

תרגול: אפליקציית To-Do בסיסית ב-React Native

מטרת התרגיל 

בניית אפליקציית To-Do פשוטה המציגה הוספה, סימון וסינון משימות, תוך יישום עקרונות של קומפוננטות, `FlatList`, `Props`, `State`, `StyleSheet`.
הדגש: קוד קריא, מבנה פרויקט מסודר, ורכיבים לשימוש חוזר.

דרישות כלליות

- יצירת, הצגה וניהול רשימת משימות.
- כל משימה כוללת: מזהה (`id`), טקסט (`title`), מצב ביצוע (`completed: boolean`), ותאריך יצירה.
- עבודה ללא ספריות חיצוניות (מעבר ל-Expo/React Native).
- תמיכה בסיסית במכשירי Android ו-iOS (אמולטור/מכשיר אמיתי).
- שימוש ב-TypeScript.

קומפוננטות נדרשות

TaskItem

מציג שורה של משימה עם סימון/ביטול וסמל מחיקה.
:Props

- `id: string`
- `title: string`
- `completed: boolean`
- `onToggle(id: string): void`
- `onDelete(id: string): void`

TaskInput

שורת קלט להוספת משימה חדשה עם כפתור "הוסף".
:Props

placeholder?: string •

onSubmit(title: string): void •

disabled?: boolean (אופציונלי) •

TaskFilterBar

כפתורי סינון: All / Active / Completed.

: Props

'value: 'all' | 'active' | 'completed' •

onChange(value): void •

Card (אופציונלי אך מומלץ לעיצוב אחיד)

מעטפת עם רקע/צל לשימוש חוזר.

: Props

title?: string •

children: React.ReactNode •

מסך ראשי: ToDoScreen

מבנה מוצע:

1. כותרת הדף (Text גדול).

2. **TextInput** להוספה.

3. **TaskFilterBar** לסינון תצוגה.

4. רשימת משימות באמצעות **TaskItem** (**FlatList** לכל פריט).

5. טקסט מצב בתחתית: "סה"כ X | פעיל Y | הושלם Z".

התנהגויות:

• הוספה: אם הטקסט לא ריק → יצירת משימה חדשה בראש הרשימה.

• **Toggle**: לחיצה על המשימה/ציקבוקס תחליף מצב **completed**.

- מחיקה: אייקון 🗑️ ימחק פריט.
- סינון: משנה את מערך הנתונים שמועבר ל-FlatList.

🏠 מבנה פרויקט מומלץ

```

      /src
      |
      |--- /components
      |      |
      |      |--- TaskItem.tsx
      |      |--- TaskInput.tsx
      |      |--- TaskFilterBar.tsx
      |      |--- (Card.tsx optional)
      |      |
      |      |--- /screens
      |      |      |
      |      |      |--- ToDoScreen.tsx
      |      |      |
      |      |      |--- /models
      |      |      |      |
      |      |      |      |--- task.ts // Tip: type Task
      |      |      |      |
      |      |      |      |--- App.tsx
      |      |      |      |
      |      |      |      |--- theme.ts
      |      |      |
      |      |      |--- // צבעים/spacing משותף (אופציונלי)

```

⚙️ הנחיות טכניות

TypeScript

- ```
type Task = { id: string; title: string; completed: boolean;
 { createdAt: number
```
- ממשקי Props לכל קומפוננטה, הקפדה על טיפוסים.

### State & Props

- state של רשימת המשימות יישב ב-`ToDoScreen` ויוזן לקומפוננטות כ-props.
- פעולות (`onSubmit`, `onToggle`, `onDelete`, `onChange`) עולות חזרה למסך.

### UI & StyleSheet

- עיצוב עם `StyleSheet.create`.
- שימוש ב-`Flexbox` (ברירת מחדל `column`), ריווחים עקביים.

- כפתורים: `Pressable` עם `hitSlop + pressed`.
- `FlatList` עם `{keyExtractor={ (t) => t.id }}` ו-`.contentContainerStyle`.

## נגישות

- `accessibilityRole="button"` לכפתורים.
- טקסט כפתורים ברור, ניגודיות מספקת.

## זרימת עבודה בסיסית (Pseudo):

- `useState<Task>([ ])`

## הוספה:

`((setTasks(prev => [{ id: uuid(), title, completed:false, createdAt: Date.now() }, ...prev`

•

## Toggle:

`(( setTasks(prev => prev.map(t => t.id===id ? {...t, completed: !t.completed} : t`

•

## Delete:

`(( setTasks(prev => prev.filter(t => t.id!==id`

•

- `(Filter: visible = tasks.filter(byCurrentFilter`

## קריטריוני קבלה (Acceptance Criteria)

- אפשר להוסיף משימה חדשה כשהטקסט לא ריק; לאחר הוספה השדה מתנקה.
- ניתן לסמן/לבטל השלמה לפריט.
- ניתן למחוק פריט.

- סינון מציג נכונה: All / Active / Completed.
  - הרשימה עובדת עם **FlatList** ללא גמגומים.
  - הקוד מחולק לקומפוננטות עם Prop Types/Interfaces.
  - עיצוב בסיסי, נקי וקריא.
- 

## עקרונות לפיתוח

- שימוש חוזר: **TaskItem**, **TaskInput**, **TaskFilterBar** ניתנים לשילוב במסכים אחרים.
  - קומפוזיציה: עטפו אזורים ב-**Card** לקבלת מראה אחיד.
  - פשטות: מינימום state דרוש, ללא לוגיקה מיותרת.
- 

## הרחבות (אופציונלי, אם נשאר זמן)

- עריכת משימה: לחיצה ארוכה תאפשר עריכה (**TextInput inline**).
  - ספירת משימות חיות/מושלמות בחלק התחתון.
  - שמירה מקומית (לשיעורים מתקדמים): **AsyncStorage**.
  - חיווי ריקנות: אם אין משימות – איור/טקסט "אין משימות".
  - בדיקות קלות: פונקציות pure ל-add/toggle/delete ב-models ניתנות לבדיקה.
- 

## זמן מוערך לביצוע

45–60 דקות (ליבה), +15 דקות הרחבות (רשות).

---

## המלצות

- התחילו בקומפוננטת **TaskItem** ו-**TaskInput**, ואז חברו ב-**ToDoScreen**.
- שמרו על שמות עקביים ל-styles (למשל **container**, **row**, **title**).