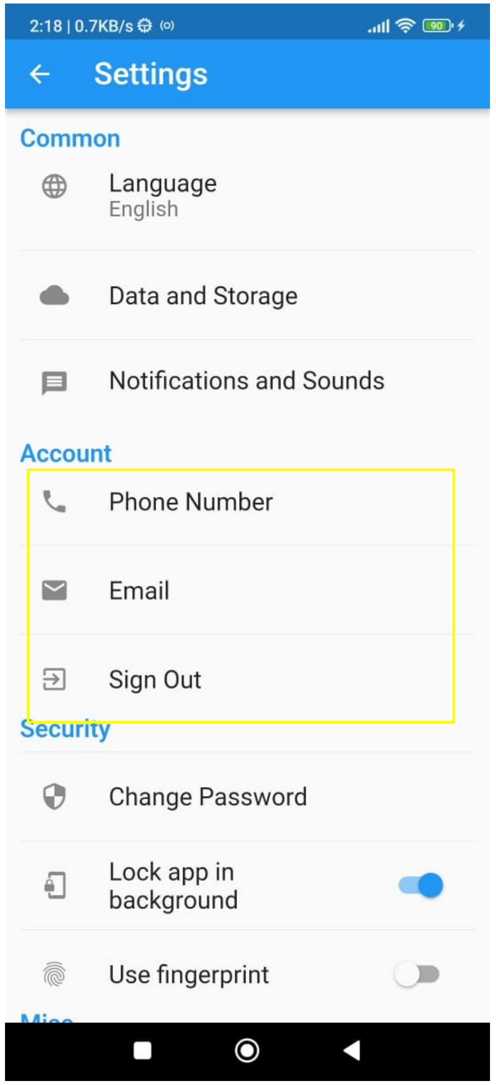
Method:

The EditResumePage allow user upload or update their resume(image format). A gesture detector is employed to detect taps on the specified area, invoking the ImagePicker to select a file from the gallery. Upon selecting a file, the page updates the user's resume path, and the selected resume is displayed. The widget also includes an "Update" ElevatedButton for update to database

THE OTHER SCREEN 14

 2:18 | 3.0KB/s About app JobForge is your go-to app for a personalized job search experience. With advanced filters, personalized recommendations, and real-time notifications, finding the perfect job has never been easier. Track applications seamlessly and take control of your professional journey with JobForge. Developer Ang Wei Kang b032110301@student.utem.edu.my Lum Fu Yuan b032110251@student.utem.edu.my Sim Weng Jin b032110376@student.utem.edu.my	File Name: AppinfoPage.dart Class: AboutUsPage Method: Yellow color box show the buildAuthor(String authorName, String websiteLink) method is a reusable Flutter widget-building function designed for the About App screen. The AboutUsPage Flutter widget creates a page displaying information about the app and its developers. It includes a logo, a description of the app's features, and details about each developer with clickable email links. The page is structured using a column layout, making it easily scrollable. The buildAuthor function organizes author information, and _launchURL enables launching URLs.

THE OTHER SCREEN 15



File Name: Setting.dart

Class: SettingsPage, _SettingsPageState

Method:

The SettingsPage utilizes a structured user interface with Column, Row, and ListTile widgets. The Column organizes content vertically, grouping settings into categories. Row arranges elements horizontally, facilitating the placement of headings and titles. ListTile components present individual settings with icons, titles and subtitle, offering a consistent and interactive layout. Only Yellow color box button can navigate to specify page such as EditPhoneFormPage, EditEmailFormPage, firstPage.