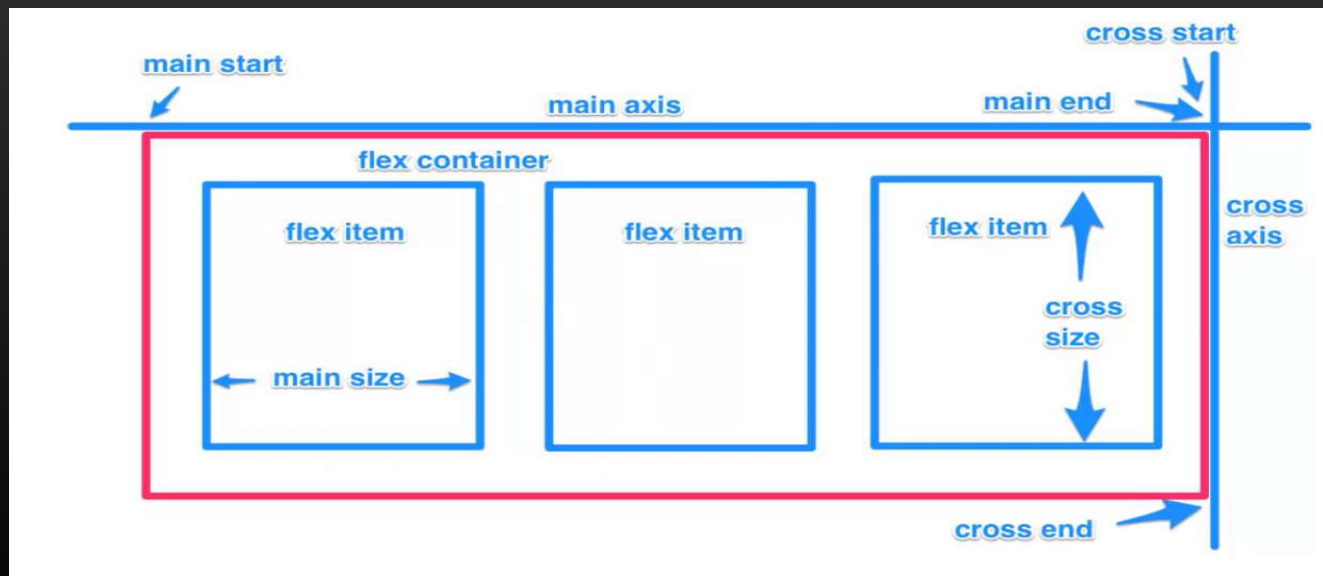


Flex

Css Flex

מהו Flexbox?

עימוד תוכן באמצעות פריסת אלמנטים (item) בתוך מיכל (container)



היתרונות של שימוש ב - Flexbox?

- מודולריות בבניית העמוד – בנייה חד ממדית
- קוד נקי וברור
- רספונסיביות
- מקל עלינו בקביעת רווחים בין האלמנטים מבלי לפגוע בתוכן האלמנט.
- אפשרות לשינוי הסדר בין האלמנטים בהתאם לשינוי בגודל מסך

מטרות השיעור




- הסטודנט ידע לעצב עמוד בעזרת Flexbox
- הסטודנט ידע לעצב עמוד בעזרת ספריית bootstrap
- הסטודנט יבין את היתרונות והחסרונות בשימוש ב - Grid


הנושאים שילמדו בשיעור



- Flex Properties
 - Display
 - Flex-direction
 - Flex
 - Flex-grow
 - Flex-shrink
 - Flex-wrap
 - Justify-content
 - Align-items
 - Align-self
 - Align-content
- Bootstrap flex
 - d-flex
 - row | column
 - col
 - Justify-content
 - Align-items
 - Navbar
 - card
- Grid Properties


הכנת הקוד לעבודה עם Flexbox

- א- ניצור תיקייה חדשה לשיעור בשם css-flexbox
- ב- נפתח את ה – vscode בתיקייה וניצור את הקובץ Flex-sandbox.html
- ג- בעזרת הפקודה ! + Shift ניצור עימוד HTML בסיסי
- ד- נחליף את כותרת העמוד ל – Flex Sandbox
- ה- בעזרת הפקודה הבאה ניצור את המיכל והאלמנטים שדרושים לנו להדגמה:
`.container>(.box.one>h3{box-1})+(.box.two>h3{box-2})+.box.three>h3{box-3}`
- ו- ניצור תגית style ב – head ובתוכה תגיות עיצוב של:

```
.container {  
  background:  #999;  
  border: 1px solid  black;  
  color:  blanchedalmond;  
  height: 500px;  
  padding: 10px;  
}
```

```
.box {  
  height: 100px;  
  border: 1px solid  black;  
  min-width: 250px;  
  padding-left: 20px;  
}
```

```
.one {  
  background:  #900;  
}  
.two {  
  background:  #090;  
}
```

```
.three {  
  background:  #009;  
}
```

Display

קביעת מטודת הפריסה של האלמנטים העיצוביים בדף.
האלמנט העיצובי הוא ברמת המיכל ולא ברמת האלמנטים

אפיון האלמנט העיצובי בברירת המחדל: `display: block;`

טבלה של הערכים העיקריים:		
1	Inline	יסדר את האלמנטים באותה שורה
2	Block	יסדר כל אלמנט מתחת לאלמנט הקודם
3	Flex	יסדר את האלמנטים בהתאם לפרמטרים שנגדיר לו. התייחסות למיכל עם שורה אחת שבה יימצאו האלמנטים.
4	Grid	יסדר את האלמנטים בהתאם לפרמטרים שנגדיר לו. התייחסות למיכל עם מס' שורות.
5	None	לא יציג את האלמנט
6	Table	סידור האלמנטים בתוך טבלה

קישור לתצוגה ויזואלית של ההבדלים ב – W3School

https://www.w3schools.com/cssref/playit.asp?filename=playcss_display&preval=inline

Flex-direction

כיווניות סידור האלמנטים בתוך המיכל

`flex-direction: row;` אפיון האלמנט העיצובי :

טבלה של הערכים העיקריים:		
1	column	יסדר את האלמנטים בתוך המיכל בסדר יורד מלמעלה למטה
2	column-reverse	יסדר את האלמנטים בתוך המיכל בסדר עולה מלמטה למעלה
3	row	יסדר את האלמנטים זה לצד זה משמאל לימין
4	row-reverse	יסדר את האלמנטים זה לצד זה מימין לשמאל

קישור לתצוגה ויזואלית של ההבדלים ב – W3School

https://www.w3schools.com/cssref/playit.asp?filename=playcss_flex-direction&preval=row

Align-items

סידור האלמנטים בצורה אנכית במיכל

`align-items: stretch;` אפיון האלמנט העיצובי :

✓ יש לשים בהערה את גובה ה – `box` על מנת לראות את האפקט בצורה מיטבית

טבלה של הערכים העיקריים:		
1	Flex-start baseline	יקבע את האלמנטים לראש המיכל
2	Flex-end	יקבע את האלמנטים בתחתית המיכל
3	Stretch	יקבע את האלמנטים לכל אורך המיכל
4	Center	יקבע את האלמנטים במרכז המיכל

קישור לתצוגה ויזואלית של ההבדלים ב – W3School

https://www.w3schools.com/cssref/playit.asp?filename=playcss_flex-direction&preval=row

Flex-wrap

קביעה עם תהיה ירידת שורה כשמגיעים לרוחב המינימאלי של האלמנט
עוזר לנו להפוך את האתר לרספונסיבי

אפיון האלמנט העיצובי : flex-wrap: nowrap ;

טבלה של הערכים העיקריים:		
1	nowrap	לא יוריד את האלמנט שורה במקרה כשיגיע ל - (min-height) אלא ירחיב את הדף
2	Wrap	יוריד שורה ברגע שהאלמנט יגיע ל - (min-height) ולא לשבור את הדף.
3	Wrpa-reverse	יהפוך את סדר האלמנטים ויוריד שורה במקרה הצורך כדי לא לשבור את הדף

קישור לתצוגה ויזואלית של ההבדלים ב – W3School

https://www.w3schools.com/cssref/playit.asp?filename=playcss_flex-wrap&preval=nowrap

Justify-content

קביעת המרווחים בין האלמנטים במיכל

אפיון האלמנט העיצובי : justify-content: flex-start;

✓ יש לשים בהערה את ה – padding ב – container טרם התצוגה

טבלה של הערכים העיקריים:		
1	Flex-start	מצמיד את האלמנטים לצד שמאל כשהכיוונית היא לצד שמאל
2	Flex-end	מצמיד את האלמנטים לצד ימין כשהכיוונית היא לצד שמאל
3	Center	מצמיד את האלמנטים לאמצע
4	Space-between	מצמיד את האלמנט הראשון לצד שמאל, האלמנט האחרון לצד ימין ואת כל השאר מפזר ברווחים שווים
5	Space-around	קובע רווח קבוע לכל אלמנט בין אלמנט לאלמנט
6	Space-evenly	קובע את אותו רווח בין כל האלמנטים וההתחלה והסוף

קישור לתצוגה ויזואלית של ההבדלים ב – W3School

https://www.w3schools.com/cssref/playit.asp?filename=playcss_justify-content&preval=initial

סיכום ביניים

שאלות ביניים:

מהו האלמנט העיצובי Flex ולמה הוא משמש?

על מה המאפיין display משפיע ומה הסוגים העיקריים שלו?

לאיזה המאפיין row מקושר ואיזה עוד סוגים של ערכי המאפיין קיימים?

על מה המאפיין align-items משפיע ומה הם סוגי הערכים העיקריים שלו?

על מה המאפיין flex-wrap משפיע ומה ערך ברירת המחדל שלו?

על מה המאפיין justify-content משפיע ואיך אני מפזר את האלמנטים באופן שווה במיכל?

משימת ביניים:

ליצור תפריט ניווט בצבע אפור כהה, שהטקט יהיה בגוון של לבן, בצידו השמאלי יהיה

לוגו, בצידו הימני יהיו תפריטי הניווט: Home | About | Contact

משימת ביניים:

- יש ליצור תיקייה חדשה בשם Flex Practice
- יש ליצור קובץ חדש בתוך התיקייה שיצרנו בשם flex-practice.html
- ליצור תפריט ניווט בגובה 50px בצבע אפור כהה, שהטקט בתוכו יהיה בגוון של לבן, פונט אריאל, שהאלמנטים בתוכו יהיו מסודרים בציר האנכי במרכז, עם padding של 10px. שהתפריט יתפוס את כל אורך השורה ולא יהיו נימות לבנות בצדדים.
- בציודו השמאלי יהיה לוגו, בציודו הימני יהיו תפריטי הניווט: Home | About | Contact
- שלמיכל תפריטי הניווט הימני יהיה מינימום רוחב של 200px
- קישורי הניווט בצד הימני: שקישור אחד יהיה מוצמד לצד השמאלי אחר לצד הימני והשלישי באמצע, שהם יהיו באותו צבע של הלוגו ובלי קו מתחת, שהם יהיו מסודרים אחד לצד השני, שיהיו במרכז הציר האנכי, ושהקישורים יובילו לאותו הדף.

Flex

קביעת המקום שהאלמנט יתפוס ביחס לאלמנטים האחרים במיכל

* מאפיין ברמת האלמנט

אפיון האלמנט העיצובי:

Flex: 1;

טבלה של הערכים העיקריים:		
1	Flex-grow	קובע כמה האלמנט יתרחב ביחס לשאר האלמנטים במיכל
2	Flex-shrink	קובע כמה האלמנט יקטן ביחס לשאר האלמנטים במיכל
3	Auto	זהה לערך flex:1
4	None	לא יציג את האלמנט
5	Flax-basis	אורך הבסיס של האלמנט
6	Flex-grow	קובע כמה האלמנט יתרחב ביחס לשאר האלמנטים במיכל

קישור להבנת flex-grow בעזרת W3Schools

https://www.w3schools.com/cssref/playit.asp?filename=playcss_flex-grow&preval=initial

קישור להבנת flex-shrink בעזרת W3Schools

https://www.w3schools.com/cssref/playit.asp?filename=playcss_flex-shrink&preval=1

Align-self

מיקום האלמנט היחיד בתוך המיכל

אפיון האלמנט העיצובי:

align-self: flex-start;

טבלה של הערכים העיקריים:		
1	auto	רווחים מכל הצדדים – התוצאה היא שהוא ימקם אותו במרכז
2	flex-start	ימקם את האלמנט בראש המיכל
3	flex-end	ימקם את האלמנט בתחתית המיכל
4	stretch	ימקם את האלמנט מראש המיכל ועד לתחתית המיכל * הכיתוב יהיה בראש המיכל
5	baseline	דומה ל – flex-start
6	Center	ימקם את האלמנט במרכז המיכל

קישור להבנת align-self בעזרת W3Schools

https://www.w3schools.com/cssref/playit.asp?filename=playcss_align-self&preval=auto

order

קביעת סדר האלמנטים במיכל

```
order: 3;
```

אפיון האלמנט העיצובי:

קישור להבנת order בעזרת W3Schools

https://www.w3schools.com/cssref/tryit.asp?filename=trycss3_order

*** שינוי הסדר עלול לבלבל את המתכנת ולכן תמיד אעדיף לקבוע את הסדר ב -
HTML מאשר להפוך את הסדר ב - CSS**

Align-content

סידור שורות שלימות בתוך המיכל

`align-content: center;`

אפיון האלמנט העיצובי:

✓ חובה לפחות שני שורות כדי שמאפיין זה יפעל.

טבלה של הערכים העיקריים:		
1	Flex-start baseline	יקבע את האלמנטים לראש המיכל
2	Flex-end	יקבע את האלמנטים בתחתית המיכל
3	Stretch	יקבע את האלמנטים לכל אורך המיכל
4	Center	יקבע את האלמנטים במרכז המיכל
5	Space-between	יקבע את האלמנט הראשון בחלק העליון של המיכל, את החלק האחרון בחלק התחתון וכל השאר ברווים שווים ביניהם.
6	Space-around	קובע רווח אחיד מסביב לשורה של האלמנטים
7	Space-evenly	קובע רווח אחיד בין כל השורות

קישור להבנת align-content בעזרת W3Schools

https://www.w3schools.com/cssref/playit.asp?filename=playcss_align-self&preval=auto

שאלות סיכום

באמצעות איזה מאפיין קובעים את סדר האלמנטים במיכל?

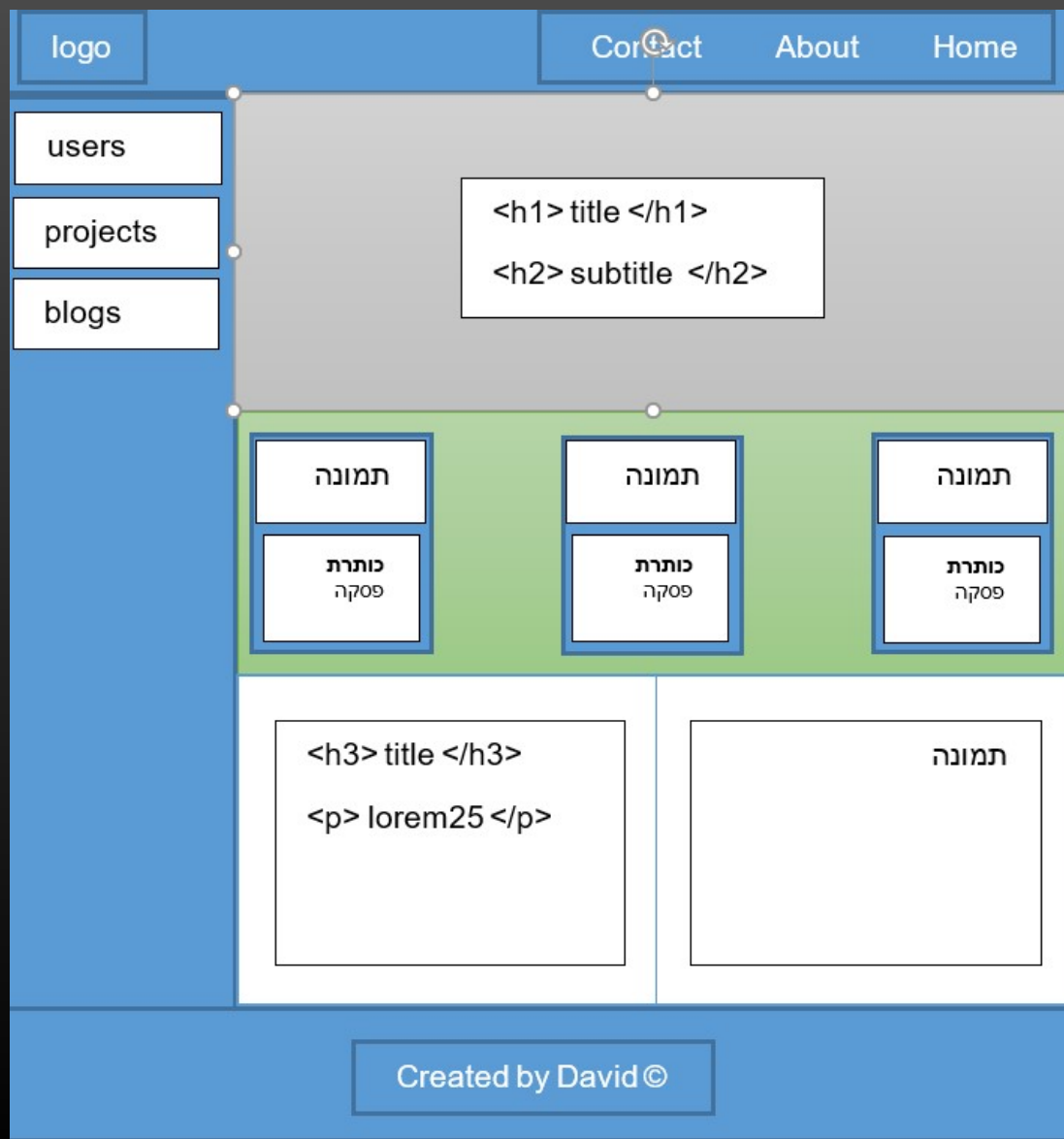
מה ההבדל בין align-self ל align-content?

איך אני קובע שאלמנט אחד יתפוס מקום כפול משאר האלמנטים במיכל?

מה יקרה אם ב container הערך של ה display יהיה block? והאם

אוכל להשתמש במאפיין flex-direction: row;

איך אני מסדר אלמנטים בצורה האנכי במיכל כך שיהיו בתחתית העמוד?



משימת סיכום בקבוצות:

להמשיך את התרגיל הקודם וליצור את הדף הבא באמצעות FlexBox:

יש ליצור קובץ חדש בשם `flex-practice.css` לקשר אותו לדף ה-HTML ובו לכתוב את כל ה-CSS

Flex בספריית Bootstrap

הספרייה bootstrap משתמשת בין היתר באלמנט העיצובי Flex על מנת לעמד את הדף וליצור רספונסיביות באתר.

היתרונות בשימוש בספרייה הזאת ושכמותה

- פחות שורות קוד לעיצוב
- שפה משותפת בין כל המתכנתים
- אפשרויות רספונסיביות נוספות
- בפרויקטים הבאים שתצטרכו להגיש תצטרכו להשתמש בספרייה הזאת.
- אפשרות להעתקת קוד מהדוגמאות שלהם

הטמעת הספרייה בקוד

- <https://getbootstrap.com/> נכנסים לדף הראשי של bootstrap
- לוחצים על כפתור Get Started
- מעתיקים את הלינק של ה – CSS
- מדביקים אותו בקובץ ה – HTML ב - vscode

הקבלה בין האלמנטים העיצוביים

בעמודים הבאים אערוך הקבלה בין האלמנטים העיצוביים של
bootstrap ל CSS – Classes של bootstrap

Display = d

d-flex

אפיון המחלקה העיצובית:

טבלה של הערכים העיקריים:		
1	Inline	יסדר את האלמנטים באותה שורה
2	Block	יסדר כל אלמנט מתחת לאלמנט הקודם
3	Flex	יסדר את האלמנטים בהתאם לפרמטרים שנגדיר לו.
4	Grid	יסדר את האלמנטים בהתאם לפרמטרים שנגדיר לו. התייחסות למיכל עם מס' שורות.
5	None	לא יציג את האלמנט
6	Table	סידור האלמנטים בתוך טבלה
7	Table-cell	הצגת תא בטבלה
8	Table-row	הצגת שורה בטבלה

קישור למאפיין ב - bootstrap

<https://getbootstrap.com/docs/5.1/utilities/display/>

@media

הספרייה bootstrap משתמשת בקיצורים על מנת להתאים את התנהגות המיכל והאלמנט לגודל המסך

d-block d-md-flex

אפיון המחלקה העיצובית:

טבלה של הערכים העיקריים:		
1	Xs	גודל מסך מ – 0 עד 575px הוא גם הערך הדיפולטיבי
2	Sm	גודל מסך מ – 576 עד 767
3	Md	גודל מסך מ – 768 עד 991
4	Lg	גודל מסך מ – 992 עד 1199
5	Xl	גודל מסך מ – 1200 עד 1399
6	Xxl	גודל מסך מ – 1400 עד אין סוף

קישור למאפיין ב - bootstrap

<https://getbootstrap.com/docs/5.1/layout/breakpoints/#media-queries>

Flex-direction = row

flex-row

אפיון המחלקה העיצובית :

טבלה של הערכים העיקריים:		
1	flex-column	יסדר את האלמנטים בתוך המיכל בסדר יורד מלמעלה למטה
2	flex-column-reverse	יסדר את האלמנטים בתוך המיכל בסדר עולה מלמטה למעלה
3	flex-row	יסדר את האלמנטים זה לצד זה משמאל לימין
4	flex-row-reverse	יסדר את האלמנטים זה לצד זה מימין לשמאל

קישור ל flex – בספריית bootstrap:

<https://getbootstrap.com/docs/5.1/utilities/flex/#align-items>

Align-items = align-items

align-items-end

אפיון המחלקה העיצובית:

✓ יש לשים בהערה את גובה ה – box על מנת לראות את האפקט בצורה מיטבית

טבלה של הערכים העיקריים:		
1	align-items-start	יקבע את האלמנטים לראש המיכל
2	align-items-end	יקבע את האלמנטים בתחתית המיכל
3	align-items-center	יקבע את האלמנטים במרכז המיכל
4	align-items-stretch	יקבע את האלמנטים לכל אורך המיכל

Justify-content = justify-content

justify-content-between

אפיון המחלקה העיצובית:

טבלה של הערכים העיקריים:		
מצמיד את האלמנטים לצד שמאל כשהכיוונית היא לצד שמאל	Justify-content-start	1
מצמיד את האלמנטים לצד ימין כשהכיוונית היא לצד שמאל	Justify-content-end	2
מצמיד את האלמנטים לאמצע	Justify-content-center	3
מצמיד את האלמנט הראשון לצד שמאל, האלמנט האחרון לצד ימין ואת כל השאר מפזר ברווחים שווים	Justify-content-between	4
קובע רווח קבוע לכל אלמנט בין אלמנט לאלמנט	Justify-content--around	5
קובע את אותו רווח בין כל האלמנטים וההתחלה והסוף	Justify-content-evenly	6

משימת ביניים:

- ליצור קובץ חדש בשם flex-bootstrap.html
- ליצור עימוד בסיסי של HTML בעזרת הפקודה `! Ctle`
- לתת כותרת לדף של Bootstrap Flex
- לשים את הקישור לספרייה של ה – CSS בדף
- ליצור את תפריט הניווט תוך שימוש בספריית bootstrap בלבד.
- נא להשתמש ב – container-fluid ו container – 1
- נא ליצור div עם class="burger" ובתוכו את ה – ascii code של `☰`
- שההמבורגר יעלם בגודל מסך md
- שתפריט הניווט הימני יופיע בגודל מסך md

Flex = col

Bootstrap מחלק את רוחב העמוד ל – 12 חלקים

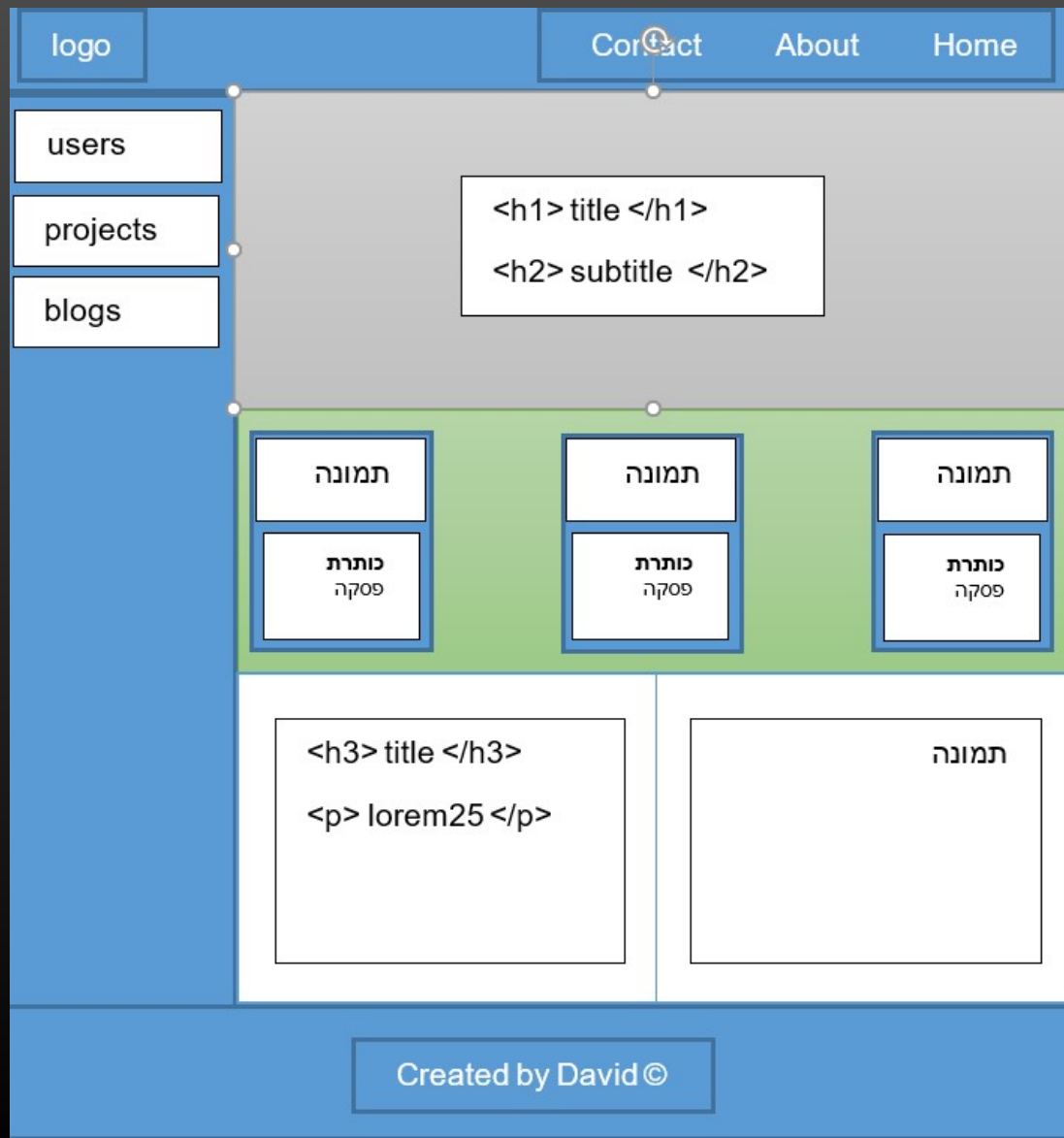
כל חלק נקרא – col

המחלקה היא ברמת האלמנט אולם היא חייבת להיות
במעטפת של row | row-reverse

col-12

דוגמה למחלקה העיצובית

דוגמה לעימוד דף בעזרת row כאשר בצד אחד יש תמונה ובצד
השני יש כיתוב עם כותרת ופסקה



משימת סיכום בקבוצות:

ליצור שוב את הדף רק הפעם
עם שימוש רק בספריית
bootstrap

שילוב CSS 1 - Bootstrap

האם כדאי לשלב בין שני צורות העבודה?

כמובן שכן. אנחנו נעשה זאת במקרים בהם אנו רוצים לחסוך בשורות קוד

לדוגמה נוכל ליצור בעצמנו אלמנט עיצובי בשם `.center{}`.

סוף