אתגר המעלית

# **הסבר כללי**

* יש לתכנן ולבנות מערכת מעליות **יעילה** תוך שימוש בשפת TypeScript. במידה ואינך מכיר ניתן לכתוב גם ב-Javascript.
* לאחר טעינת העמוד בדפדפן יראה המשתמש את הבניין שיכלול את הקומות, המעליות ופקדי המעליות.
* בכל קומה יוצג פקד קריאה למעלית שעליו כתוב מספר הקומה. לחיצה על הכפתור תזמין מעלית לקומה (**גם אם אין אף מעלית פנויה כרגע**).
* **עם הזמנת מעלית לקומה, יוצג ליד כפתור המעלית מספר יורד המייצג את מספר השניות שנותרו עד להגעת המעלית.**
* **על אלגוריתם המעליות להביא לזמן ההמתנה המינימלי האפשרי למעלית מבלי להאריך את זמני ההמתנה של אלה שכבר הזמינו מעלית**.
* יש לאפשר הגדרה/שינוי של מספר הקומות ומספר המעליות בבניין באופן פשוט (באמצעות שינוי הגדרות בקוד או קובץ הגדרות).
* המעליות ינועו במהירות של **חצי שניה לקומה** (יש להציג את התנועה באנימציה חלקה – ולא למשל קפיצות בין הקומות), וכאשר יגיעו לקומת היעד (אליה הוזמנה המעלית), יתעכבו למשך שתי שניות.
* סרטוני הדגמה [נמצאים בתיקייה המצורפת](http://../../../../%D7%93%D7%95%D7%92%D7%9E%D7%901.mp4).

# **הגדרות ממשק משתמש**

* כל קומה תוצג עם רקע של לבנים. לצורך כך יש להשתמש בהגדרת **floor** שנמצאת ב- help.css.
* כל קומה תהייה בגובה כולל של **110** פיקסלים.
* בין כל שתי קומות יוצג פס שחור בעובי של **7** פיקסלים שיחושב כחלק מגובה הקומה שמתחתיו.
* המעלית תיוצג ע"י שימוש בתמונה נתונה (קובץ **elv.png**).
* בעת הגעת מעלית לקומה יש להשמיע קול נתון (קובץ **ding.mp3**).
* להצגת פקדי הקריאה למעלית יש להשתמש בתבנית הקוד הבאה:  
  <button class="metal linear">0</button>  
  הגדרות **metal** ו- **linear** נמצאות ב- help.css.
* עם הלחיצה עליו, יש לצבוע את טקסט פקד הקריאה למעלית בירוק ולהחזירו לצבעו הרגיל עם הגעת המעלית אליו.

# **הערות**

* **את קובצי העזר ניתן למצוא בתיקיה.**
* הפתרון צריך לכלול שימוש ב- best practices הרלוונטיים לפיתוח קליינט ובכלל.
* מומלץ להשתמש במתודולגיית OOP (כל אייטם יהיה אובייקט).
* יש לצרף מסמך קצר **באנגלית** המסביר על הארכיטקטורה בפרויקט, והסבר על האלגוריתם הראשי שפותח עבור הפרויקט.
* **יש להעלות את הפרויקט לGitHub, כולל מסמך האפיון, ולהגיש לינק לפרויקט.**

# **דוגמה לפריסת ממשק הבניין והמעליות:**

