Logo

Description automatically generated

**אתר שאלות ותשובות PureStack**

*פרוייקט גמר בהנדסת תוכנה תכנון שירותי רשת*

מגיש: **טוהר אננייב**

כיתה יב3

מנחים גדי הרמן וזאב בנקבצר

תוכן עניינים

[מבוא 2](#_Toc68443724)

[ייזום 2](#_Toc68443725)

[דרישות עיקריות 2](#_Toc68443726)

[בחירת טכנולוגיה 3](#_Toc68443727)

[ארכיטקטורה כללית 3](#_Toc68443728)

[מסד נתונים 4](#_Toc68443729)

[מחלקות וגישה למסד 5](#_Toc68443730)

[ממשק משתמש 6](#_Toc68443731)

[דף הבית\נחיתה 6](#_Toc68443732)

[הרשמה 7](#_Toc68443733)

[הזדהות 8](#_Toc68443734)

[צפייה בשאלות 8](#_Toc68443735)

[הוספת שאלה 9](#_Toc68443736)

[צפייה ומענה על שאלה 9](#_Toc68443737)

[פרופיל משתמש 10](#_Toc68443738)

[סטטיסטיקות 10](#_Toc68443739)

[סוגיות נבחרות 11](#_Toc68443740)

[מימוש הצבעה 11](#_Toc68443741)

[תצוגת סטטיסטיקה 11](#_Toc68443742)

[ניהול Sessions 12](#_Toc68443743)

[מראה דפים עקבי 12](#_Toc68443744)

[העלאת תמונות 12](#_Toc68443745)

# מבוא

‏ הפרוייקט שלי מייצר אתר אינטרנט שמאפשר לקהילת משתמשים לשתף פעולה במענה על שאלות שונות של הקהילה. בחרתי בפרוייקט הזה בגלל הרקע שלי במשחקי רשת וקהילות אינטרנטיות, אני שותף פעיל בהרבה פורומים אינטרנטיים כגון StackOverflow, Reddit, Discord וכו', ותמיד עניין אותי איך אתרים כאלו בנויים.

# ייזום

**שם הארגון**: PureStack

**מטרת הפרוייקט** היא לספק פורום שנותן למשתמשים מקום בטוח לשאול שאלות ולקבל תשובות על שאלות אלו, ובנוסף לכך מקום בטוח לענות על שאלות שאחרים שאלו ולקבל ניקוד על שאלות לעניין ותשובות שעזרו לאחרים.

**קהל היעד** של האתר מורכב מקהל רחב ששואל שאלות, וקהל זה מסוגל לענות על שאלות שאחרים שאלו, והם יודעים את התשובה להם, כך שכביכול, קהל היעד הוא כל אדם שמסוגל ומעוניין לענות על שאלות שנשאלו באתר.

כמו בכל קהילה וירטואלית, נדרש עידוד לאנשים לענות על שאלות. דבר זה מושג על ידי Swag points, המהווה אלמנט של תחרותיות (gamification ). בנוסף, Swag points מהווים אסמכתה למוניטין של העונה ומאפשרים למשתמשים אחרים ליצור אמון בתשובות.

## דרישות עיקריות

1. משתמשים יכולים לצפות בשאלות ותשובות שנמצאים באתר
2. משתמשים יכולים להגיש שאלות חדשות באתר.
3. משתמשים יכולים לענות על שאלות קיימות של אחרים.
4. משתמשים יכולים לדרג שאלות ותשובות.
5. משתמשים יכולים לראות דירוג של שאלות ותשובות.
6. משתמשים יכולים לראות דירוג של משתמשים אחרים.
7. משתמשים יכולים לצפות בפרופיל של אנשים אחרים.
8. משתמשים יכולים לערוך ולמחוק שאלות ותשובות שהם הגישו.
9. משתמשים יכולים לצפות בלוח דירוג של משתמשים באתר.
10. המערכת תאפשר רישום של משתמשים חדשים בקלות.
11. המערכת תאפשר הזדהות של משתמש בזמן כניסה לאתר.
12. משתמש חייב להזדהות כדי להגיש שאלה, לענות על שאלה ולדרג שאלות ותשובות.
13. משתמש לא חייב להזדהות כדי לצפות בשאלות ותשובות.
14. פרופיל של משתמש יכלול את הפרטים שלו כולל תמונה לפי בחירתו.
15. ממשק משתמש חייב להיות קל ונוח כדי לעודד משתמשים לשתף פעולה.
16. ניהול משתמשי אדמין שמקבלים הרשאות נוספות כגון מחיקת שאלות ותשובות של משתמשים אחרים.
17. המערכת תתמוך בדפדפנים מודרניים כולל Chrome, Edge, FireFox.

## בחירת טכנולוגיה

בחרתי לממש את הפרוייקט בעזרת מסד נתונים של MySQL וASP.NET/Razor

בחירה ב**מסד נתונים SQL** נעשתה משני טעמים עיקריים:

1. יכוולת הכנסה ועדכון נתונים בקלות (CRUD)
2. יכולת הצלבה קלה בין כמה טבלאות שונות, דבר שבא לידי ביטוי במיוחד לחישוב סטטיסטיקות מסובכות.

בחרתי ב**טכנולוגיית ASP.NET** כדי לממש ממשק html web דינמי. ASP.NET מאפשר יצירת ממשקי html בקלות, אשר נתמכים ברוב דפדפנים מודרניים ומאפשרים יכולת שימוש באתר מכל מחשב, ואפילו טלפון סלולרי. בנוסף בחרתי להשתמש ב-**Razor** מהטעמים הבאים:

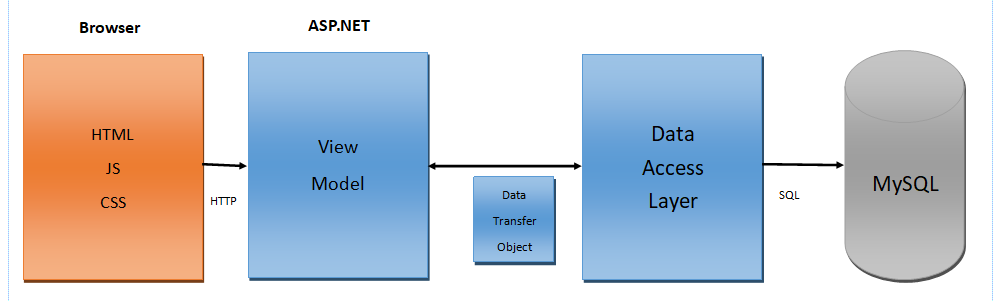
1. הפרדת אחריות בין מחלקות view וmodel כאשר כל אחד מהם מבצעת את התפקיד שלה כאשר view נשאר "נקי", וניתן לעריכה ויזואלית בקלות.
2. Razor מצויין לבניית ממשקי web שבעיקר מציגים תוכן ולא דורשים לוגיקה מורכבת כגון routing.

בנוסף נעשה שימוש בתכנות client side בעזרת JS והפעלת פעולות בצד השרת ללא טעינה נוספת, כדי לאפשר ממשק נוח וקל לשימוש (דוגמה, הצבעות).

## ארכיטקטורה כללית

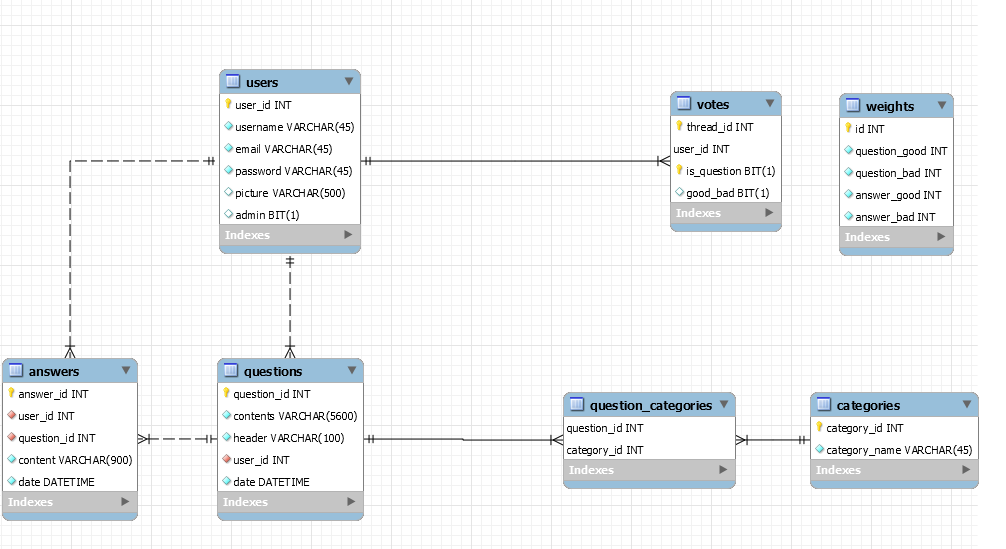
ארכיטקטורת האתר בנויה בצורה רגילה של שלוש שכבות:

1. מסד נתונים
2. שכבת לוגיקה עסקית המאפשרת גישה לנתונים במסד
3. שכבת הפרזנטציה שכוללת קוד ASP.NET הרץ בשרת וhtml/JS שרץ בתוך דפדפן



# מסד נתונים

להלן שרטוט ERD של מסד הנתונים:



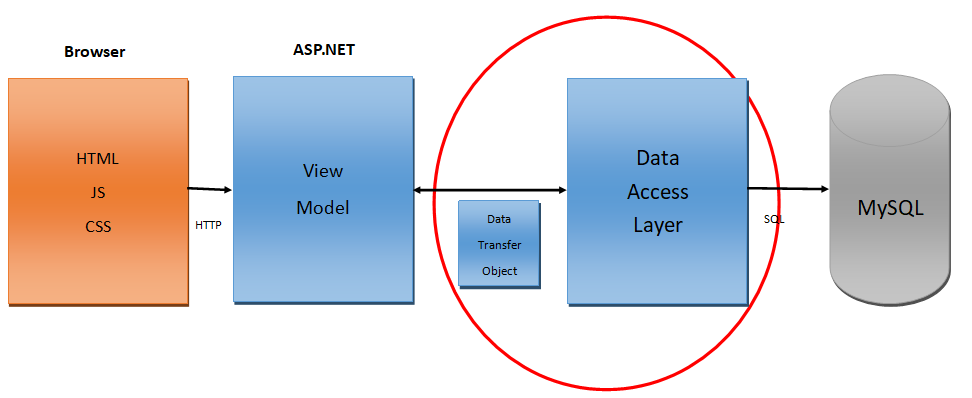
מסד נתונים כולל טבלאות הבאות:

|  |  |
| --- | --- |
| שם טבלה | תיאור |
| Users | מכיל את הפרטים של המשתמשים, כולל שם משתמש, סיסמה ואימייל. בנוסף לכך יש סימון של אדמין ושם קובץ של התמונת פרופיל. |
| Questions | מכיל את הפרטים של השאלות, כולל מפתח משתמש ששאל את השאלה, ותאריך שאילת השאלה. |
| Answers | מכיל את פרטי התשובה, כולל מפתח המשתמש שענה את התשובה, ומפתח לשאלה שהתשובה מתייחסת אליה. |
| Votes | מכיל את פרטי ההצבעות, עבור שאלות ותשובות יחד. הטבלה מכילה עמודה שמתארת אם ההצבעה הינה עבור שאלה או עבור תשובה. בנוסף לכך, יש עמודה שמתארת האם ההצבעה בעד או נגד (good\_bad). |
| Categories | מכיל קטגוריות שלפיהם אפשר לקבץ ולחפש שאלות |
| Question\_Categories | טבלת קישור בין שאלות לקטגוריות, מפני ששאלה יכולה להשתייך לכמה קטגוריות (many to many) |
| Weights | טבלה קונפיגורציה של משקל ההצבעות. |

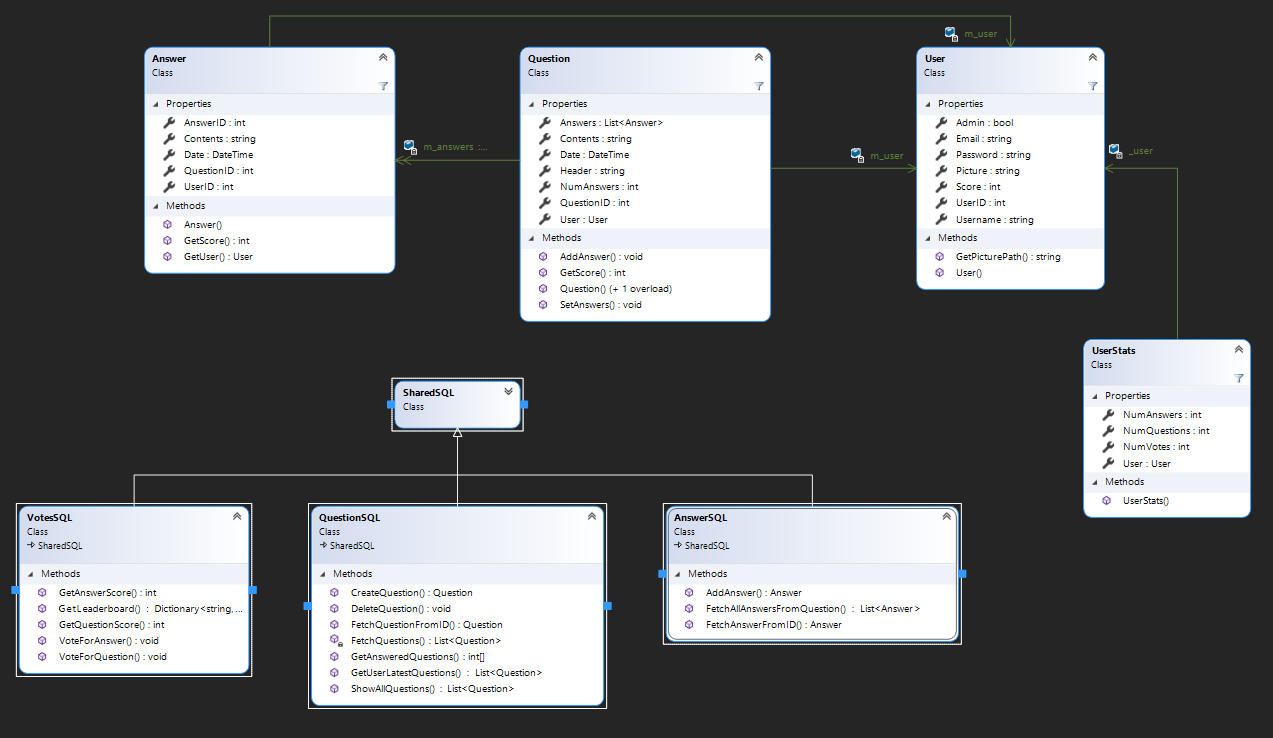
# מחלקות וגישה למסד

לצורך גישה למסד יצרתי שתי סוגי מחלקות:

1. **מחלקות Data Access Layer** כגון AnswerSQL, QuestionSQL וכדומה. מחלקות אלו מהוות אבסטרקציה מעל מסד נתונים, וחושפות ממשק לוגי לגישה לעצמים רלוונטים, כאשר המימוש שלהם מתרגם את הפעולות לפקודות SQL המתאימות.
2. **מחלקות Data Transfer Object** כגון User, Question, Answer אשר כוללות פרטים של עצמים רלוונטיים. האובייקטים האלו לרוב נוצרים על ידי Data Transfer Layer כתרגום של תוצאות של שאילתות.



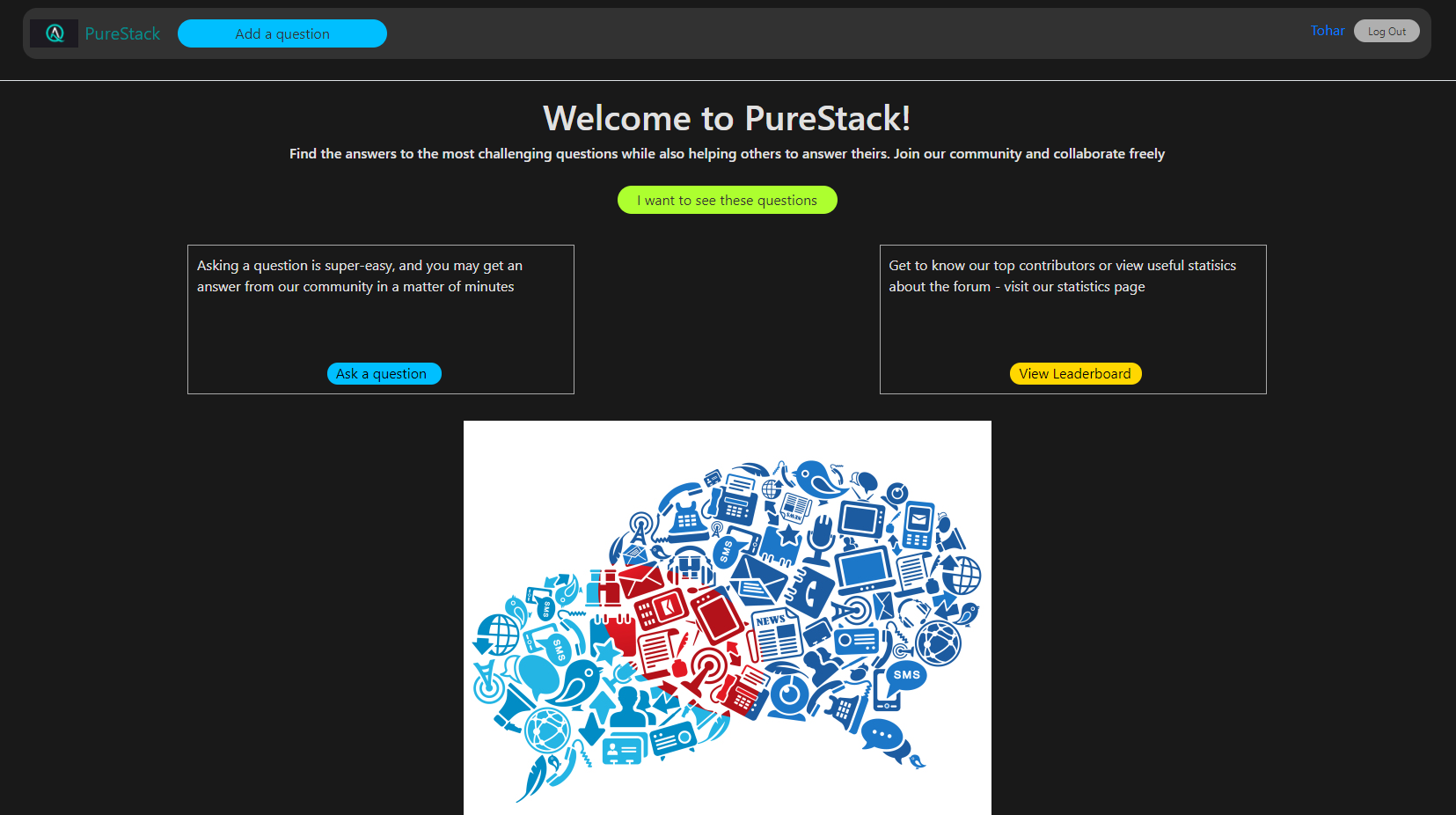
להלן דיאגרמת UML למחלקות אלו:



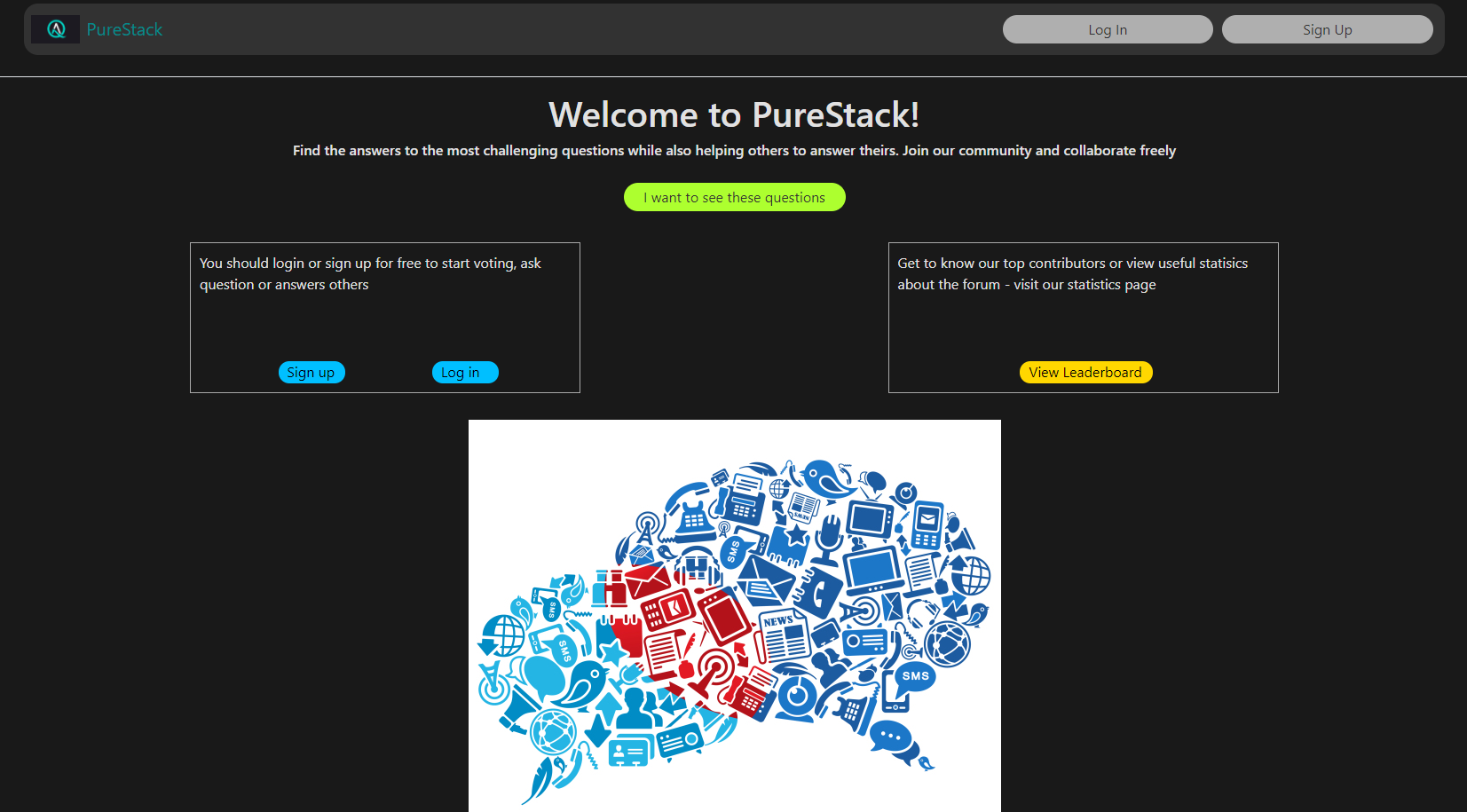
# ממשק משתמש

## דף הבית\נחיתה

דף זה משמש כשער ראשי לאתר, זהו דף שמתמש חדש אמור לפגוש בפעם הראשונה שהוא נכנס לאתר. תפקידו להזמין משתמש לקחת חלק בקהילה, ולהציע לו בצורה קלה אופציות כגון הרשמה, צפיה בשאלות וצפיה בסטטיסטיקות.



דף הבית כשהמשתמש לא רשום



## הרשמה

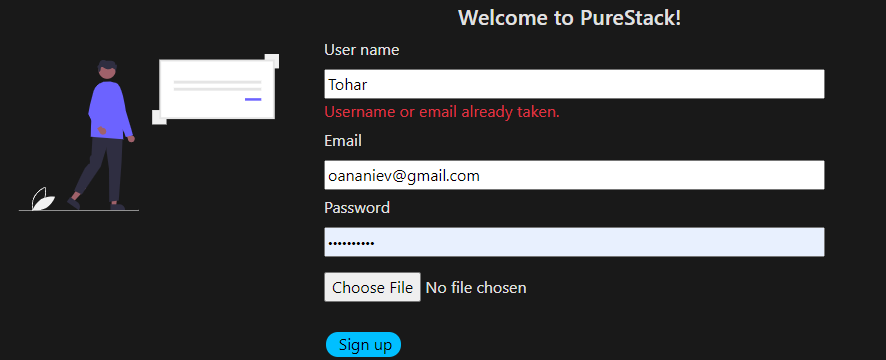
דף זה משמש כדף רישום לאתר. המשתמש מתבקש להזין שם משתמש, דואר אלקטרוני, סיסמה. בנוסף לכך הוא מוזמן להעלות תמונת פרופיל, אך אם הוא לא מזין אחת, תמונת ברירת מחדל תשמש לו כתמונת פרופיל.



תמונת ברירת המחדל

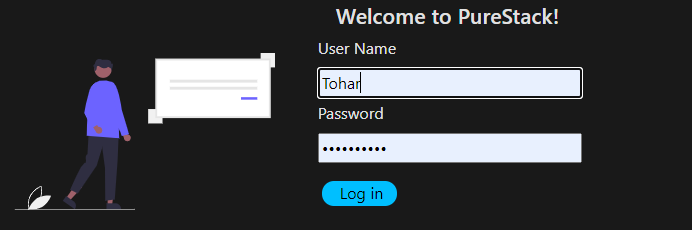


המערכת בודקת ששם המשתמש והדוא"ל הם חוקיים ולא רשומים במערכת. המערכת גם בודקת גם שאורך הסיסמה תקין.

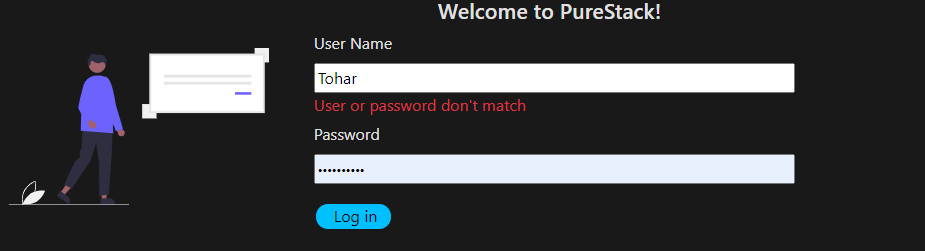


## הזדהות

דף זה משמש כדף שבו משתמש מזדהה במערכת, בכך שהוא מזין שם משתמש וסיסמה.

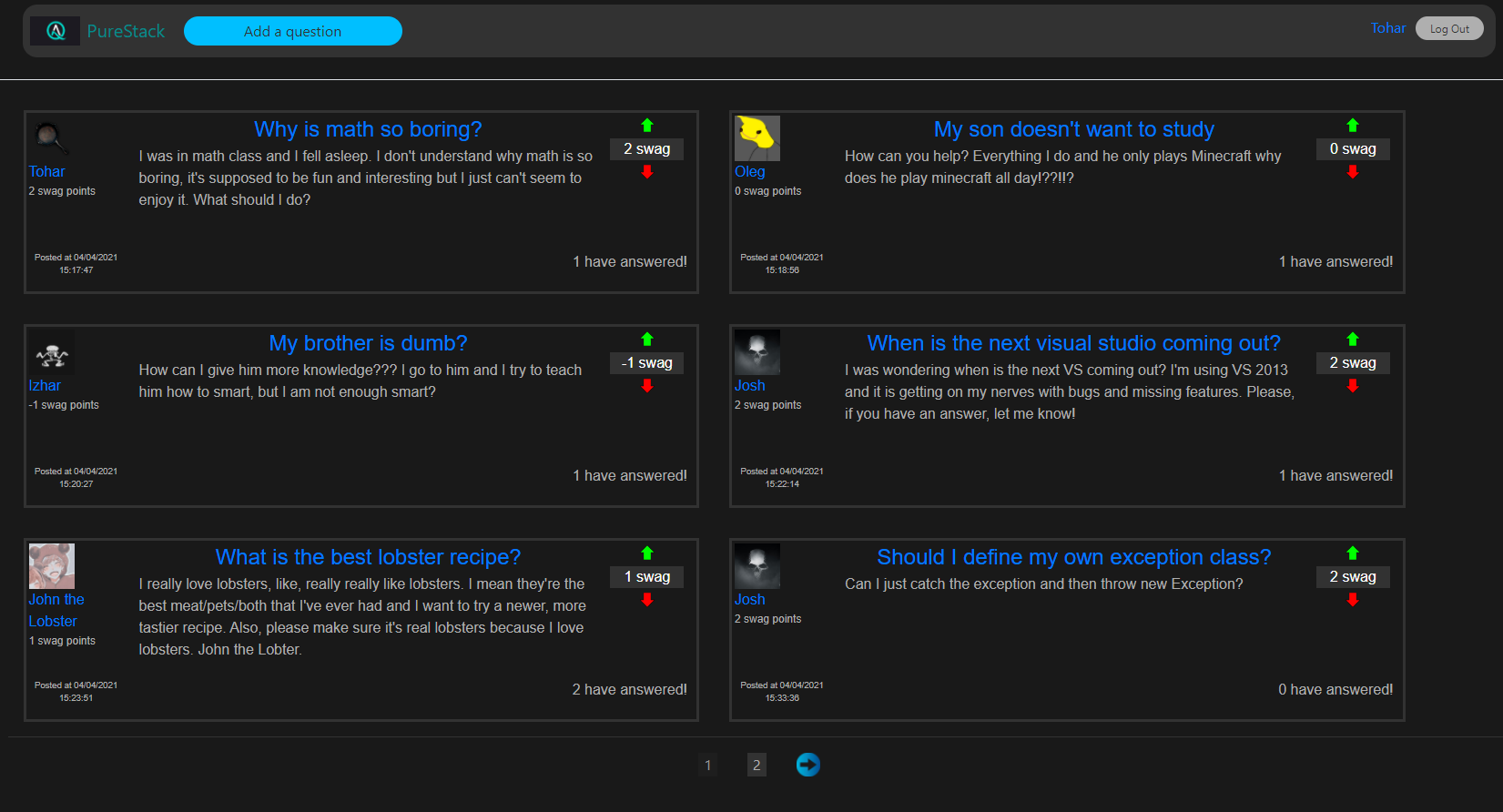


המערכת בודקת ששם המשתמש וסיסמה תואמים, על ידי שאילתה במסד הנתונים.



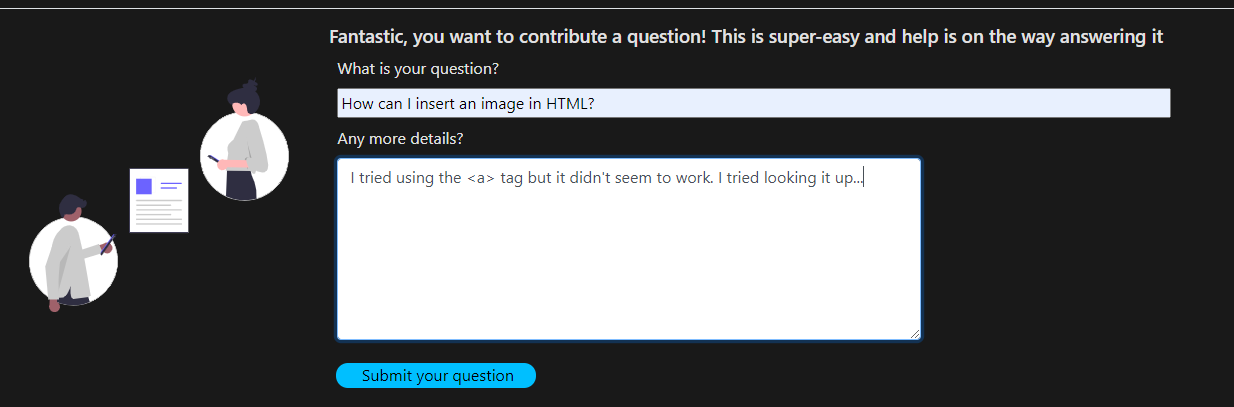
## צפייה בשאלות

דף זה הוא הדף הראשי של המערכת. לכאן מגיעים משתמשים כדי לצפות ולעיין בשאלות הקיימות באתר. למימוש הדף השתמשתי במנגנון דפדוף לוגי אשר מציג 4 שאלות בכל פעם.ומאפשר מעבר קל לשאלות הבאות.



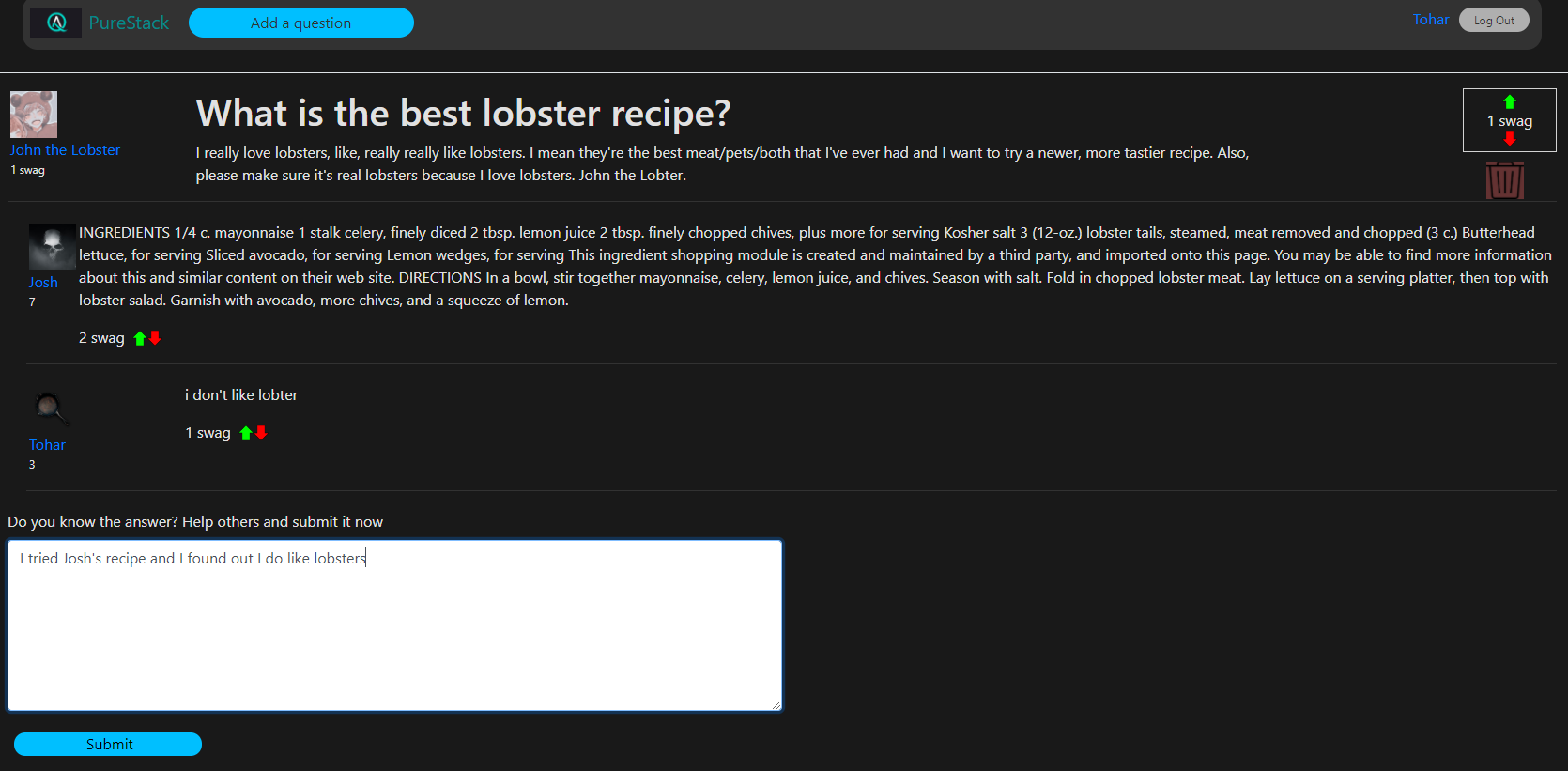
## הוספת שאלה

בדף זה משתמש מתבקש להזין שאלה. לדף זה מורשים להיכנס רק משתמשים שרשומים ומזוהים באתר.



## צפייה ומענה על שאלה

בדף זה משתמשים יכולים לצפות בשאלה מסויימת, בתוכן שלה, להצביע עבורה, לענות עלייה ולראות תשובות של משתמשים אחרים. בנוסף לכך, אם המשתמש הוא אדמין, או שואל השאלה, הוא מורשה למחוק את השאלה במסך זה.



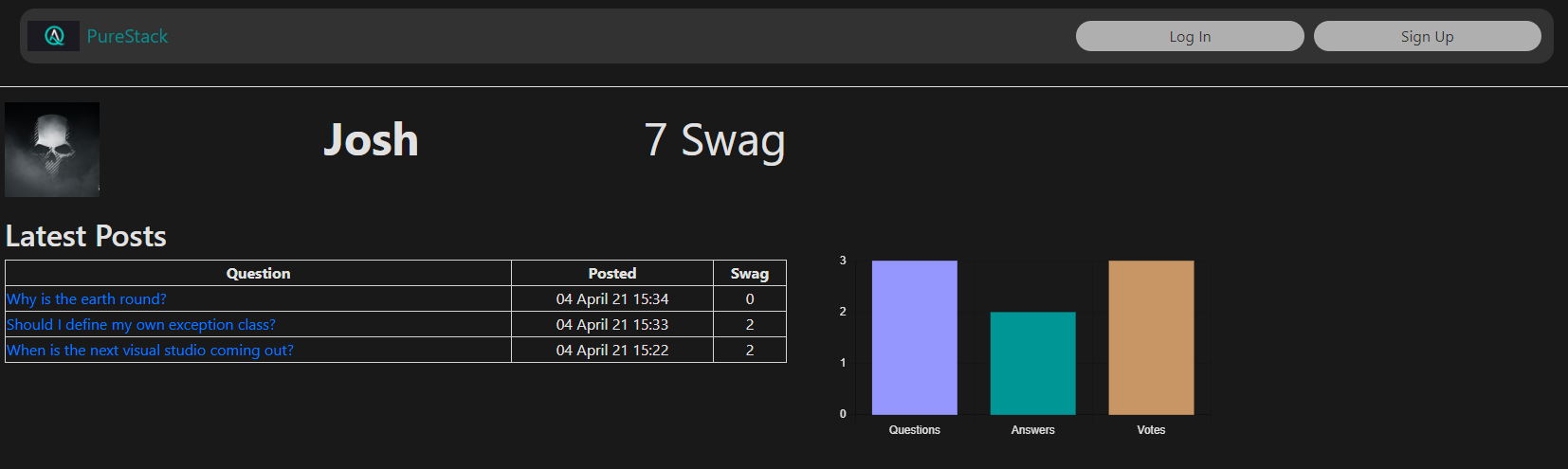
משתמשים יכולים לצפות בשאלות ותשובות גם כאשר הם לא מזוהים במערכת, אך אינם מסוגלים להצביע או לענות על השאלה.



## פרופיל משתמש

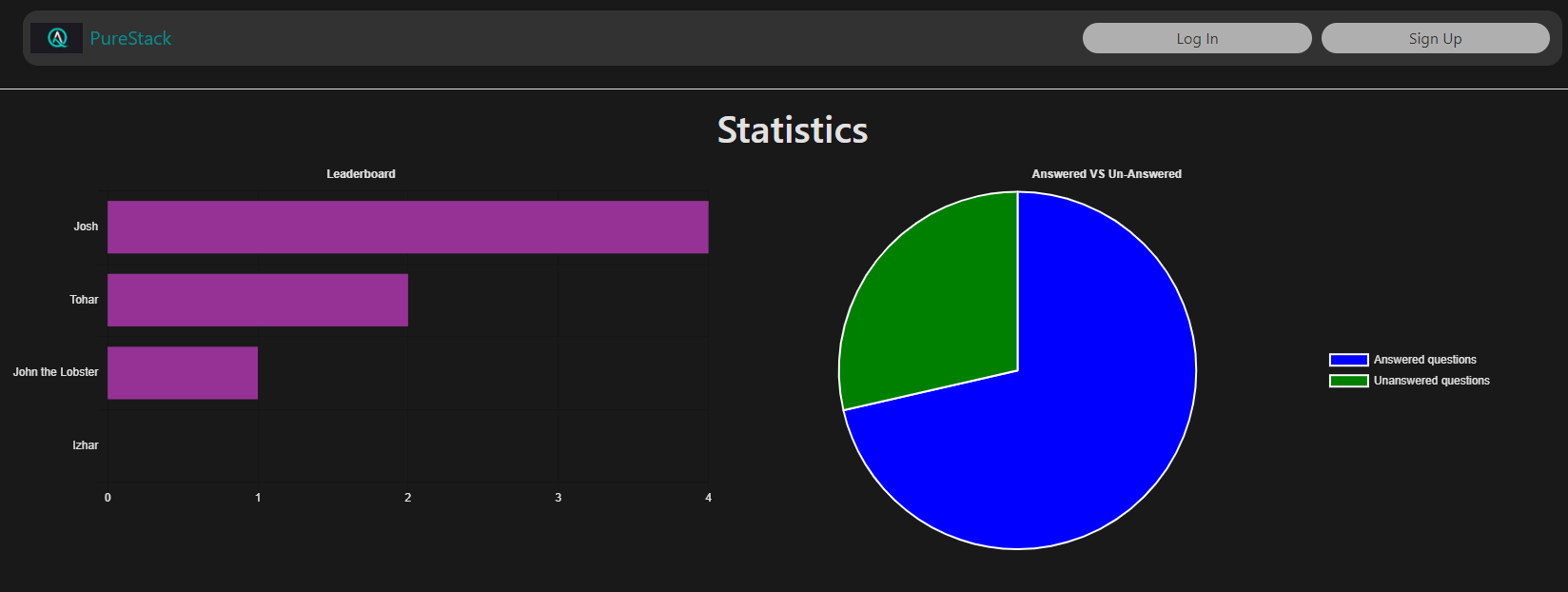
בדף זה משתמשים יכולים לצפות בפרטים של משתמשים אחרים שרשומים במערכת. הדף מציג:

* פרטי משתמש
* פוסטים אחרונים של המשתמש
* סטטיסטיקות של המשתמש



## סטטיסטיקות

דף המראה סטטיסטיקות לגבי האתר, כגון טבלת המובילים בSwag Points, וסטטיסטיקות של כמות השאלות שנענו מתוך כלל השאלות.



# סוגיות נבחרות

## מימוש הצבעה

בהרבה מקומות במערכת ניתנת למשתמש אופציה להצביע על תשובות ושאלות באופן הבא:



ההצבעה הזו צריכה להיות קלה ופשוטה וחשוב לבצעה בלי לטעון דף מחדש.

כדי להשיג תוצאה זו השתמשתי בJS ב XMLHttpRequest כדי לשלוח בקשה לשרת. לצורך חשיפה של web api בצד של השרת יצרתי Razor Page מיוחד (Actions) אשר מתפקד בתור קונטרולר פשוט כאשר הפעולה הנדרשת מזוהה על ידי Razor Handler.

## תצוגת סטטיסטיקה

לצורך תצוגת הסטטיסטיקה התמודדתי עם שתי סוגיות:

1. שליפות מורכבות ממסד
2. תצוגה גרפית

כדי לבצע שליפות יעילות בחרתי לממש מתודות ייעודיות בData access layer כדי להעביר חישוב של כל הסטטיסטיקה למסד הנתונים ולחסוך בכח ביצוע וזיכרון של שרת web.

להלן שאילתות לחישוב הסטטיסטיקה

חישוב של המשתמשים המובילים בSwag Points

SELECT u.username, q.user\_id, SUM(IF(good\_bad=1, 1, -1)) as score FROM forum.questions q

INNER JOIN forum.votes v ON q.question\_id = v.thread\_id

INNER JOIN forum.users u ON q.user\_id = u.user\_id

where v.is\_question=1

GROUP BY q.user\_id

ORDER BY score DESC;

‏חישוב כמות שאלות שנענו מכלל השאלות באתר

SELECT IF(isnull(a.question\_id), 'Not Answered', 'Answered') as IsAns, count(distinct q.question\_id) FROM forum.questions q

LEFT OUTER JOIN forum.answers a ON q.question\_id = a.question\_id

GROUP BY IsAns

ORDER BY IsAns;

לצורך תצוגה גרפית השתמשתי בספריית JS open source בשם Chart.js

## ניהול Sessions

כדי לממש הזדהות של המשתמשים השתמשתי במנגנון הsession management של asp.net. לאחר הזדהות המשתמש אני רושם את השם והמפתח המזהה של המשתמש בתוך session. דבר זה מהווה סימן לכך שהמשתמש מזוהה במערכת וזה מקנה לו הרשאות לפעולות נוספות.

את כל הלוגיקה של ניהול הsession מימשתי במחלקת SharedModel שהיא מחלקת בסיס לכל מחלקות מודלים של שכבת הפרזנטציה. דבר זה מאפשר אבסטרקציה של המימוש בצורה נכונה.

## מראה דפים עקבי

האתר שלי כולל מספר רב של דפים אך חשוב שתיהיה לו תצוגה עקבית שגם קל לערוך אותה במקום מרכזי. כדי להשיג זאת השתמשתי במנגנון layout של razor.

## העלאת תמונות

לצורך ניהול התמונות השתמשתי בתמיכה מובנת בRazor. בחרתי במימוש הפשוט שבו אני שומר את קבצי התמונה על הדיסק של שרת האתר. המימוש נמצא במחלקה SignUpModel.