Dynamic theming(Adding UI Components)

Workshop #1

Flutter Developer Bootcamp

# **Purpose**

This workshop demonstrates how to implement dynamic theming functionality in a Flutter application.

# **Problem**

In the given workshop,it will display a screen with UI component like a button to toggle between light and dark themes.You need to create UI components such as:

* Circle Avatar:
* Text Field for Name:

1. Add a text field labeled "Name".
2. Provide a hint text saying "Enter your name".
3. Ensure appropriate styling for the text field.

* Text Field for Phone Number:

1. Include a text field labeled "Phone Number".
2. Provide a hint text saying "Enter your phone number".
3. Set the keyboard type to accept phone numbers.
4. Apply suitable styling to the text field.

* Text Field for Password:

1. Implement a text field labeled "Password".
2. Add a hint text prompting "Enter your password".
3. Ensure secure text entry for passwords.
4. Apply appropriate styling to the text field.

# 

# **How to Solve**

1. Checkout the workshop from Git Repo:

git clone -b <user-branch> <repo-URL>

2. Open the root folder inside VS Code

3. To build the app click the run option in the main method{}

4. Add various UI components like a circle avatar, text fields for name, phone number, and password.

5. Click the toggle button, you can see it can toggle between light theme and dark theme.

6. Go To File: <specific-file-with-theming-method> à <method-name>, implement your dynamic theming logic. Make the theming work

**You will Achieve**

When you complete this workshop you will learn the following:

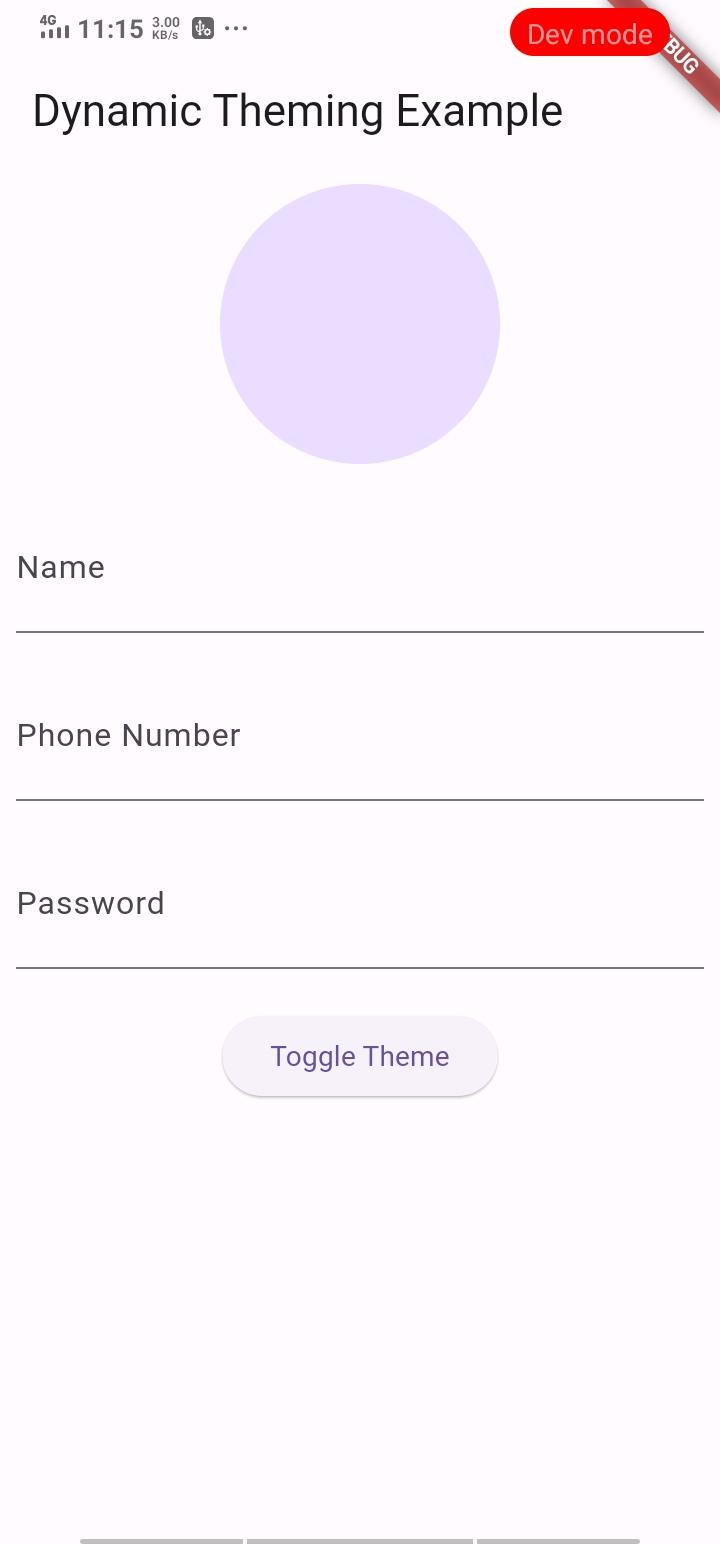
* **Toggle Theme**: You'll learn to implement a toggle switch for switching between different app themes, such as light and dark themes.
* **Checkboxes**: Incorporating checkboxes allows users to make selections or choices within your app.
* **Appbar**: Utilizing the app bar widget for navigation, branding, and displaying actions.
* **Text Fields**: Adding text fields enables users to input text or data into your app.
* **Decoration**: Understanding how to apply decoration to widgets for styling and visual enhancement.
* **Buttons**: Integrating buttons for triggering actions or navigating within the app.
* **Circle Avatar**: Displaying circular profile images or placeholders for user avatars.
* **Icons**: Utilizing icons to provide visual representation and enhance user interaction.
* **Padding**: Adjusting spacing around widgets to improve layout aesthetics and readability.
* **Sizedbox**: Managing widget sizes and spacing within your app's layout.
* **Text**: Displaying text content within your app's interface.
* **Label Text and Hint Text**: Differentiating between label text (text displayed alongside form fields) and hint text (text displayed within form fields as a hint).
* **Adjusting Text Alignment**: Controlling the alignment of text elements to enhance readability and design consistency.

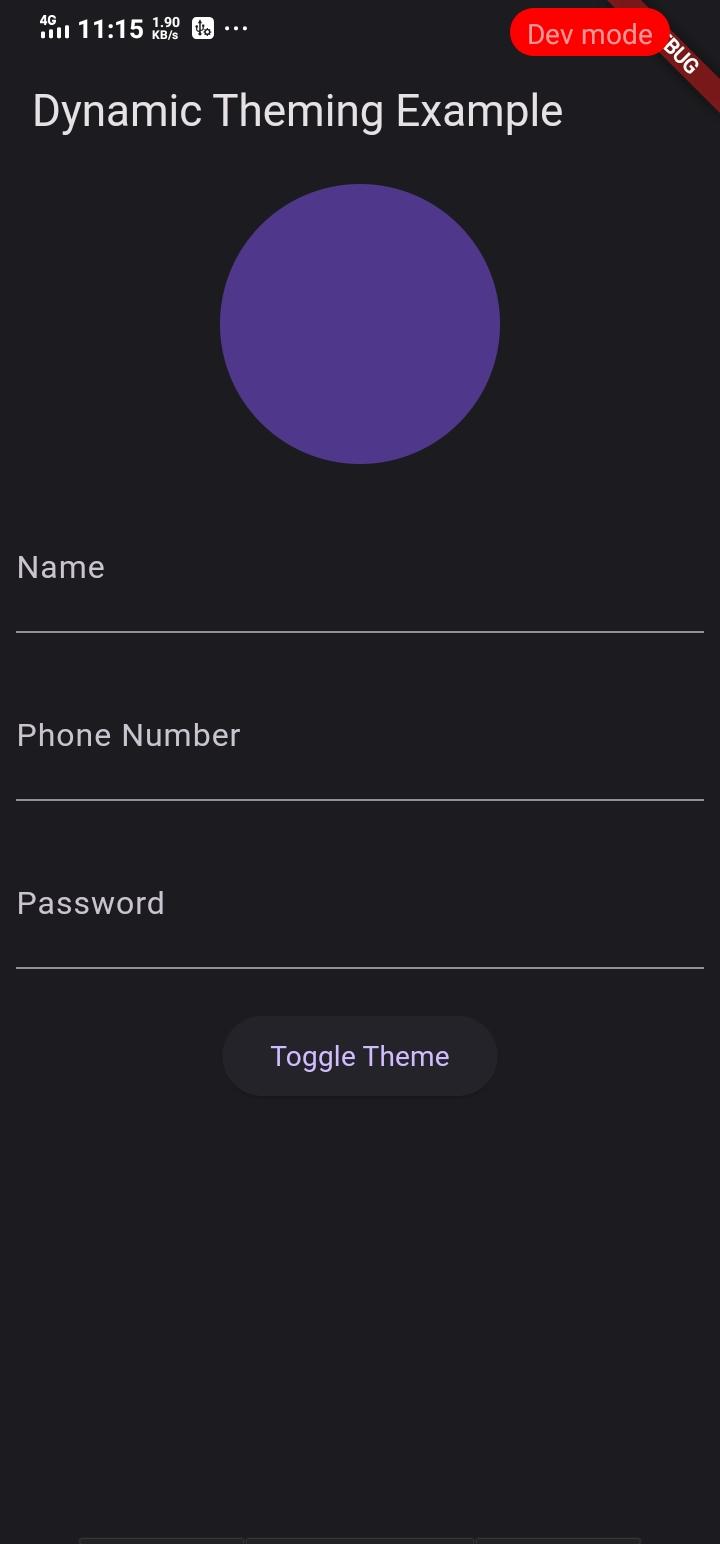
# **Screenshots**

## **Before implementation (without UI components)**

## 

## **After implementation (with UI components)**





# **How to submit your workshop**

Push your project back to the same git branch using command:

<command name>

# **Happy Coding!**