BuildContext를 비지니스 로직으로 넘기지 말기

BuildContext는 UI 레이어에서만 다루어야 합니다.

절대로 비지니스 로직 레이어에 BuildContext를 넘기면 안됩니다.

> 엥? 그러면 dialog같은건 어떻게 띄우나요?

→ 로직 레이어로부터 응답 결과를 받아서 해당 결과에 따라 dialog를 띄웁니다.

> ref는 넘겨도 되나요?

→ 그런 일이 없었으면 좋겠지만.. 있게 된다면 `ref.watch, [ref.select](<http://ref.select>)` 이 두가지는 위젯에서만 사용해야하고, 사용하게 된다면 `ref.read` 로 접근하여 사용해 주세요.

print 사용하지 말기

print 문을 사용하지 마세요.

디버깅 목적이라면 개발 단계에서만 사용하고, 릴리즈 전에 모두 제거해야 합니다.

대신 debugPrint나 log를 사용하세요.

메모리 누수 방지하기

특정 stateful widget에서 다음과 같은 객체를 사용할 때:

- `AnimationController`
- `TextEditingController`
- `ScrollController`
- `FocusNode`
- `StreamSubscription`

반드시 `dispose()` 메서드에서 해당 리소스를 정리해야 합니다.

```
@override
void dispose() {
  // 컨트롤러 정리
  _animationController.dispose();
  _textEditingController.dispose();
  _scrollController.dispose();
  _focusNode.dispose();

  // 구독 취소
  _subscription.cancel();

  super.dispose();
}
```

이렇게 하지 않으면 메모리 누수가 발생합니다.

위젯 트리 깊이 관리하지 않기

위젯 트리를 너무 깊게 쌓아가지 마세요.

```
// 이렇게 하지 마세요
return Container(
  child: Column(
    children: [
      Container(
        child: Row(
          children: [
            Container(
              child: Text('깊은 위젯 트리'),
            ),
          ],
        ),
      ),
    ],
  ),
);
```

이렇게 하면:

1. 가독성이 크게 떨어집니다.
2. rebuild 시 성능 문제가 발생합니다.

대신 위젯을 적절히 분리하세요:

```
...
return MainContainer(
  child: HeaderSection(),
);
...

class HeaderSection extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Row(
      children: [
        Text('분리된 위젯'),
      ],
    );
  }
}
```

반응형 레이아웃 사용하기

디바이스 크기는 매우 다양하므로 항상 상대적으로 레이아웃을 구성하세요.

Figma에서 제공되는 디자인을 고정 크기로 그대로 구현하면 다음과 같은 문제가 발생합니다:

- overflow 에러
- 작은 화면에서 레이아웃 깨짐
- 큰 화면에서 공간 낭비

대신:

- `MediaQuery`를 사용하여 디바이스 크기 확인
- `Expanded`, `Flexible` 위젯 활용
- `LayoutBuilder`로 상위 위젯 제약조건 활용

- `AspectRatio`로 비율 유지

```
// 좋은 예시
Widget build(BuildContext context) {
  return LayoutBuilder(
    builder: (context, constraints) {
      return Container(
        width: constraints.maxWidth * 0.8,  // 화면 너비의 80%
        child: Column(
          children: [
            Expanded(
              flex: 2,
              child: HeaderWidget(),
            ),
            Expanded(
              flex: 3,
              child: ContentWidget(),
            ),
          ],
        ),
      );
    }
  );
}
```