Flutter Test

Soch College of IT

2 June 2024

Task: Digital Clock Application in Flutter

Objective:

Create a digital clock application using Flutter that displays the current date and time. The time should be shown in a 12-hour format with AM/PM, and the date should be fetched from an API. The application should be well-designed, synchronized with network time, and include features for refreshing the time and error handling.

API Details:

Use the following API to obtain the current date and time:

- URL: http://worldtimeapi.org/api/timezone/Asia/Kathmandu
- Sample Response:

```
{
"abbreviation": "+0545",
"client_ip": "49.244.40.247",
"datetime": "2024-06-01T18:34:30.405119+05:45",
"day_of_week": 6,
  "day_of_year": 153,
  "dst": false,
"dst_from": null,
"dst_offset": 0,
  "dst_until": null,
  "raw_offset": 20700,
  "timezone": "Asia/Kathmandu",
"unixtime": 1717246170,
"utc_datetime": "2024-06-01T12:49:30.405119+00:00",
  "utc_offset": "+05:45",
"week_number": 22
}
```

Marking Criteria:

The task is divided into the following marking criteria:

1. **UI Design of the Clock (30 marks)**
   - The clock must display the current time in 12-hour format with AM/PM.
   - The date must be displayed below the time.
   - The design should be aesthetically pleasing and user-friendly.
   - Use appropriate colors, fonts, and layout.
2. **Synchronizing the Clock with Network Time and Date (25 marks)**
   - Correctly fetch the date and time from the provided API.
   - Parse the response to extract the necessary date and time information.
   - Display the fetched date and time in the application.
3. **Real-Time Updating of Time (20 marks)**
   - The time should update every second without requiring a manual refresh.
   - Implement a timer to increment the time accurately in the app.
4. **Refresh Button for Time Synchronization (10 marks)**
   - Provide a button that, when pressed, refreshes the time by re-fetching data from the API.
   - Ensure the button works without causing the application to crash.
5. **Error Handling (10 marks)**
   - Implement error handling for network requests.
   - Display appropriate error messages to the user if the API request fails or if there is a connectivity issue.
   - Ensure the application remains functional and stable under error conditions.
6. **Code Organization and Best Coding Practices (5 marks)**
   - Follow best practices for code organization, such as separating UI code, business logic, and API calls.
   - Use descriptive variable names and comments to explain your code.
   - Ensure the code is clean, readable, and follows Dart and Flutter conventions.

Notes for Implementation:

- Use the **http** package to make network requests.
- Parse the JSON response using Dart's **dart:convert** library.
- To parse and format the date and time from the API response in Flutter, you can use the DateTime and DateFormat classes from the **intl** package, or use any ideas as you like.
  - DateTime Object.minute => get minutes
  - DateTime Object.second => get seconds
  - DateTime Object.add(Duration(..)) => add duration to current datetime
  - .. and so on

- Use the Future.delayed() function or **Timer** class from the **dart:async** library to update the time every second.
- Use try-catch blocks to handle exceptions.
- Use descriptive names for variables, functions, and classes.
- Comment your code to explain the purpose of complex sections.
- Add necessary network permissions in the **AndroidManifest.xml** file:

```
<uses-permission android:name="android.permission.INTERNET"/>
<application ...
```

- Build the APK file using **flutter build apk**.
- Copy/Paste the generated .apk file in project's root folder and submit the task.

Submission:

- **GitHub Repository:**
  - Create a public repository on GitHub and upload your complete Flutter project source code.
  - Upload the source code along with the generated APK file to the GitHub repository.
  - Show the output manually to the instructor before leaving the exam hall.
- **Google Form:**
  - Share the link to your public GitHub repository in this Google Form: https://forms.gle/HvknaPFqphoxZ7d87.

Good luck, and happy coding!