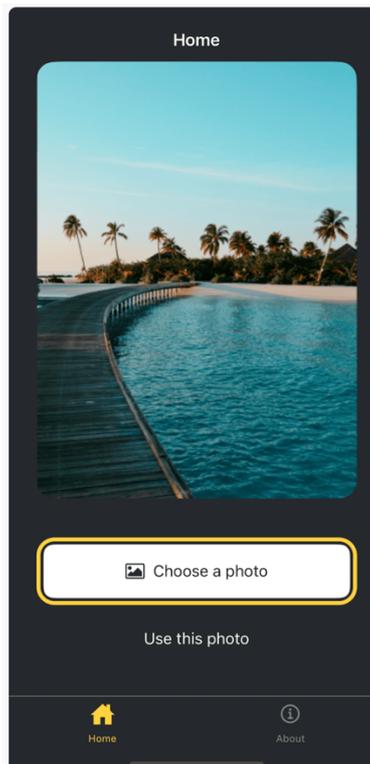


# Лекція-практикум 3

## Будуємо екран

У цій лекції ми навчимося використовувати компоненти, такі як React Native's `Pressable` та `Expo Image`, щоб побудувати екран.

Створимо перший екран у застосунку **StickerSmash**.



Екран має показувати зображення та дві кнопки. Користувач може:

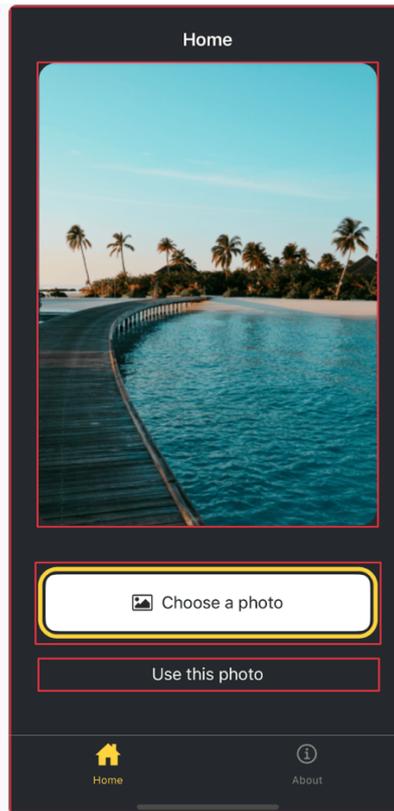
- вибрати зображення зі свого пристрою, або
- продовжити з дефолтним зображенням, яке вже є в застосунку.

Після вибору зображення користувач може додавати стікери до нього. Тож почнемо створення цього екрану.

---

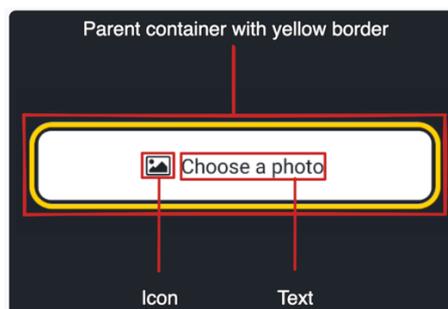
### 1. Розберімо екран на частини

Перш ніж писати код, подивимось, з яких елементів складається екран:



- Велике зображення в центрі екрана.
- Дві кнопки в нижній частині екрану.

Перша кнопка має жовту рамку та містить іконку і текст в одному ряду.



Після цього ми можемо написати код, використовуючи ці компоненти як будівельні блоки.

## 2. Відобразити зображення

Ми будемо використовувати бібліотеку `expo-image` для відображення зображення — вона дає кросплатформений компонент `<Image>`.

1. Зупини сервер розробки (`Ctrl + C` у терміналі).

## 2. Встанови бібліотеку:

```
npx expo install expo-image
```

## 3. Потім знову запусти сервер (npx expo start).

Компонент Image має source — це може бути:

- статичний ресурс (наприклад, з папки assets/images), або
- URL (uri) з мережі.

У файлі app/(tabs)/index.tsx імпортуємо Image і використовуємо його:

```
import { View, StyleSheet } from 'react-native';
import { Image } from 'expo-image';

const PlaceholderImage = require('@assets/images/background-image.png');

export default function Index() {
  return (
    <View style={styles.container}>
      <View style={styles.imageContainer}>
        <Image source={PlaceholderImage} style={styles.image} />
      </View>
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#25292e',
    alignItems: 'center',
  },
  imageContainer: {
    flex: 1,
  },
  image: {
    width: 320,
    height: 440,
    borderRadius: 18,
  },
});
```

## 3. Поділ компонентів на окремі файли

Коли код розростається, зручно винести частини екрану в окремі компоненти:

1. Створи папку `components/`, і у ній — файл `ImageViewer.tsx`.
2. Перенеси туди код відповідальності за зображення:

```
import { ImageSourcePropType, StyleSheet } from 'react-native';
import { Image } from 'expo-image';

type Props = {
  imgSource: ImageSourcePropType;
};

export default function ImageViewer({ imgSource }: Props) {
  return <Image source={imgSource} style={styles.image} />;
}

const styles = StyleSheet.create({
  image: {
    width: 320,
    height: 440,
    borderRadius: 18,
  },
});
```

Важливо: компоненти, які не відповідають за навігацію, не повинні знаходитися в папці `app/`, яка призначена для маршрутів та лейаутів. Такі компоненти краще тримати поза `app/`.

В `app/ (tabs) /index.tsx` імпортуй `ImageViewer` і використовуй його:

```
import { StyleSheet, View } from 'react-native';

import ImageViewer from '@components/ImageViewer';

const PlaceholderImage = require('@assets/images/background-image.png');

export default function Index() {
  return (
    <View style={styles.container}>
      <View style={styles.imageContainer}>
        <ImageViewer imgSource={PlaceholderImage} />
      </View>
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#25292e',
    alignItems: 'center',
  },
  imageContainer: {
    flex: 1,
  },
});
```

#### 4. Створіть кнопки за допомогою Pressable

У React Native є кілька компонентів для обробки торкань, але **<Pressable>** рекомендований завдяки гнучкості: він уміє реагувати на короткі/довгі натискання, окремо відслідковувати натиск і відпускання тощо. У дизайні нам потрібно дві кнопки з різним стилем і підписом. Почнемо зі **спільного багаторазового компонента**.

Створіть файл `components/Button.tsx` з таким кодом:

```
import { StyleSheet, View, Pressable, Text } from 'react-native';

type Props = {
  label: string;
};

export default function Button({ label }: Props) {
  return (
    <View style={styles.buttonContainer}>
      <Pressable style={styles.button} onPress={() => alert('You pressed a button.')}>
        <Text style={styles.buttonLabel}>{label}</Text>
      </Pressable>
    </View>
  );
}

const styles = StyleSheet.create({
  buttonContainer: {
    width: 320,
    height: 68,
    marginHorizontal: 20,
    alignItems: 'center',
    justifyContent: 'center',
    padding: 3,
  },
  button: {
    borderRadius: 10,
    width: '100%',
    height: '100%',
    alignItems: 'center',
    justifyContent: 'center',
    flexDirection: 'row',
  },
  buttonLabel: {
    color: '#fff',
    fontSize: 16,
  },
});
```

Підключіть компонент у `app/ (tabs) /index.tsx` і додайте стилі для контейнера з кнопками:

```
import { View, StyleSheet } from 'react-native';

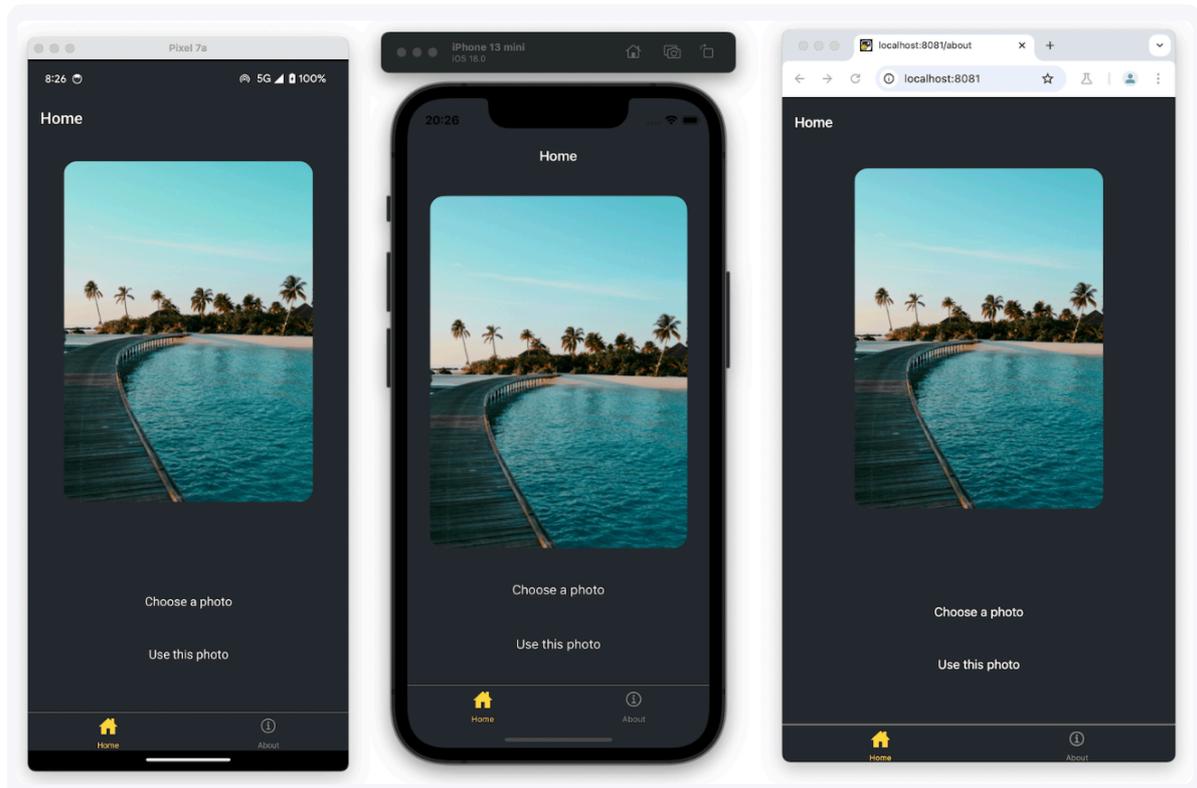
import Button from '@components/Button';
import ImageViewer from '@components/ImageViewer';

const PlaceholderImage = require("@/assets/images/background-image.png");

export default function Index() {
  return (
    <View style={styles.container}>
      <View style={styles.imageContainer}>
        <ImageViewer imgSource={PlaceholderImage} />
      </View>
      <View style={styles.footerContainer}>
        <Button label="Choose a photo" />
        <Button label="Use this photo" />
      </View>
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#25292e',
    alignItems: 'center',
  },
  imageContainer: {
    flex: 1,
    paddingTop: 28,
  },
  footerContainer: {
    flex: 1 / 3,
    alignItems: 'center',
  },
});
```

Тепер друга кнопка (“Use this photo”) уже схожа на дизайн. Першу треба додатково стилізувати.



## 5. Поліпште багаторазовий компонент кнопки

Кнопка “**Choose a photo**” має інший стиль, тож додамо **prop theme** (наприклад, 'primary') і **іконку** перед текстом за допомогою бібліотеки @expo/vector-icons (візьмемо FontAwesome). Оновіть `components/Button.tsx`:

```
import { StyleSheet, View, Pressable, Text } from 'react-native';
import FontAwesome from '@expo/vector-icons/FontAwesome';

type Props = {
  label: string;
  theme?: 'primary';
};

export default function Button({ label, theme }: Props) {
  if (theme === 'primary') {
    return (
      <View
        style={[
          styles.buttonContainer,
          { borderWidth: 4, borderColor: '#ffd33d', borderRadius: 18 },
        ]}>
        <Pressable
          style={[styles.button, { backgroundColor: '#fff' }]}
          onPress={() => alert('You pressed a button.')}>
          <FontAwesome name="picture-o" size={18} color="#25292e" style={styles.buttonIcon} />
          <Text style={[styles.buttonLabel, { color: '#25292e' }]}>{label}</Text>
        </Pressable>
      </View>
    );
  }

  return (
    <View style={styles.buttonContainer}>
      <Pressable style={styles.button} onPress={() => alert('You pressed a button.')}>
        <Text style={styles.buttonLabel}>{label}</Text>
      </Pressable>
    </View>
  );
}
```

```
const styles = StyleSheet.create({
  buttonContainer: {
    width: 320,
    height: 68,
    marginHorizontal: 20,
    alignItems: 'center',
    justifyContent: 'center',
    padding: 3,
  },
  button: {
    borderRadius: 10,
    width: '100%',
    height: '100%',
    alignItems: 'center',
    justifyContent: 'center',
    flexDirection: 'row',
  },
  buttonIcon: {
    paddingRight: 8,
  },
  buttonLabel: {
    color: '#fff',
    fontSize: 16,
  },
});
```

І використайте тему для першої кнопки в `app/ (tabs) / index.tsx`:

```
import { View, StyleSheet } from 'react-native';

import Button from '@components/Button';
import ImageViewer from '@components/ImageViewer';

const PlaceholderImage = require('@assets/images/background-image.png');

export default function Index() {
  return (
    <View style={styles.container}>
      <View style={styles.imageContainer}>
        <ImageViewer imgSource={PlaceholderImage} />
      </View>
      <View style={styles.footerContainer}>
        <Button theme="primary" label="Choose a photo" />
        <Button label="Use this photo" />
      </View>
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#25292e',
    alignItems: 'center',
  },
  imageContainer: {
    flex: 1,
  },
  footerContainer: {
    flex: 1 / 3,
    alignItems: 'center',
  },
});
```

Після цього перша кнопка матиме жовту рамку та іконку, як у макеті; друга — темну заливку.

