**מבוא למחשוב ענן - סמסטר חורף התשפ"ה**

**תרגיל בית 1** -– **עבודה בצוותי העבודה**

מועד הגשה: 18.12.2024

יש למנות מהנדס.ת מערכת בכל צוות, אשר יהיה אחראי על הגדרת הדרישות ההנדסיות, ועל הממשק מול החומרה. נא לרשום את שם הסטודנט.ית בתרגיל זה. על מהנדס.ת המערכת לכתוב כיצד נעשתה חלוקת העבודה מול הצוות, מה היו המשימות של כל חבר צוות, האם היה ממשק בין חברי הצוות, והאם המשימות מולאו:

|  |  |  |
| --- | --- | --- |
| **שם חבר הצוות** | **משימות שהוקצו** | **משימות שהושלמו** |
| דרור הרשקוביץ – מהנדס מערכת | תרגיל 1 – משותף לכולם אפיון פרסונה + כתיבת שאלות לראיון  ניתוח דרישות מערכת יצירת אב טיפוס | הושלם |
| לידור אנפינגר | תרגיל 1 – משותף לכולם  פתיחת גיט קבוצתי אפיון פרסונה + כתיבת שאלות לראיון ניתוח דרישות מערכת יצירת אב טיפוס | הושלם |
| אביב אייטק | תרגיל 1 – משותף לכולם  אפיון פרסונה + כתיבת שאלות לראיון ניתוח דרישות מערכת יצירת אב טיפוס | הושלם |
| היבה אבו קנדיל | תרגיל 1 – משותף לכולם  ניתוח דרישות מערכת UseCase | הושלם |
| פאטמה זועבי | תרגיל 1 – משותף לכולם  ניתוח דרישות מערכת UseCase | הושלם |
| בר הרוש | תרגיל 1 – משותף לכולם  ניתוח דרישות מערכת UseCase | הושלם |

**תרגיל 1:**

יש לבחור סיפור הצלחה של הטמעת ענן לבחירתכם, ולנתח אותו לפי הקריטריונים הבאים:

NETFLIX

1. האם נעשה שימוש בענן פרטי/ציבורי/היברידי?

ציבורי - נטפליקס משתמשת בתשתיות AWS : ספק ענן ציבורי שמשרת לקוחות רבים על אותה תשתית משותפת.  
 נטפליקס לא מחזיקה דאטה סנטר משלה (ענן פרטי) ולא משלבת תשתיות פרטיות עם ציבוריות   
(מודל ענן היברידי).   
לכן זהו מודל של ענן ציבורי.

1. מודל שירות – SAAS/PAAS/IAAS  
     
   IaaS -

* שימוש בשרתים וירטואליים של AWS.
* אחסון מאסיבי לתוכן וידאו.
* רשתות וירטואליות.

PaaS -

* קידוד וידאו או ניתוחי דאטה, כדי להתרכז בפיתוח חוויית המשתמש.
* מערכות ניהול מסדי נתונים מנוהלות.
* כלי פיתוח ואוטומציה מובנים.

1. הציעו שלוש מטריקות לבדיקת הצלחת ההטמעה. נמקו במשפט קצר כל הצעה. מטריקות לדומגא נמצאות בהרצאה 3, ראו קישור:  
   <https://guidingmetrics.com/content/cloud-services-industrys-10-most-critical-metrics/>  
     
   זמן תגובה - כאשר צופה משתמש באפליקציית נטפליקס זמן התגובה הוא חשוב מאחר והלקוח משלם באופן חודשי על מנת להינות מתכנים שישודרו ללא עצירות ומגבלות ולשם כך הענן צריך לספק זמן תגובה מהיר וללא הפרעות.   
   אמינות - בעת תקלה בשימוש באפליקציה יש צורך בתיקון מהיר ככל האפשר על מנת שהלקוח יוכל להמשיך לצפות בתכנים ( ניקח לדוגמה את המקרה שארע כאשר נטפליקס שידרו בשידור ישיר קרב איגרוף ומעומס הצופים נפלו השרתים והיה צורך בטיפול מהיר על מנת לתקן את הבעיה ולהחזיר ללקוחות את האפשרות להנות מחווית צפייה חלקה )   
   יכולת להתרחב (scalability) -סרטים וסדרות של נטפליקס מצולמים באופן שוטף ועולים תכנים חדשים כמעט בכל יום ולצורך כך יש לאפשר אופציה להתרחבות משמעותית כאשר בנוסף נטפליקס הוסיפה שירות גיימינג הכולל משחקים, זהו שירות שהתווסף לפלטפורמה הקיימת ולכן היכולת להתרחבות חשובה.
2. האם הייתם מציעים לארגון ענן אחר? מודל אחר? התיחסו למסקנות הסיפור.  
     
   נכון להיום, AWS מספקת לנטפליקס פתרונות טובים שמותאמים לצרכיהם, והמעבר עלול להיות יקר ומורכב. עם זאת, אם שיקולים כמו עלות, עצמאות עסקית, או דרישות טכנולוגיות חדשות יהפכו לדומיננטיים יותר, ייתכן שנטפליקס תצטרך לבחון מעבר לארגון ענן אחר.  
   לדעתנו נטפליקס לא צריכה לשנות את המודל מאחר ונטפליקס היא שירות סטרימינג.

1. יש לצרף קישור **מלא** לאתר האינטרנט ממנו נלקח הסיפור.

ניתן להעזר למשל באתר: <https://aws.amazon.com/solutions/case-studies/>

<https://www.pc.co.il/news/206937/>

תרגיל 2: Design thinking

בהמשך לסדנה בשבוע 4, תכננו את אפליקציית

"[Cloud Access\_[BEE"

(החליפו את המילה TEAM בשם הצוות שלכם, אפשר גם להוסיף לוגו)

עליכם לתכנן מנוע חיפוש, המסייע לגולשים להכיר נושאים שונים בתחום מחשוב ענן.

כל צוות יבחר אחד מהאתרים הנמצאים ברשימה להלן:

<https://docs.google.com/spreadsheets/d/11Lyni94IKg1O5Py2t3mIL1QhITCnJCMV_-MB_Ey73NE/edit?gid=0#gid=0>

בצעו תהליך של חשיבה עיצובית כפי שעשיתם בסדנה בהרצאה:

1. רשמו את שם האתר שנבחר, ופסקה קצרה של הסבר והקשר (קונטקסט).

HEROKU -

היא פלטפורמת PaaSשבבעלות Salesforce. היא מאפשרת למפתחים להתמקד בכתיבת קוד ופיתוח אפליקציות מבלי לדאוג לניהול תשתיות. הפלטפורמה תומכת במגוון שפות תכנות כמו Ruby, Node.js, Python ו-Java, ומציעה כלים מובנים לפריסה, ניטור וניהול אפליקציות בענן.

1. בצעו ראיון קצר עם דמות מרכזית (אמיתית) המייצגת משתמש במערכת. הגדירו את הפרסונה.ציירו empathy map.

- הגדרת פרסונה:

מאפיינים:

שם: ליאון קלרק

גיל:32

תפקיד: Solutions Architect

רקע: 6 שנות ניסיון בפיתוח ותכנון פתרונות ענן, מומחה בPaaS עובד Heroku מזה 3

שנים.

אחריות: מסייע ללקוחות בתכנון וארכיטקטורת הפתרונות שלהם על פלטפורמת Heroku

עובד מול צוותי פיתוח ומנהלי מוצר.

-ראיון:

1. באיזה סיטואציות במהלך העבודה היומיומית שלך אתה נדרש לחפש מידע על מושגי ענן?
2. כשאתה מסביר ללקוחות על טכנולוגיות ענן, איזה סוג מידע אתה צריך שיהיה נגיש לך במהירות?
3. מתוך ניסיונך ב-Heroku, אילו מושגי ענן הכי חשוב להבין כדי לעבוד עם הפלטפורמה?
4. כיצד אתה מעדיף שיוצג הקשר בין מושגי ענן שונים?
5. איזה סוג השוואות בין פתרונות ענן שונים היית רוצה לראות כחלק מהגדרות המושגים?
6. כשאתה מחפש מידע ספציפי על שירות מסוים, איזה פרטים טכניים חשוב לך למצוא במהירות?
7. אילו כלים או תכונות באתר יעזרו לך להישאר מעודכן בחידושים ושינויים בעולם הענן?

-תשובות לראיון:

1. במהלך העבודה היומיומית אני נדרש לחפש מידע על מושגי ענן בעיקר בשתי סיטואציות: כשאני מכין פתרון מותאם ללקוח חדש שדורש אינטגרציה עם שירותים שונים, וכשאני נתקל בבעיות ביצועים שדורשות הבנה מעמיקה של הארכיטקטורה.
2. בהסברים ללקוחות, אני צריך גישה מהירה להשוואות בין שירותים דומים, מגבלות טכניות של כל פתרון, ודוגמאות קוד מעשיות. חשוב לי במיוחד לראות use cases אמיתיים שיכולים להמחיש את היתרונות של כל פתרון.
3. מניסיוני עם Heroku, המושגים החשובים ביותר הם: Dynos (סוגים והתנהגות שונים), תוספים (ואיך הם משתלבים), Container Registry, Build Packs ו-Release Phase. הבנה מעמיקה של מושגים אלה היא קריטית לתכנון נכון של יישום.
4. לגבי הקשרים בין מושגים, אני מעדיף ויזואליזציה אינטראקטיבית שמראה את הקשרים ההיררכיים והתפעוליים. למשל, איך Dyno משתמש ב-Containers וכיצד זה מתקשר למערכת הBuild
5. בהשוואות בין פתרונות ענן, הייתי רוצה לראות טבלאות השוואה מפורטות שכוללות: מחירים, מגבלות טכניות, יכולות scaling, ואפשרויות ניטור. במיוחד חשוב לי להשוות בין Heroku לפתרונות PaaS אחרים.
6. כשאני מחפש מידע על שירות ספציפי, חשוב לי למצוא במהירות: מגבלות resource, אפשרויות קונפיגורציה, מחירים, ודרישות תשתית. במיוחד חשוב לי לראות best practices וידע על troubleshooting נפוץ.
7. הכלים שיעזרו לי להישאר מעודכן הם: RSS feed של עדכונים טכניים, התראות על שינויים במושגים שסימנתי כמועדפים, ואפשרות לראות השוואה היסטורית של שינויים בהגדרות ובמאפיינים של שירותים שונים.

|  |  |
| --- | --- |
| THINKS:   * חושב על דרכים לאופטימיזציה של עלויות עבור הלקוחות * חושש מהחמצת פרטים טכניים חשובים בפתרונות * חושב על איך לשפר את הביצועים והיעילות של פרויקטים | DOES:   * מנהל פגישות ייעוץ טכניות * נותן פתרונות בזמן אמת * משתתף בתכנון אסטרטגי * מתעדכן בחידושי הפלטפורמה |
| FEELS:   * מרגיש סיפוק כשלקוחות מצליחים עם הפתרונות שלו * מרגיש אחריות גדולה לספק פתרונות אמינים ללקוחות * מתרגש מאתגרים טכניים מורכבים | SAYS:   * מייעץ ללקוחות על ארכיטקטורה מיטבית * חוקר פתרונות חדשים * משתתף בפגישות תכנון טכני * כותב תיעוד טכני * מנטר ביצועי מערכת |

1. בצעו תהליך של divergent thinking. רשמו את כל הרעיונות שעלו.

* חיפוש קולי למושגים
* תרגום אוטומטי של מושגים לשפות שונות
* כלי השוואה ויזואלי בין פלטפורמות
* קטעי וידאו קצרים להסבר מושגים מורכבים
* בנק שאלות נפוצות מלקוחות
* תבניות מוכנות לארכיטקטורות נפוצות
* היסטוריית חיפושים חכמה
* דשבורד אישי עם סטטיסטיקות שימוש
* צ'אטבוט לשאלות טכניות
* פורום מומחים
* קבוצות דיון לפי נושאים

1. בצעו תהליך של convergent thinking. רשמו את כל השיפורים שעלו.

נתמקד ב-2 היבטים מרכזיים:  
1. מידע מקצועי ומעשי –

* חיפוש מתקדם עם סינון
* הצגת דיאגרמות מפולח לפי נושאים

2. כלים –

* היסטוריית חיפושים
* סימון מושגים מועדפים
* דשבורד אישי עם סטטיסטיקות שימוש
* צ'אטבוט

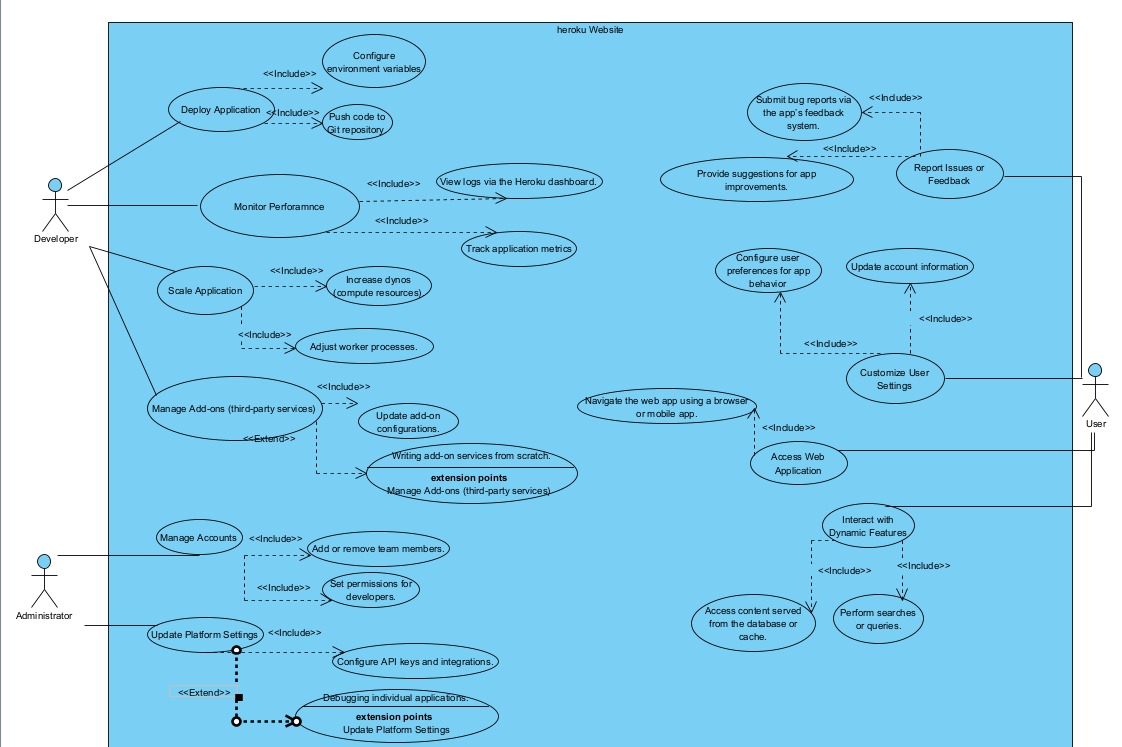
1. רשמו 5 דרישות פונקציונליות מרכזיות ו-5 דרישות לא פונקציונליות מרכזיות. יש לסווג את הדרישות הלא פונקציונליות לפי:  
   <https://en.wikipedia.org/wiki/Non-functional_requirement>

**דרישות פונקציונאליות:**

* + - 1. מנוע חיפוש מתקדם: חיפוש מושגים וסינון תוצאות לפי רמת מורכבות, תאריך עדכון, וסוג הפתרון (טכני, מעשי).
      2. השוואת שירותים בענן: הצגת טבלאות השוואה בין פתרונות ענן (מחירים, מגבלות טכניות, scaling, ניטור).
      3. ויזואליזציה אינטראקטיבית: כלי להצגת הקשרים בין מושגי ענן כמו Dynos, Containers, ותהליכי Build.
      4. דשבורד מותאם אישית: מעקב אחרי היסטוריית חיפושים, מושגים מועדפים, ועדכונים טכניים מותאמים.
      5. כלי תמיכה טכני משולב: צ'אטבוט לתשובות מהירות, גישה למאגר FAQ.
      6. נגישות: המערכת תאפשר הנגשת מידע לבעלי מוגבלויות ע"י לחיצה על כפתור.

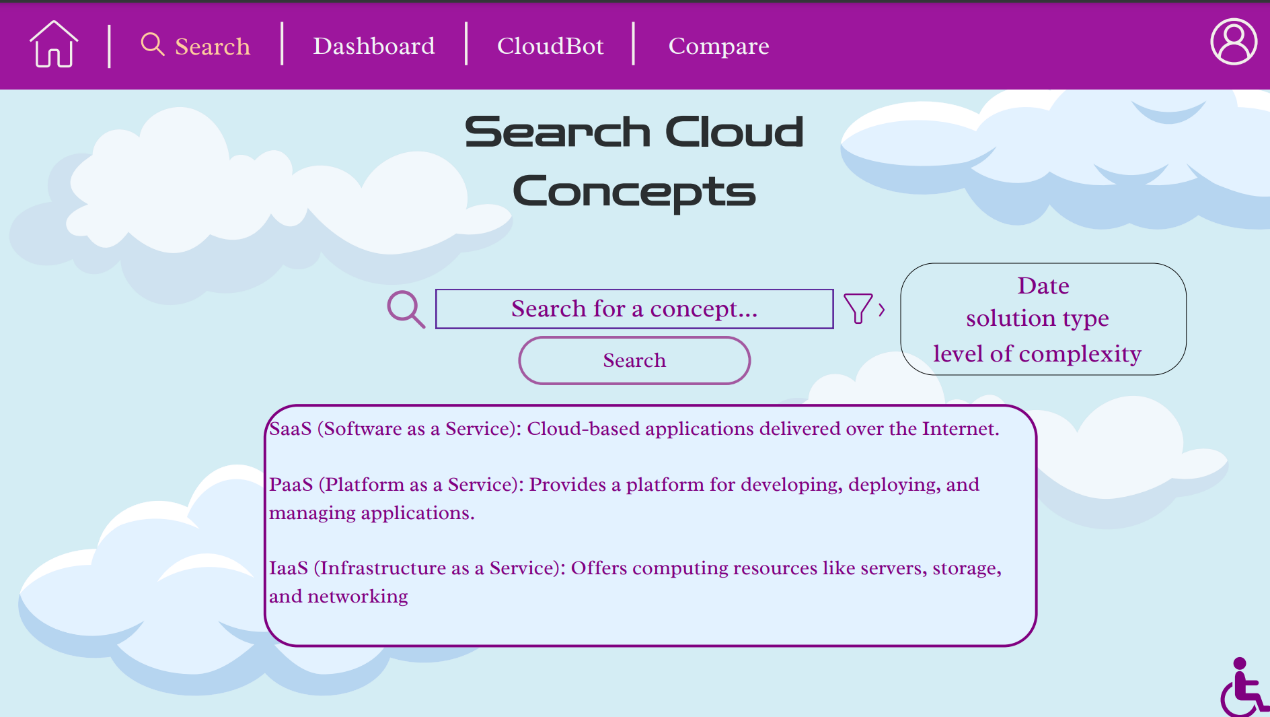
**דרישות לא פונקציונליות :**

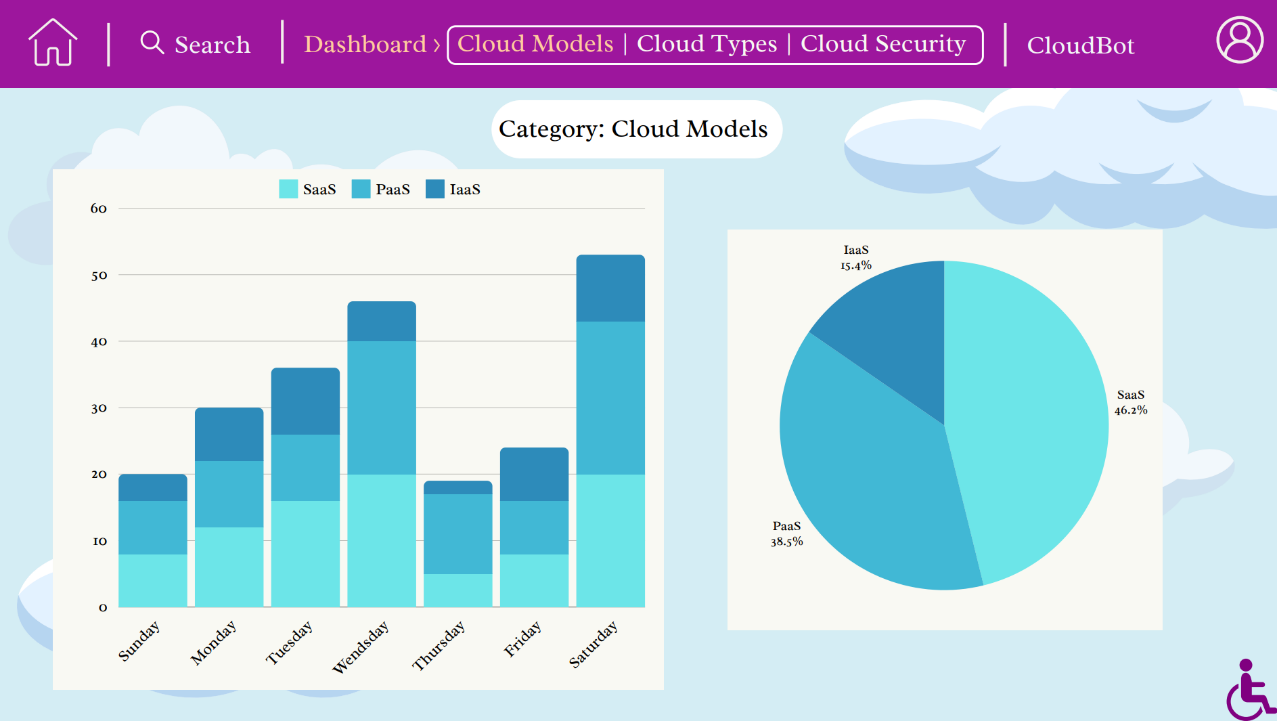
1. **ביצועים גבוהים**: זמני תגובה מהירים לשאילתות חיפוש וכלי השוואה.
2. **ממשק משתמש נוח**: עיצוב UI/UX שמותאם למשתמשים מקצועיים, כולל פשטות ויזואלית ונגישות למידע קריטי.
3. **תאימות:** האפליקציה תתפקד באופן רספונסיבי לכל גדלי המסך השונים בהם המשתמש יבצע את החיפושים.
4. **תמיכה בריבוי שפות**: יכולת להציג מושגים והסברים באנגלית, עברית ושפות נוספות.
5. **זמינות גבוהה**: הבטחת זמינות מערכתית למניעת השבתות בזמן קריטי.
6. הציגו תרשים USE CASE של האתר.

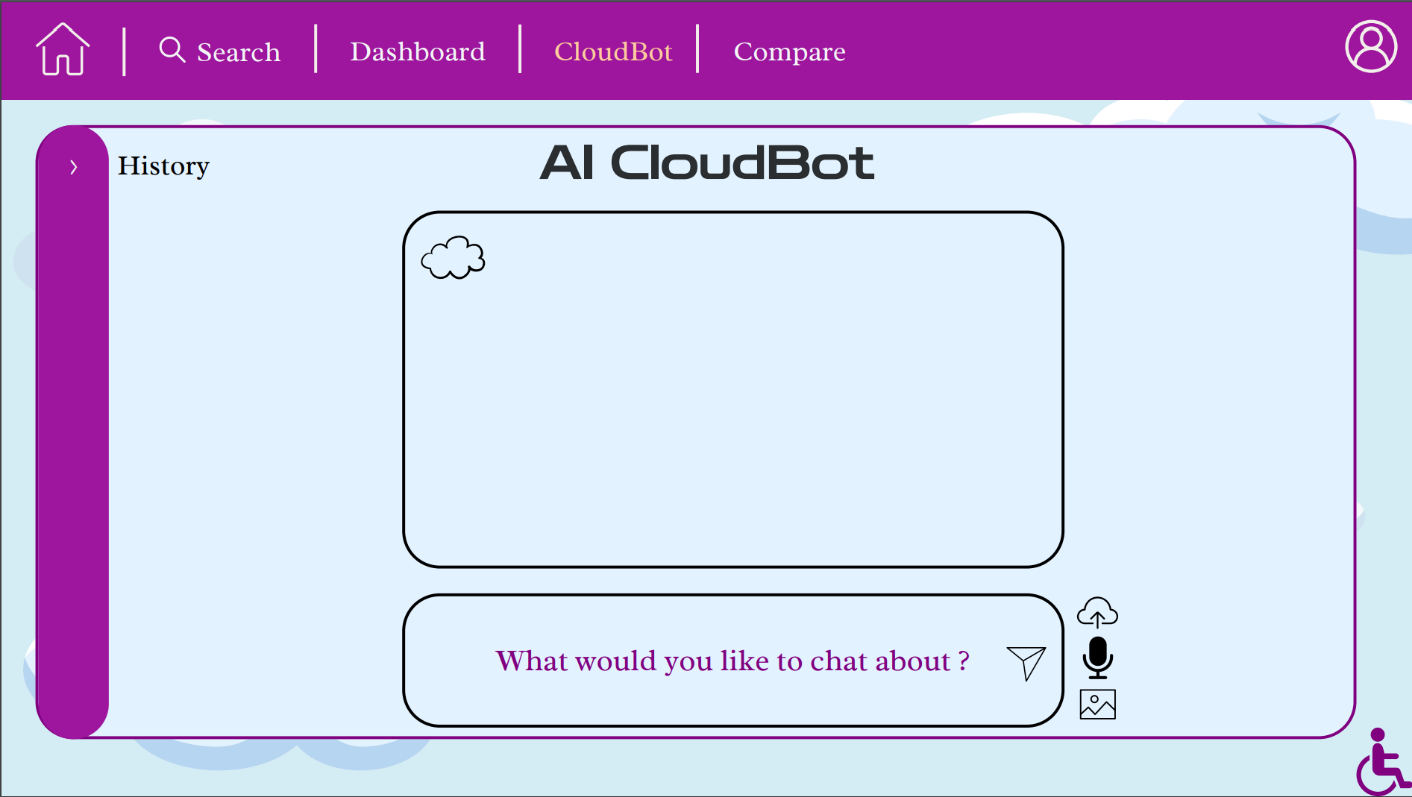


1. הדגימו אב טיפוס מנייר (מסכים המתארים את המערכת) ,והסבירו את כל האלמנטים המרכזיים בו. התייחסו להערות שניתנו לכם בהרצאה 5 על המסכים שהראיתם בכיתה.

לנוחותכם, אתר הקורס כולל תבנית לכל המשימות (כפי שביצעתם בכיתה)



הסבר: מסך חיפוש, ניתן לבצע סינון לפי המסננים.  
קבלת תשובה בחלק נפרד (החלק התחתון) – בנוסף קיים כפתור נגישות.  
הסבר: מסך סטטיסטיקות לפי קטגוריות שונות, מראה גרפים שונים בהתאם לשימוש באפליקציה.



A screenshot of a computer

Description automatically generatedהסבר: צ'אטבוט מבוסס בינה מלאכותית להסבר מורחב יותר ותחושת שיח מול אדם מהצד השני לצורך הבהרות.  
בנוסף קיים כפתור היסטוריה לראות שיחות קודמות.

הסבר: מסך השוואות בין ספקי ענן מובילים לפי קטגוריות מסוימות.

הנחיות:

1. יש להגיש את התרגיל בצוותים, בתיקיית ה –GIT שלכם (צרפו קישור), וכן בתיקייית התרגיל ב moodle
2. כותרתו של הקובץ תהיה HW1\_TEAMNAME
3. שימו לב כי כל העבודות חייבות להיות שונות זו מזו. עבודות שייראו דומות ייפסלו ויינתן עליהן ציון 0.

בהצלחה!