

מטלה 2 : Component Life Cycle

הנחיות כלליות

במטלה זו תתרגלו עבודה עם **Class Components** בלבד. המטרה היא ליצור רכיב המדמה קבלת נתונים בזמן אמת ("Logs") משרת, תוך ניהול נכון של מחזור החיים של הקומפוננטה (Lifecycle Methods) והצנת השימוש הנתונים בצורה דינמית.

הנחיות הגשה

- **אוף העבודה:** המטלה תבוצע בזוגות או בקבוצות הפרויקט שלכם.
- **חריגות:** הגשה ייחדנית מחייבת קבלת אישור מפורש ממנהל.
- **אוף ההגשה:** יש לאנד את כל קבצי הקוד הרלוונטיים לפתרון המטלה לקובץ קיז אחד.
- **העלאה:** יש להעלות את קובץ הקיז למערכת הגשת המטלות. (רק אחד/אחד לחברו הוצאות צירף להניש את הקובץ).

דרישות

1. ייצרת הקומפוננטה **ServerLog**

- .1.1 צרו קומפוננטת מחלקה (Class Component) בשם `ServerLog` (בשם `.ServerLog`)。
- .1.2 הגדרו את ה-state הראשון ב-`Constructor`. ה-state צריך להכיל מערך ריק בשם `logs`.

2. שימוש ב **componentDidMount**

- .2.1 כאשר הקומפוננטה נטענת ל-DOM, הפעילו טימר (`setInterval`) בטיעור (`val`)
- .2.2 שרצ **כל 2 שניות**.

- .2.2.1 בכל ריצה של הטימר, יש להוסיף לוג חדש למערך `logs` שב-state.
- .2.3 הלוג החדש צריך להיות אובייקט המכיל:
 - .2.3.1 pi: מספר רץ או חותמת זמן ייחודית.
 - .2.3.2 message: הודעה כללית (למשל: "OK")
 - .2.3.3 timestamp: השעה הנוכחיית.

3. ניקוי משאבים (componentWillUnmount)

- 3.1. חשוב מאוד למנוע דליפוט זיכרון. כאשר הקומפוננטה מוסרת מהמסך (Unmounting), וDAO שאתם עיצרים את הטימר שיצרתם בסעיף הקודם.

4. רינדור הרשימה (render & map)

- 4.1. בפונקציית `render`, השתמשו ב-`map.logs.map(this.state.logs)` כדי להציג את רשימת הלוגים על המסך.
- 4.2. עבור כל לוג, הציגו `vip` או זו המכיל את השעה ואת ההודעה.
- 4.3. חובה: להשתמש ב-`key` ייחודי לכל פריט ברשימה.

5. כפטור שליטה

- 5.1. הוסיף כפטור בקומפוננטת האב (למשל ב-`App`) שמאפשר להציג או להסתיר את קומפוננטת `ServerLog`. זאת כדי שתוכלו לבדוק שאכן הטימר נעצר כאשר הקומפוננטה יורדת מהמסך (אפשר לבדוק זאת באמצעות `console.log` בתוך `onInterval`).