**ארכיטקטורת תוכנה – פרויקט  
  
מגישים**:  
ניקיטה קונובלנקו 321237406

כפיר כהן 316112176

גיא זמיר 318471216

שני פחימה 319006698

אביתר סלומון 318744976

יאיר זנבה 316350933

**הסבר על אלמנטים שלא נלמדו בכיתה:**

**Html & CSS:**

1. שימוש בתווית tbody – בשורה <tbody id="tableBody"></tbody>: המשמעות של תווית tbody היא לאגד את התוכן של טקסט או רכיב מסויים בתוך טבלת HTML.
2. שימוש בטקסט &copy – מציג את הסימן ©.
3. שימוש בתווית colspan ו – rowspan – לאחד מספר עמודות\שורות לשורה\עמודה אחת לדוגמא:
4. <th colspan="3"><h2>Reservation Form</h2></th> הפכנו את הtable header- של טבלת ההזמנות לשורה אחד שמתפרשת על 3 עמודות לשם נראות וסדר בטבלה עצמה.
5. שימוש בתווית <img alt = “”> אפשרות להוסיף כינוי אלטרנטיבי לתמונה שאותה אנחנו רוצים להוסיף, בנוסף לכך, בעת ואם לא תוצג התמונה יוצג הכינויי וכך נדע איזו תמונה לא נראית.
6. שימוש בתווית footer – מגדיר כותרת תחתונה למסמך או לקטע.
7. שימוש בתווית class – תכונת class משמשת לציון מחלקה על מנת לאגד רכיבים שונים כך שיהיו שייכים לאותו הגוף\מחלקה\כינויי (שימושי עבור שינויים עתידיים בחלק מהרכיבים בלי לשנות במקומות נוספים).
8. שימוש ב font-family ב CSS: מגדיר את גופן הטקסט.
9. שימוש ב margin ב CSS: מגדיר את הרווח שנוצר מסביב לרכיב.
10. שימוש ב border-style ב CSS: מגדיר את סגנון ארבעת הגבולות של רכיב.
11. שימוש ב border-collapse ב CSS: קובע אם גבולות הטבלה צריכים להתכווץ לגבול בודד או להיות מופרדים.
12. שימוש בהגדרה של CLASS ב CSS: עד עכשיו למדנו איך לתת שם ייחודי לרכיב וב CSS לשנות לו את המאפיינים, לעומת זאת ה CLASS מגדיר מחלקה שנותנת את אותם המאפיינים לרכיבים שונים שאנחנו בוחרים להכנס למחלקה ומגדירים אותם ב CSS על ידי נקודה לעומת סולמית שתמיד היינו מגדירים לשמות יחודיים.
13. שימוש ב border ב CSS: מגדיר את עובי הגבול וצבעו.
14. שימוש ב display ב CSS: מציין את האופן בו מוצג רכיב בדף.
15. שימוש ב align-items ב CSS: מגדיר את היישור עבור רכיבים.
16. שימוש ב justify-content ב CSS: מיישר את הרכיבים כאשר הרכיבים אינם מנצלים את כל השטח הפנוי בציר הראשי, האופקי.
17. שימוש ב padding ב CSS: יוצר מרחב סביב התוכן של הרכיב, בתוך כל סוגי הגבולות המוגדרים.
18. שימוש ב border-radius ב CSS: מגדיר את הרדיוס של פינות הרכיב, כלומר מאפשר לעגל פינות.
19. שימוש ב cursor ב CSS: מציין את סמן העכבר שיוצג בעת הצבעה על רכיב.
20. שימוש ב position ב CSS: מציין את סוג שיטת המיקום המשמשת עבור רכיב (סטטי, יחסי, קבוע, מוחלט או דביק).
21. שימוש ב bottom ב CSS: משפיע על המיקום האנכי של רכיב ממוקם. Bottom לא משפיע על רכיבים לא ממוקמים.
22. שימוש ב box-shadow ב CSS: יוצר צל לרכיב לפי עובי מבוקש וצבע.
23. שימוש במספר קבצי CSS אשר מתחלפים על ידי לחיצה על כפתור Light/Dark mode אשר מפעיל פונקציה בשכבה השנייה.

**JavaScript:**

**שכבה 2:**

* addEventListener('DOMContentLoaded', function()

בזמן ביצוע רענון מפעילה את הפונקציות בתוכה אורך מחרוזת length.

* פונקציה שבודקת אם ה – checkbox בכניסה הראשית מוסמן ומחליט לגבי סוג ההקלדה (מוסתר סוג password או גלויי מסוג text).
* פונקציה המשנה את תוכנית הCSS שמופעלת כעת בדף על ידי לחיצה מהמשתמש על כפתור Light/Dark mode.
* שימוש בפוקנציית split – הפונקציה split מבצעת הכנסה של סטריניגים לתוך המערך המופרדים בסטרינג ארוך על ידי התו שהיא מקבלת. בדוגמא הזו, התו הוא '|' וה – split מבצעת הכנסה של המחרוזות לתוך המערך studentData. דוגמה לשימוש: var studentData = studentInfo.split('|');.
* שימוש בפוקנציית push – הפונקציה push מקבלת מערך חד מימדי studentData ומוסיפה אותו למערך הדו מימדי students כאיבר במערך. דוגמה לשימוש: students.push(studentData);.

**שכבה 3:** switch - ביצוע פעולה על בסיס משתנה שהפונקציה מקבלת.

**שכבה 4:** let – סוג משתנה גלובלי, class שימוש במחלקות עם תבנית עיצוב singelton, throw new Error – שולח הודעה בConsole, this. – פונה לאוייבקט הנוכחי\למתודות שקיימות בתוכו, parseInt - ממירה מחרוזות למספר, isNaN - אם המשתנה הוא מספר – TRUE אחרת FALSE , const - סוג משנה קבוע – השם שמור.

**נושא נבחר: Keyboard events**

אירועי מקלדת מתייחסים לאינטראקציות המופעלות על ידי לחיצה או שחרור של מקשים במקלדת המחשב.  
מתי כדאי להשתמש בו?

טיפול בקלט משתמש: אירועי מקלדת חיוניים לטיפול בקלט משתמש, בעיקר כאשר משתמשים צריכים ליצור אינטראקציה עם האתר באמצעות המקלדת.

נגישות: יישום אירועי מקלדת משפר את הנגישות, ומאפשר למשתמשים שעלולים להתקשות עם עכבר לנווט ולקיים אינטראקציה יעילה.

קיצורי דרך ומקשי קיצור: שימושי ליצירת קיצורי מקלדת או מקשי קיצור כדי לייעל את אינטראקציות המשתמש עם האפליקציה.

איך אתה משתמש בו:

על ידי קישור מקש או קבוצת מקשים לפונקציה מהשכבה השנייה, הפונקציות יופעלו על סמך המקשים שנבחרו, שילוב אירועי מקלדת עשוי להיות חיוני עבור תכונות כמו שליחת טפסים, ניווט או יישום קיצורי דרך לפעולות ספציפיות.

מדוע בחרת בנושא זה:

חווית משתמש משופרת: יישום אירועי מקלדת משפר את חווית המשתמש הכוללת על ידי מתן דרכים מרובות לאינטראקציה עם האפליקציה.

פונקציונליות ויעילות: קיצורי מקלדת ואירועים יכולים להגביר משמעותית את היעילות של משתמשים באינטראקציה עם המערכת, במיוחד עבור משימות שחוזרות על עצמן.

**שימוש בפרויקט:**

* when press ALT+ R /(‘ר’) - cleanForm - Keyboard event.
* when press Enter- submit reserve Keyboard event.
* When press control + Q /(‘/’) - changeMode from dark to light and vice versa - Keyboard event.

**התייחסות למשובים- סטודיו**

|  |  |  |  |
| --- | --- | --- | --- |
| **מסד** | **הערת משוב** | **האם התבצע שינוי באפליקציה בעקבות ההערה** | **נימוק** |
| 1 | חזות חיצונית | כן | עיצוב האתר בכניסה הראשית שונה ושופר |
| 2 | עיצוב\עיצוב האתר | כן | נוספה אפשרות להחשכת דף האתר |
| 3 | הוספה של כמות חניות בכל חניון\אם החניון מלא או לא\להוסיף אם חניה פנויה | כן | נוסף בתצוגה לסטודנט בעת הזמנת חניה |
| 4 | לשנות מיקום של אלמנטים כדי שיהיה יותר קריא | כן | שונו מיקומי האלמנטים המרכזיים באתר |
| 5 | לסדר את האתר מבחינת גדלים שיהיה נוח לעין ולמרכז את האתר | כן | גדלים שונו בהתאם לדעת הרוב בקבוצה |
| 6 | העיצוב מרוכז לחלק קטן מהדף ולא מתפרש על כולו­ | כן | הותאם לפרישה כוללת על כל הדף לתצוגה נעימה לעין ושימוש נוח לסטודנט |
| 7 | במקום שיופיע בטופס עדיף שגישת מנהל תהיה דף משלו | כן | נפתח דף מקושר ל"LOGIN" בו המשתמש בוחר להיכנס לדף סטודנט להזמנת חניה או לדף מנהל לניהול החניות |
| 8 | נראות ויזואלית פחות מונוטונית\ פונט, תמונת רקע | כן | התווספו צבעים רבים ותמונות רקע שמשתנות ממצב מואר לחשוך ולהיפך |
| 9 | הצבע של הכפתורים | כן | שונו בהתאם לכל דף באתר, דף מנהל, דף סטודנט ודף כניסה ראשי, בנוסף לכך התווסף כפתור החשכה והארה |
| 10 | המערכת לא מאוד קלה לשימוש ולא תמיד היה ברור מה לעשות | כן | נפתח דף ראשי להכוונת המשתמש, ובכל עמוד מפורט מה עליו למלא בתבניות |
| 11 | הוספת בדיקת קלט לאלגוריתם ת"ז תקין, שמירת נתונים בדף נפרד | כן | נוסף אלגוריתם לבדיקת קלט ת"ז שהזמין כבר חניה, בנוסף לכך נוסף אלגוריתם לבדיקת חניה שהוזמנה לפי מספר רכב והנתונים יוצגו אך ורק למנהל |

**התייחסות למשובים – הצגת פרויקט**

|  |  |  |  |
| --- | --- | --- | --- |
| **מסד** | **הערת משוב** | **האם התבצע שינוי באפליקציה בעקבות ההערה** | **נימוק** |
| 1 | עיצוב דף הזמנה | כן | עיצוב דף ההזמנה שונה ושופר |

**תבנית עיצוב:singleton**

תבנית העיצוב singleton מבטיחה שלמחלקה יהיה מופע אחד בלבד ומספקת גישה גלובלית לאותו מופע.

מדוע בחרתם בתבנית עיצוב זו:

כיוון שיש לנו בפרויקט רק מחלקה אחת אין צורך להתאים אותה לממשק או מחלקה אחרת – לכן Adapte לא רלוונטי, ואין צורך לייצר אובייקט בזמן הריצה לכןPrototype גם לא רלוונטי. רצינו לוודא שלמחלקת CounterManeger יהיה מופע אחד בלבד על מנת כדי לוודא שאין חפיפה בין משתנים ובמידה וכבר קיים האובייקט התוכנית תעבוד עליו.

כיצד ממשתם אותה באתר:

מימשנו אותה בJavaScript כאשר יצרנו מחלקת CounterManeger אשר מכילה 4 מונים ו2 מתודות: 1.עדכון מונה ספציפי. 2. קריאת הערך במונה ספציפי.  
המחלקה מכילה constructor אשר בודק האם כבר קיים מופע של המחלקה – במידה וכן שולח ההערה, אחרת - יוצר אובייקט מסוג המחלקה ומאתחל אותן.

פונקציית אתחול אשר מגדירה 4 משתנים שנשמרים localstorage ומאתחלת ל5 כל חניה.  
כאשר בתחילת התוכנית מופיע שורה שיוצרת אובייקט חדש במידה ולא קיים.  
const counterManager = new CounterManager();

**תכנות מקבילי:**

בחרנו להשתמש בתכנות מקבילי זה כדי לדמות איך שני תהליכים שקורים יכולים לקרות בו זמנית ולמעשה לחסוך זמן ריצה.

מימשנו בעזרת בJavaScript  כאשר יצרנו מאזין לטופס אשר בכל רגע נתון לוקח מידע רלוונטי מה-Student ID ומה-Car ID, לאחר מכאן אנחנו בודקים בתנאי האם המידע שקלטנו הוא לא ריק ואין בו רק מספרים. אם כן מוחק באמצעות slice את התו האחרון ומציג הודעת שגיאה מתאימה.

אתגר משמעותי שהיה לנו היה בעצם ללמוד והלבין שצרי להשתמש במאזין לטופס ורק כך נוכל בכל רגע נתון לשלוף את המידע שהסטודנט יכניס לאתר.

**אתגרים במהלך העבודה ופתרונות:**

1. חוסר ידע במימוש פונקציות מסוימות באתר – למדו את הנושאים מהאינטרנט באתרים כמו freeCodeCamp W3School ובמהלך המימוש נעזרנו בChatGPT על מנת למצוא טעויות או באגים בקוד.
2. קושי בהתאמת זמנים לביצוע המטלות – נפגשנו בZOOM כולם לפני העבודה, הבנו ביחד את המטלה וחילקנו עבודה לצמדים, דנו בממשקי עבודה בין הצוותים ונתנו להם פתרונות ביחד. לאחר שכל הצוותים השלימו את משימתם נפגשו שוב בZOOM ועברנו יחד על מנת לוודא שכל התוכניות רצות יחד בצורה טובה.
3. קושי בסנכרון המטלה וביצוע עדכונים – כיוון שהפרויקט בנו מ3 חלקים שנשענים אחד על השני כל עדכון שלך אחד החלקים דורש התאמה וסנכרון של השאר, לכן על כל שינוי היה צריך להודיע לכולם.

הפתרון היה שחילקנו את עדכון המטלה לימים כאשר שכבת התצוגה מעדכנת ראשונה, לוגיקה שנייה, ומסד הנתונים אחרונה.

הערות:

יש להכנס לאתר דרך דך html login ממנו אפשר להגיע לשאר הדפים.

קיצורי המקלדת מיועדים רק לדף student reservation