

# Лекція-практикум 1. Створюємо перший мобільний застосунок на Expo (React Native)

## Навчальні цілі

- Розгорнути новий проєкт Expo з шаблону.
- Запустити застосунок на телефоні (Expo Go) та у браузері.
- Зрозуміти базову структуру (file-based routing) і відредагувати перший екран.
- Ознайомитися з dev-циклом Expo (зміна коду → миттєве оновлення).

## 1) Передумови (Prerequisites)

- **Expo Go** на фізичному пристрої (Android/iOS).
- **Node.js (LTS)**.
- Редактор коду (**VS Code** або інший).
- ОС: macOS / Linux / Windows (PowerShell; для Linux середовищ у Windows — WSL2)

## 2) Ініціалізація проєкту (Create a new Expo app)

У терміналі:

```
# Create a project named StickerSmash
- npx create-expo-app@latest StickerSmash

# Navigate to the project directory
- cd StickerSmash
```

## 3) Підготовка матеріалів (Assets)

1. Завантажити архів з активами для туторіалу.
2. Розпакувати і **замінити** дефолтні зображення у `your-project-name/assets/images`.
3. Відкрити проєкт у редакторі коду.

## 4) Очищення шаблону (reset-project)

Щоб будувати застосунок «з нуля», приберемо зайвий boilerplate:

```
- npm run reset-project
```

Скрипт залишить у app лише `index.tsx` та `_layout.tsx`, а попередні приклади перенесе в `app-example` (її можна видалити).

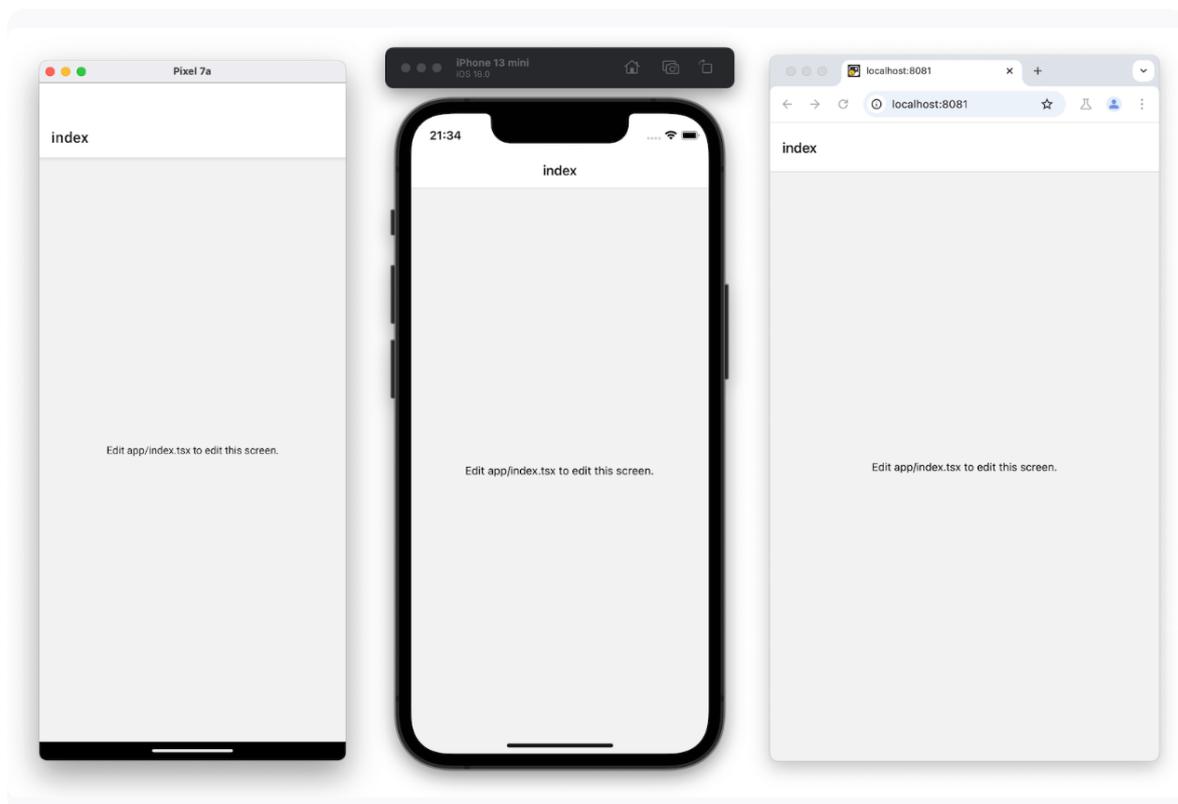
## 5) Запуск на телефоні та у вебi

У корені проєкту:

```
- npx expo start
```

Далі:

1. У терміналі з'явиться **QR-код**.
2. Скануємо його через **Expo Go** (Android — “Scan QR code”; iOS — стандартна **Камера**).
3. Для веб-версії натисніть **w** у терміналі — відкриється у браузері



## 6) Перше редагування екрана (`index.tsx`)

Сутність: головний екран у `app/index.tsx` показує текст. Змінимо фон і текст:

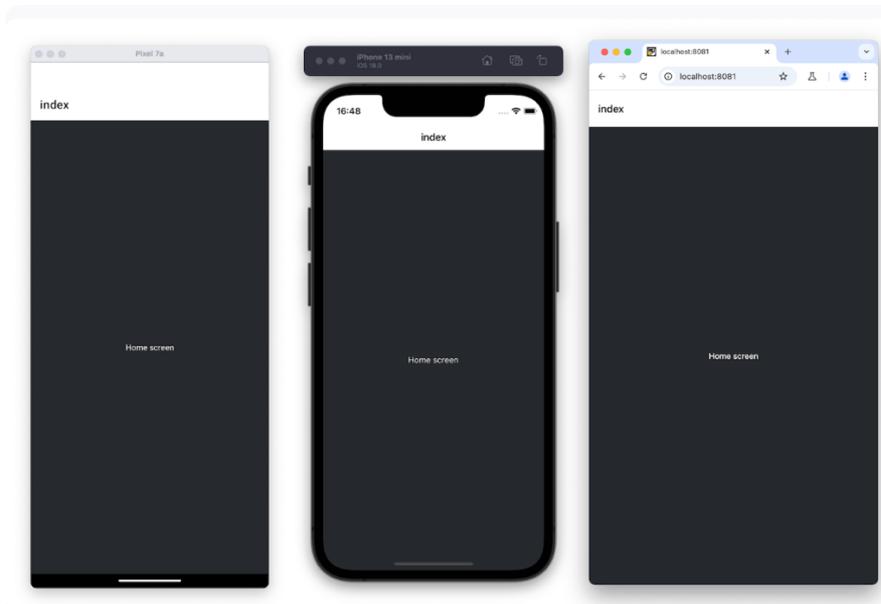
```
import { Text, View, StyleSheet } from 'react-native';

export default function Index() {
  return (
    <View style={styles.container}>
      <Text style={styles.text}>Home screen</Text>
    </View>
  );
}

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

## 7) Контрольні точки (що має вийти)

- Проєкт запускається на Android/iOS через Expo Go і у браузері.
- На екрані — текст **Home screen** з білим кольором, фон — #25292e.



## 8) Типові помилки та швидкі рішення

- **Не бачите QR-коду / немає підключення:** перевірте, що телефон і ПК в одній мережі; перезапустіть `npm expo start`.
- **Порожній екран у вебi:** перевірте синтаксис у `index.tsx` (незакриті дужки/теги), перезавантажте сторінку.
- **Помилка Node версії:** оновіть Node до LTS (рекомендовано для Expo).

## Прикінцеві положення: Налаштування Node.js

### • Вибір версії Node.js

Для стабільної роботи інструментів (Expo, React Native, npm) рекомендовано використовувати **LTS-версію Node.js** (Long-Term Support). Це гарантує сумісність та довготривалу підтримку.

### • Перевірка встановлення

Після інсталяції перевірте доступність Node.js і npm у терміналі:

```
node -v
npm -v
```

- Ви повинні отримати номери версій без помилок.

### • Менеджери версій (nvm / nvm-windows)

Для уникнення конфліктів і легкого перемикання між версіями Node.js варто використовувати **Node Version Manager (nvm)**. Це особливо корисно при роботі з різними проектами.

- Установка для Linux/macOS:

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/master/install.sh | bash
```

- Для Windows: [nvm-windows](#)

### • Оновлення Node.js

Якщо необхідна нова версія:

- через nvm:

```
nvm install node  
nvm use node
```

- 

або вручну, завантаживши інсталятор з офіційного сайту.

- **Перевірка середовища**

Для успішного запуску проєктів переконайтеся, що Node.js, npm (або yarn/pnpm) та Expo CLI працюють узгоджено:

```
npm expo --version
```

### Рекомендації

- Завжди тримайте робоче середовище актуальним.
- Для навчальних і лабораторних робіт використовуйте **LTS-версію**.
- У випадку конфліктів між бібліотеками — перевірте сумісність версій у документації Expo та React Native.