Assignment No.: 2
Title: Designing Flutter UI Using Common Widgets
Name: Harshit Raheja
Class: D15B
Roll Number: 45

Aim: To design a Flutter UI by including commonly used widgets like Column, Row, Stack, Container, and ListView.

Theory: Flutter is a UI toolkit by Google for crafting natively compiled applications for mobile, web, and desktop from a single codebase. In this experiment, we use common widgets that form the building blocks of most Flutter UIs. These include layout widgets like Row and Column, display widgets like Text and Image, and interactivity widgets like ElevatedButton and IconButton.

Widgets and Their Properties:

1. **Column**: Arranges children vertically. Properties: mainAxisAlignment, crossAxisAlignment, children.
2. **Row**: Arranges children horizontally. Properties: mainAxisAlignment, crossAxisAlignment, children.
3. **Stack**: Allows widgets to overlap. Properties: alignment, children.
4. **Container**: A box model with margin, padding, border, and color.
5. **ListView**: A scrollable list of widgets. Properties: children, scrollDirection.

Code:

```
// main.dart
DropdownButtonFormField<String>(
 decoration: InputDecoration(
  labelText: "Source Station",
  prefixIcon: Icon(Icons.train, color: Colors.black),
  contentPadding: EdgeInsets.symmetric(horizontal: 16, vertical: 12),
 ),
 value: _source,
 items: _stations.map((station) {
  return DropdownMenuItem(
   value: station['value'],
   child: Text(station['label']!),
  );
 }).toList(),
 onChanged: (value) => setState(() => _source = value),
```

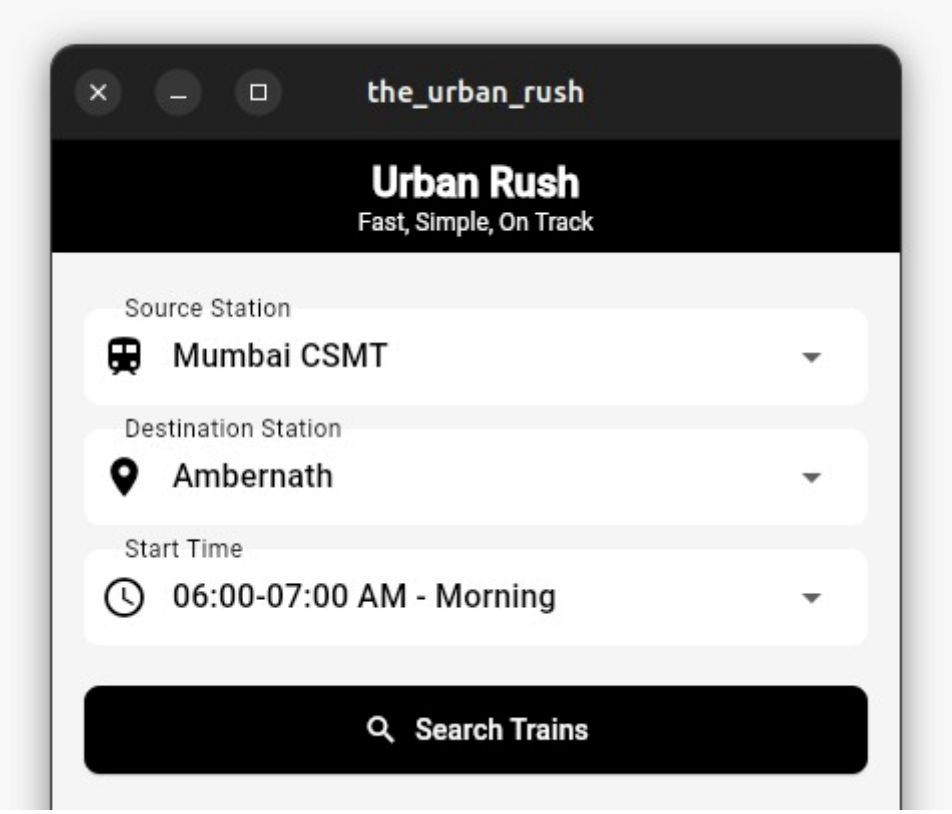
```
),

SizedBox(height: 12),

DropdownButtonFormField<String>(
 decoration: InputDecoration(
  labelText: "Destination Station",
  prefixIcon: Icon(Icons.location_on, color: Colors.black),
  contentPadding: EdgeInsets.symmetric(horizontal: 16, vertical: 12),
 ),
 value: _destination,
 items: _stations.map((station) {
  return DropdownMenuItem(
   value: station['value'],
   child: Text(station['label']!),
  );
 }).toList(),
 onChanged: (value) => setState(() => _destination = value),
),

SizedBox(height: 12),

DropdownButtonFormField<String>(
 decoration: InputDecoration(
  labelText: "Start Time",
  prefixIcon: Icon(Icons.access_time, color: Colors.black),
  contentPadding: EdgeInsets.symmetric(horizontal: 16, vertical: 12),
 ),
 value: _time,
 items: _timeSlots.map((slot) {
  return DropdownMenuItem(
   value: slot['value'],
   child: FittedBox(
    fit: BoxFit.scaleDown,
    alignment: Alignment.centerLeft,
    child: Text(slot['label']!),
   ),
  );
 }).toList(),
 onChanged: (value) => setState(() => _time = value),
),
```

```
SizedBox(height: 20),

SizedBox(
 width: double.infinity,
 child: ElevatedButton.icon(
  onPressed: _searchTrains,
  icon: Icon(Icons.search, color: Colors.white),
  label: Text("Search Trains", style: TextStyle(color: Colors.white)),
  style: ElevatedButton.styleFrom(
   backgroundColor: Colors.black,
   padding: EdgeInsets.symmetric(vertical: 20),
   shape: RoundedRectangleBorder(
    borderRadius: BorderRadius.circular(8),
   ),
  ),
 ),
),

SizedBox(height: 20),

if (_trainSchedule.isNotEmpty)
 ListView.builder(
  shrinkWrap: true,
  physics: NeverScrollableScrollPhysics(),
  itemCount: _trainSchedule.length,
  itemBuilder: (context, index) {
   var train = _trainSchedule[index];
   return GestureDetector(
    onTap: () {
     Navigator.push(
      context,
      MaterialPageRoute(
       builder: (context) =>
         TrainDetailsScreen(railId: train['railid']),
      ),
     );
    },
    child: Card(
     shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(8),
```

```
),
margin: EdgeInsets.symmetric(vertical: 8),
elevation: 2,
child: Padding(
 padding: const EdgeInsets.all(16.0),
 child: Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: [
   Text(
    '${train['journey_start']} ${train['journey_from']} - ${train['journey_to']}',
    style: TextStyle(
     fontSize: 16,
     fontWeight: FontWeight.w600,
     color: Colors.black,
    ),
   ),
   SizedBox(height: 6),
   Text(
    '${train['railid']}: ${train['train_name']}',
    style: TextStyle(
     fontSize: 14,
     color: Colors.grey[700],
    ),
   ),
   SizedBox(height: 6),
   Wrap(
    crossAxisAlignment: WrapCrossAlignment.center,
    children: [
     Text(
      'Arrival: ${train['journey_end']}, ',
      style: TextStyle(fontSize: 13),
     ),
     Text(
      '${train['train_speed']}',
      style: TextStyle(
       fontSize: 13,
       fontWeight: FontWeight.bold,
       color: train['train_speed'].toString().toLowerCase() == 'slow'
          ? Colors.green
          : train['train_speed'].toString().toLowerCase() == 'fast'
            ? Colors.red
```

```
                    : Colors.blue,
                ),
              ),
              Text(
                ' | Journey Duration: ${train['journey_duration']} min',
                style: TextStyle(fontSize: 13),
              ),
            ],
          ),
        ],
      ),
    ),
  ),
 );
 },
),
```

Output:



Conclusion: A functional UI was successfully created in Flutter using commonly used widgets. The interface is composed using structural widgets like Column and Row, and interactive components like ElevatedButton and IconButton, ensuring usability and layout responsiveness.