***מטלת בית -דיסקונט***

**מבוא:**

המטלה באה להציג נתונים שהתקבלו מהלקוח על תוצאות של חניכים.   
הוספת נתונים עבור חניך חדש, עדכון פרטים של חניך קיים והוספת תוצאות בחינות אחרות.  
תרשימי מידע וסיכום על החניך.

במימוש שלי השתדלתי להשים לב על רסופנסיביות האתר.  
לפי המוניטור (ולא מכשיר פיזי P: )

**טכנולוגיות בשימוש:**

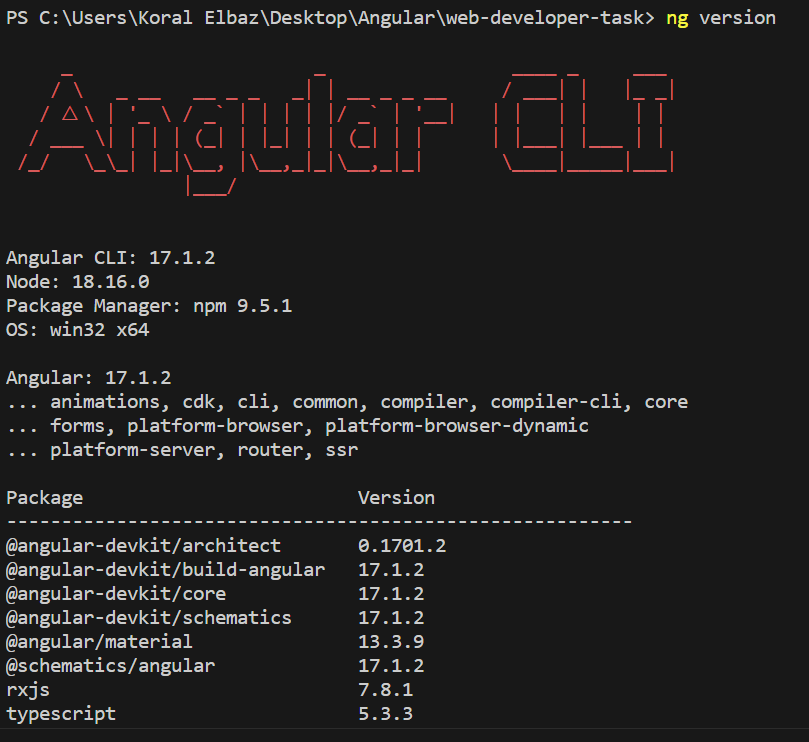
הטכנולוגיות והספריות שהשתמשתי בהם הם:

Rxjs – הספרייה הכי שימושית

Ag-grid – עבור הgrid עם נתוני החניכים.

Primeng - dialogs – חלוני פופ-אפ קופצים.

chart.js – תרשמים שמוצגים בדף הAnalsis.

**גרסאות:**  


תמונה שמכילה טקסט, צילום מסך, גופן, עיצוב

התיאור נוצר באופן אוטומטי**מבנה הפרויקט:**  
המבנה הוא כזה:

Modules

יצרתי 3 מודלים חדשים, כל מודל עבור עמוד.

כלומר יש לנו מודל עבוד עמוד הDATA , מודל עבור עמוד הANALYSIS, מודל עבור עמוד Monitor.

Routing

אז ישנו ה Routingהראשי:  
  
כאן נעשה שימוש ב Lazy loading.

Lazy loading היא טכניקה המשמשת בפיתוח תוכנה, כולל Angular, כדי לשפר את ביצועי האפליקציה על ידי טעינת מודולים או רכיבים רק כאשר הם נחוצים, במקום טעינת הכל מראש.

  
ועבור כל מודל הגדרתי Routing לרכיב הראשי שלו.

Services  
עבור כל מודל הוגדר השירות הנחוץ לו.  
נמנעתי מלעשות קובץ אחד ארוך ופיצלתי ל3 קבצי שירותים.  
  
  
הדמתי שרת נתונים.  
חיקתי תקשורת עם שרת נתונים מרוחק באמצעות In-memory Web API module.

**רכיבים:**

הרכיב הראשון הוא הרכיב המרכזי שמציג את שאר הרכיבים.   
הוא היחידי במודל שיש לו אינטראקציה עם ה Services.  
שאר הרכיבים הם רכיבים "טיפשים" מציגים נתונים / עושים פעולות "פשוטות".  
  
הרכיב הראשי במודל DATA הוא DataComponent.  
הרכיב הראשי במודל ה ANALYSIS הוא ChartDisplayComponent   
הרכיב הראשי במודל הMONITOR הוא MonitorDataComponent

רכיבים עבור מודל הDATA

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי

רכיבים עבור מודל הANALYSIS

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי  
  
רכיבים עבור מודל הMONITOR

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי

התקנה:   
npm I – עבור התקנה של כל מה שקיים בpackage.json  
npm start – הרצת התוכנית.

**אתגרים ומסקנות:**  
  
מבחינת אופטימיזציה של ביצועים:  
היה חשוב לי לשמוע על ביצועים טובים.  
לאחר חקר לא קטן ראיתי שמה שעונה על צרכי זה ה lazy loading.  
בנוסף, יש רכיב שהייתי עושה אותו גנרי ששני מודלים יוכלו להשתמש בו.  
זה הסינון לפי הID.

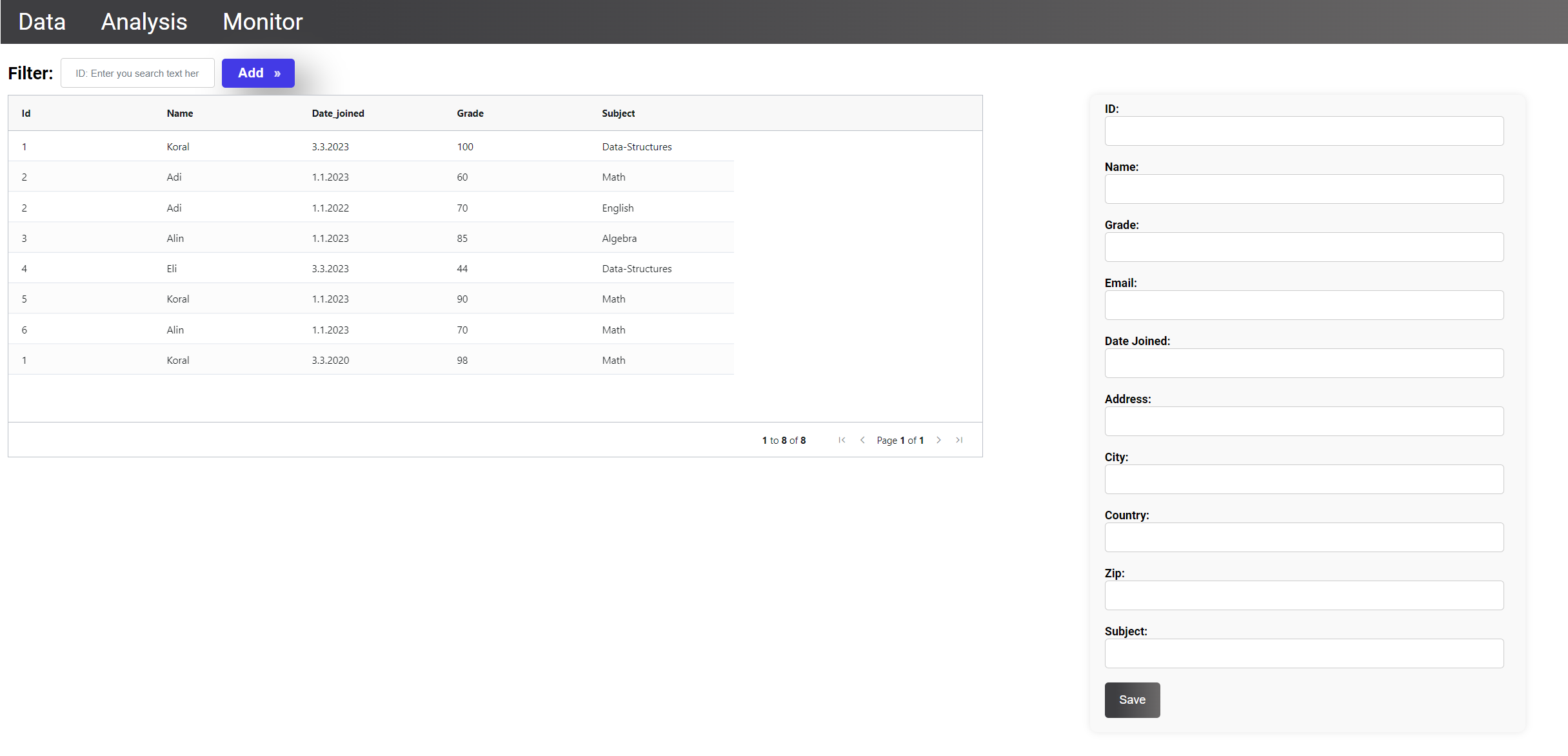
שיפורים בחוויית משתמש (UX):  
בפרויקט שלי לא הוספתי ולידציה בכמה מקומות.  
דבר חשוב ונוח למשתמש, אין ספק שהוסיף.  
  
בדיקה וניפוי באגים:  
פה מתקשר הסניף הקודם, חשוב פה עיניין הולידציה אך בהנחה והמשתמש מזין פרטים תקינים לפי הסבר הToolTips שעשיתי הכל אמור לעבוד כהלכה.  
גם כאן הוסיף בהמשך בדיקות יחידה.

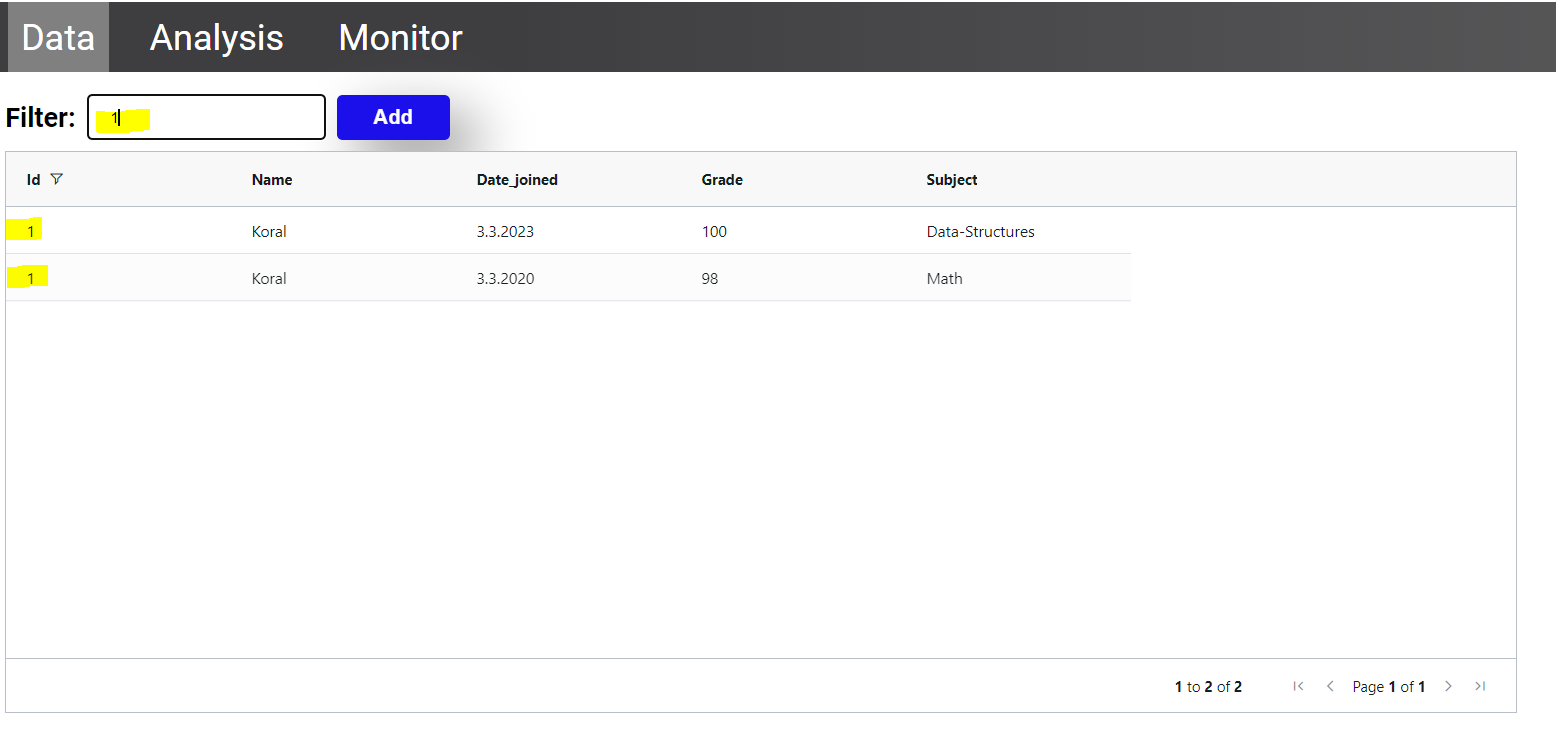
**לסיכום:**  
משתמשים יכולים לנווט בצורה חלקה באפליקציה ולבצע משימות בקלות.  
השתדלתי ליישם ביעילות את התכונות שנדרשו וזה כולל את הLazy Loding.  
  
במבט קדימה, מספר תחומים מצדיקים שיפור נוסף:  
ולידציה לInputs, בדיקות יחידה ותמיד יש מה להוסיף בUI 😊  
  
לסיכום, פרויקט זה היה מאמץ מתגמל, המציג את העוצמה והרבגוניות של Angular.

**צילומי מסך:**  
  
*מודל הDATA:*

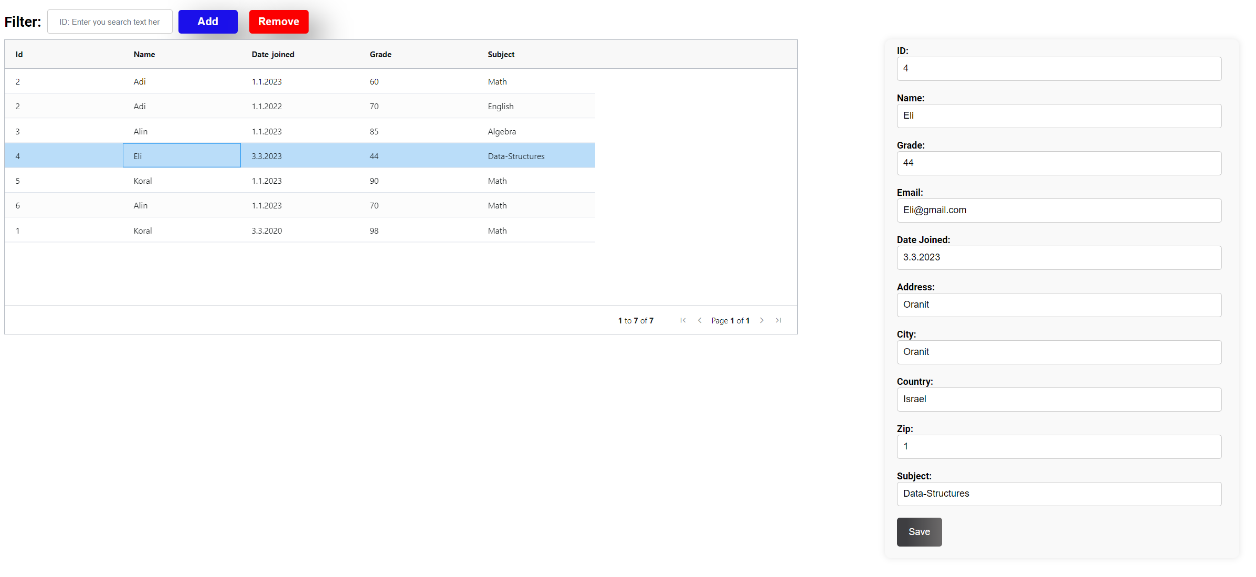
בעת לחיצה על כפתור Add:

נפתח רכיב ה פרטים.



כאשר נבצע חיפוש בשורת הסינון נשים לב שהטבלה תתעדכן בהתאם.  
בנוסף כאשר נעמוד על שורת הסינון יפתח tooltip הסבר.  
בחלק עשיתי בstate management כלומר כאשר נעבור לדף אחר ונחזור נראה שהזין מוזן הערך שהזנו קודם.  


כאשר נבחר שורה נראה כי רכיב הפרטים התמלא בפרטי החניך ובנוסף יוצג לנו כפתור מחיקה.

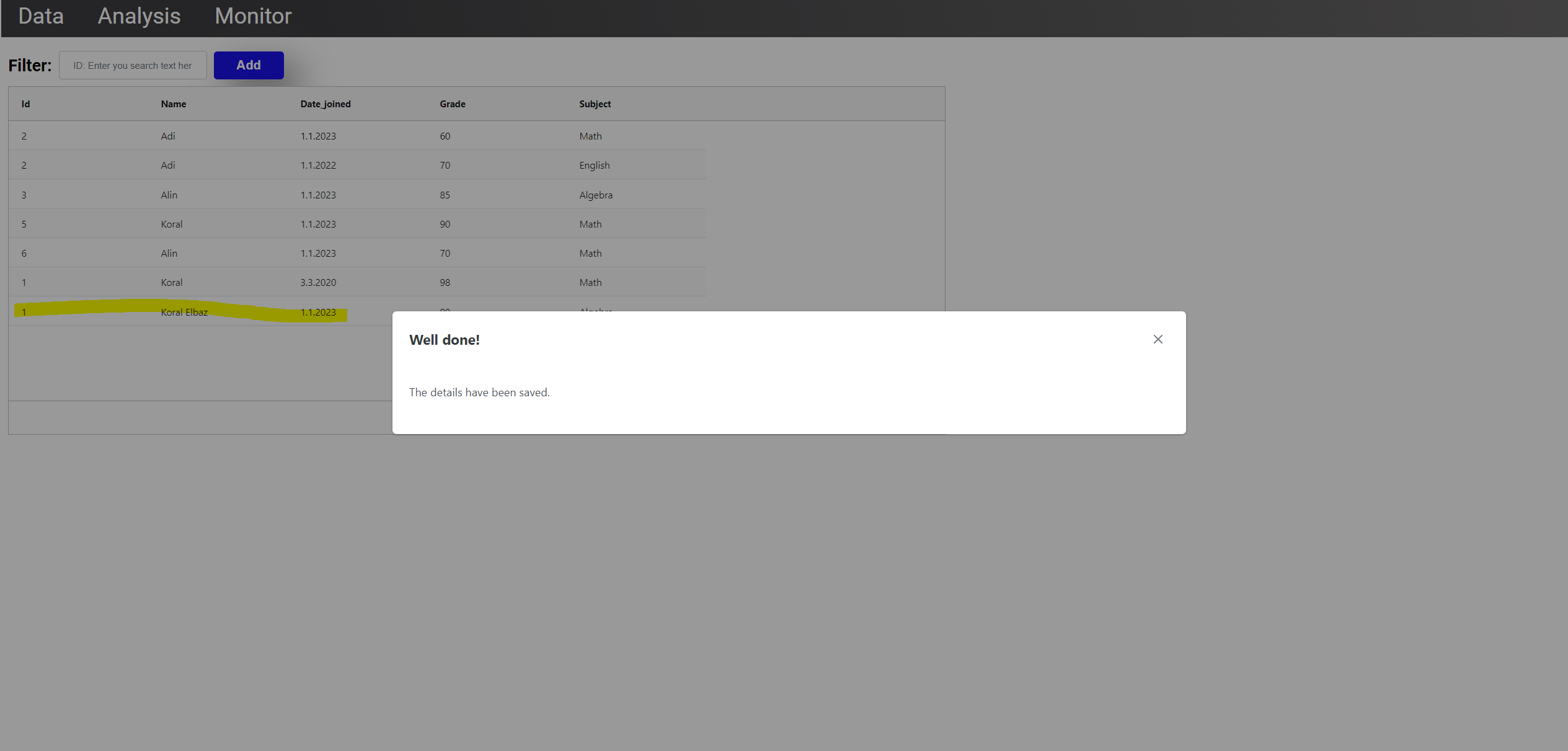


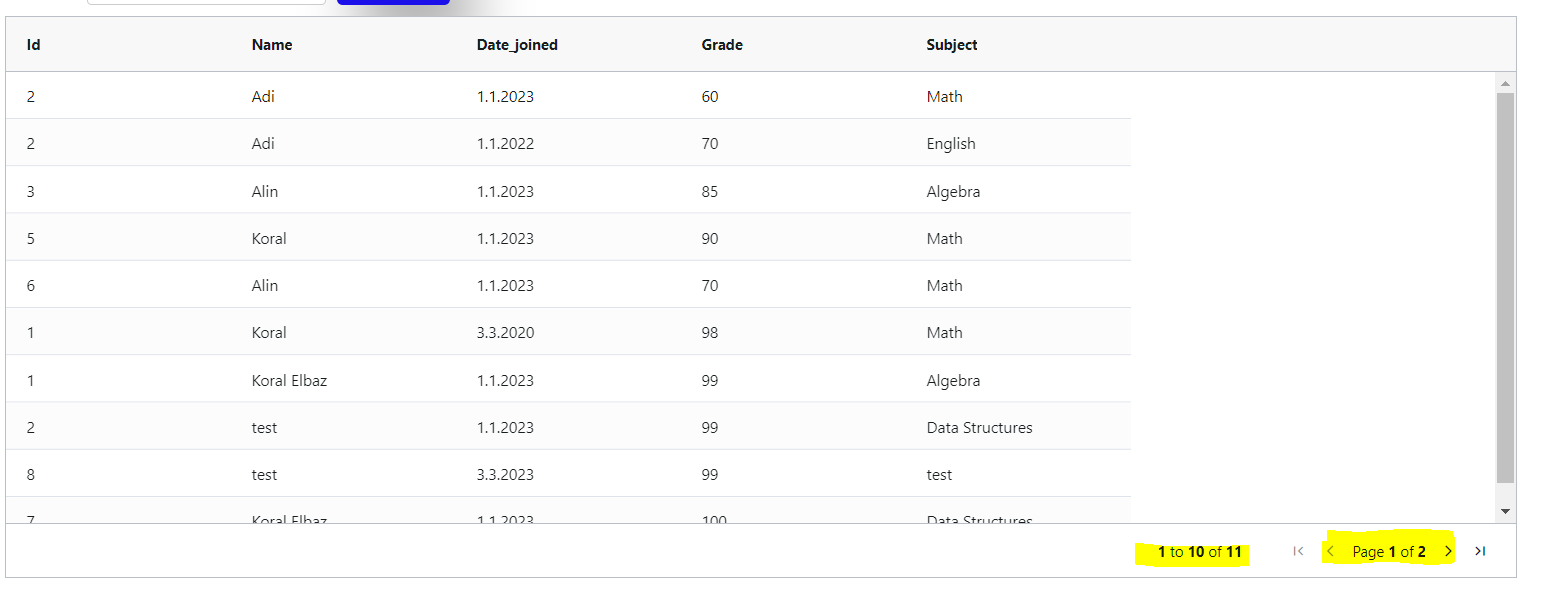
נלחץ על המחיקה נראה ש:

חלונית הפרטים נסגרה, הרשומה נמחקה וגם קופץ חלון שמודיע שהפעולה אכן הסתיימה!  
  
תמונה שמכילה צילום מסך, טקסט, תוכנה, תכונות מולטימדיה

התיאור נוצר באופן אוטומטי

הוספת רשומה חדשה:

לחצתי על כפתור הוספה מילאתי פרטים קובץ חלון סיום פעולה ומאחוריו ניתן לראות את הפרטים שהוספתי.  


נוסיף עוד כמה חניכים ונראה כי אכן יש הגבלה ל10 בדף ואחרי שיש 10 נדפדף לדף אחר בטבלה.  
  
תמונה שמכילה תוכנה, טקסט, סמל מחשב, צילום מסך

התיאור נוצר באופן אוטומטי

*מודל הANALYSIS*  
מסך ראשי:

תמונה שמכילה טקסט, חשמל, צילום מסך, תצוגה

התיאור נוצר באופן אוטומטי

לאחר הזנת הסינון ניתן לראות שהתרשימים מתמלאים בהתאם לדרישות:

תמונה שמכילה טקסט, צילום מסך, תוכנה, תצוגה

התיאור נוצר באופן אוטומטי

בנוסף ניתן לראות שיש להם את היכולת של Drag & Drop שהתבקשנו:

(שימו לב לסדר מהתמונה הקודמת)  
  
תמונה שמכילה טקסט, צילום מסך, תוכנה, תרשים

התיאור נוצר באופן אוטומטי *מודל ה MONITOR*כברירת מחדל הלחצים לחוצים ומראים לפי צבע איזה ממוצע נכשל (פחות מ65)  
  
  
תמונה שמכילה צילום מסך, טקסט, תוכנה, סמל מחשב

התיאור נוצר באופן אוטומטי  
  
כאשר נלחץ על הכפתור "נכשל"

תמונה שמכילה צילום מסך, טקסט, תוכנה, סמל מחשב

התיאור נוצר באופן אוטומטי  
כאשר נלחץ על כפתור "עבר"

תמונה שמכילה טקסט, צילום מסך, קו, מספר

התיאור נוצר באופן אוטומטי

סינון:

תמונה שמכילה צילום מסך, תוכנה, טקסט, תכונות מולטימדיה

התיאור נוצר באופן אוטומטי

סינון לפי שם וכפתור לחוץ:

תמונה שמכילה צילום מסך, טקסט, תוכנה

התיאור נוצר באופן אוטומטי

סינון לפי שם וID

תמונה שמכילה צילום מסך, טקסט, תוכנה

התיאור נוצר באופן אוטומטי

*רכב – דף לא נמצא:*

בשורת הURL נכתוב דבר שלא נמצא בניתוב:



תמונה שמכילה טקסט, צילום מסך, גופן, עיצוב

התיאור נוצר באופן אוטומטי

*תמונות המעידות על רספונסיביות*

בלחיצה על השורה או לחצן הוספה הרכיב פרטים יפתח למטה:תמונה שמכילה טקסט, צילום מסך, מספר, גופן

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, מספר, תוכנה

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי