סעיפים לפרויקט גמר – סרגיי ומיכאל

1. **AcademIQ Hub, Universifocus**

**מערכת לניהול הלמידה**

1. **תיאור ורקע כללי:**
   1. AcademIQ הוא שותף ייחודי לסטודנטים באקדמיה, מספק פלטפורמה מתקדמת המשפרת את חוויית הלימוד וניהול הזמן. המערכת מאפשרת לסטודנטים להתמקד בלימודים ולהימנע מלחץ המועדים והמשימות הקרובות. עם AcademIQ, המסלול האקדמי מוזקק לחווית לימוד מקיפה ומאורגנת, מאפשרת לארגן מטלות ודרישות, ומקדמת פוקוס והצלחה בלימודים.
   2. מטרות המערכת:
      1. תזכורות מותאמות אישית למשימות ולמועדי הגשה, לסיוע בניהול הזמן ובמניעת שכחת משימות חשובות.
      2. איסוף מטלות כולל וארגון מערכתי שלהן לפי תאריכי הגשה ולפי קורסים, על מנת לספק מבט מסודר וממוקד על המשימות הפתוחות.
      3. מעקב וניהול מערכת הלימודים והמבחנים, עם ניתוח פרסונלי של התקדמות הסטודנט והמלצות לשיפור.
      4. ספריית מקורות אישית ושיתוף פעולה עם סטודנטים אחרים, על מנת לאפשר ניהול חומרי לימוד ושיתוף פעולה בצורה מאובטחת ומקיפה.
      5. צאט כיתתי לתקשורת ושיתוף פעולה בין הסטודנטים במסלול.
      6. כלים לניהול הזמן והתמקדות, כולל טיימר ללימוד, תזמון הפסקות, והמלצות לפעולות שיפור ריכוז בזמן הלמידה.
2. **המצב בשוק והבעיות:**

יתרונות

חסרונות

עוד אחד

יתרונות

חסרונות

בשוק קיימת מערכת "ידיעון" המשמשת סטודנטים ומרצים במוסדי לימוד. מערכת זו מספקת שירותים כגון פרסום והגשת מטלות, ניהול נוכחות, הודעות, ולוח שנה. בנוסף למערכת הידיעון, קיימים בשוק כלים נוספים בצורת אתרי אינטרנט ואפליקציות המציעים כלים לניהול זמן ויעילות.

* 1. הבעיות במצב הקיים

המערכת הקיימת מאופיינת בחוויה לא אינטואיטיבית וממשק לא אטרקטיבי, המונע שימוש נרחב ויעיל בכלים הזמינים לסטודנטים. בנוסף, נתקלים בבעיות כגון:

* אי־סידור המטלות לפי זמנים, דחיפות וקורסים, המפריע לניהול מדויק ויעיל של הלימודים.
* חוסר הסנכרון בין לוח השנה לבין מערכת השעות, מה שיכול לגרום בלבול בקביעת זמנים ומטלות.
* חוסר רלוונטיות של התראות שאינן ניתנות לסינון או ביטול, מה שמפריע לניהול יעיל וברור של המשימות והדרישות.

1. מה הפרוייקט אמור לחדש או לשפר

מטרת הפרויקט היא לשפר את הממשק הקיים במערכת, הן ביצירתיות והן ביעילות, על מנת להתאים אותו בצורה מדויקת יותר לצרכי הסטודנטים. באמצעות הפרויקט תוכננו השיפורים הבאים:

* שדרוג ויזואלי של הממשק כך שיהיה נעים לעין ומעניין לשימוש.
* התאמה מדויקת יותר לצרכי הסטודנטים על ידי יכולות ניהול וסינון מורחבות (קסטומיזציה) של הממשק.
* הוספת כלים שיבטיחו נוחות ויעילות לסטודנטים, כגון ניהול לוחות זמנים ולוחות שנה יעילים ופונקציות ייחודיות כמו יצירת קשר עם גורמים, כלים ועצות ללמידה (כגון טיימר "פומודרו").

1. **דרישות פונקציונאליות / מערכת**
   1. **פונקציונאליות**

**סטודנט**

מערכת שעות

העלאת חומר מצולם

צפיה בהודעות

מצב למידה

צאט

כתיבת הודעה למרצה

הגשת מטלות

**מרצה**

מערכת שעות

העלאת חומרי לימוד

כתיבת הודעה לקורס

העלאת מטלה

בדיקת מטלה

כתיבת הודעה לסטודנט

**מזכירה**

בניית מערכת שנתית

עדכון מערכת שעות

**אדמיניסטרטור**

תחזוקת טבלאות

שדרוג גירסאות

**טיימר**

חישוב זמן פעילות (להתנתקות)  
תזכורת על הודעות שלא נקראו  
תזכורת מטלות פתוחות  
פומודורו

* + 1. המערכת תעדכן מטלות מתקבלות לפי סדר הגשה ולפי קורס, מספקת ייצוב ואמינות בעבודה עם מטלות ושעות שיעור.
    2. לוח השנה יהיה מסונכרן עם שעות השיעורים והמטלות לאותו קורס, מספק עדכון ותיאום יעיל של פעילויות לימודיות.
    3. לכל קורס יהיה דף מיוחד המציג את כל המטלות ציונים/ הישגים וחומרי הלימוד מסודרים לפי נושאים, מאפשר ניהול אישי ושמירת חומרים
    4. פתרון צ'אט כיתתי יאפשר לתלמידים לשתף חומרי לימוד ולתקשר בינהם בקלות ובנוחות.
    5. המערכת תספק סביבה מאובטחת שמשמרת על פרטיות המשתמשים ומבטיחה את שמירת חומרי הלימוד והמידע בצורה מאובטחת.
    6. מערכת שעות שבועית המציגה שיעורים לשבוע כולל מטלות נדרשות
    7. במידה וישנו שיעור מקוון באוויר, המערכת תודיע על כך ותכווין בלחיצה לדף הקורס הספציפי שבו יופיע הכפתור שמוביל לשיעור המקוון, הכפתור יופיע רק במידה וישנו שיעור באוויר.
    8. דף "מצב למידה", קיצורי דרך שימושיים, טיימרים, טיימר פומודורו, וכל מה שהסטודנט צריך ללמידה ארוכה ומרוכזת.
  1. **דרישות מערכת**
     1. יש לממש מערכת המספקת את כל הפונקציות הנדרשות ולהבטיח יציבות וביצועים אופטימליים.
     2. המערכת צריכה להיות ידידותית למשתמש ולספק חווית משתמש משופרת.
     3. יש לפתח מנגנונים לניהול ותחזוקה של המערכת, כולל תמיכה טכנית ועדכונים רציפים.
     4. נדרשת אפשרות להתאמה והרחבה עתידית של המערכת בהתאם לצרכי המשתמשים ולדרישות השוק.

1. **בעיות צפויות במהלך הפיתוח ופתרונות**

6.1. **תיאור הבעיות:**

* אבטחת מידע: שמירה על integrety של הנתונים והגנה על פרטיות המשתמשים היא דרישה חשובה בפרויקט.
* UI ו UX: יצירת ממשק משתמש אינטואיטיבי ומעניין לידיעה היא משימה אתגרית.
* תאימות למכשירים: הצגת המערכת בצורה יעילה ומותאמת למכשירים שונים (מחשבים ניידים, טלפונים חכמים, טאבלטים) היא חשובה.

6.2. פתרונות אפשריים

אבטחת מידע:

* JWT: השתמש ב JWT (JSON Web Token) לאימות המשתמשים והגנה על נתונים.
* הצפנה: הצפנת נתונים רגישים ב המחשב או ב השרת.

UI ו UX:

* עיצוב ממשק משתמש: השתמש ב עקרונות עיצוב UI ו UX מקובלים כדי ליצור ממשק אינטואיטיבי ותומך.
* בדיקות משתמשים: בצע בדיקות משתמשים לפני השקת המערכת כדי לקבל משוב ולשפר את חוויית המשתמש.

תאימות למכשירים:

* Responsive Design: השתמש ב Responsive Design כדי להציג את המערכת ב צורה מתאימה ל מכשירים עם גודל מסך שונה.

1. **פתרון טכנולוגי נבחר**

7.1. **טופולוגית הפתרון:**

* **שרת: שרת ASP.NET Core עם Entity Framework Core (EF Core) ל ניהול ה DB SQL Server.**
* **MSSQL Server ל אחסון נתונים.**
* **DB MongoDB: MongoDB ל אחסון נתונים עבור צאט כיתתי.**

**7.2. טכנולוגיות בשימוש:**

* **JWT: JSON Web Token ל אימות המשתמשים והגנה על נתונים.**

**7.3. שפות הפיתוח:**

* **JavaScript: שפת פיתוח עבור האתר בReact .**
* **C#: שפת פיתוח עבור שרת ASP.NET Core.**
* **SQL: שפת פיתוח ל ניהול ה DB SQL Server.**
* **TypeScript: שפת פיתוח ל client-side צאט (MongoDB).**

**7.4. תיאור ה ארכיטקטורה הנבחרת:**

* **Client-Server Architecture: אתר האינטרנט React (client) מתקשרת אם שרת ASP.NET (server) ל קבלת נתונים ו ביצוע פעולות בבסיס בנתונים.**
* **REST API: שרת ASP.NET Core מספק RESTful API לאתר בReact לשיתוף נתונים.**
* **Entity Framework Core משמש לניהול הSQL Server.**
* **Auth and Security: JWT משמש לאימות המשתמשים והגנה על הנתונים במערכת.**

7.5.

AcademiQ

אדמין

תחזוקת טבלאות

שדרוג גירסאות

טיימר

צפיה בזמן פעילות

תזכורת על הודעות שלא ניקראו

תזכורת על מטלות פתוחות

מזכירות

בניית מערכת שנתית

עדכון מערכת שעות

מרצה

מערכת שעות

העלאת חומרי לימוד

כתיבת הודעה לקורס

העלאת מטלה

כתיבת הודעה לסטודנט

סטודנט

צפייה במערכת שעות

העלאת חומר מצולם

צפיה בהודעות

מצב למידה

צאט

כתיבת הודעה למרצה

**7.6.?**

**7.7. ממשק המשתמש/לקוח – GUI:**

* **React: לפיתוח ממשק משתמש אינטואיטיבי.**
* **User Experience: התמקדות בחוויית משתמש נוחה.**

**7.8. ממשקים למערכות אחרות/API:**

* **RESTful API: שרת ASP.NET Core מספק RESTful API להתקשרות עם בבסיס הנתונים.**
* **Authentication: אימות JWT להתחברות למערכת וכבלת שירות מהשרת.**

**7.9. שימוש בחבילות תוכנה:**

* **MaterialUI: שימוש בספריות עיצוב MaterialUI לעיצוב ממשק המשתמש.**
* **Entity Framework Core: חבילת ASP.NET Core לניהול הDB.**
* **ASP.NET Core JWT: חבילת ASP.NET Core לאימות המשתמשים.**

1. שימוש במבני נתונים וארגון קבצים
   1. נירמול:
   2. מבני הנתונים:

בבסיס הנתונים העיקרי של AcademIQ הוא בבסיס הנתונים טבלאי SQL Server .

במערכת AcademIQ ישנו צאט כיתתי הממומש על ידי MongoDB.

שיטת האחסון

אחסון מבוסס-עמודים:

יתרונות: מהירות, יעילות, תמיכה חזקה בשלמות נתונים.

חסרונות: מורכבת יותר מפתרונות קבצי, יקרה יותר.

דוגמאות: SQL Server.

אחסון מבוסס-שורות:

יתרונות: טובה לעיבוד שורות בודדות, יעילה לחיפושים ספציפיים.

חסרונות: פחות יעילה לעיבוד נתונים גדולים, פחות גמישה.

דוגמאות: MongoDB.

* 1. מנגנוני התאוששות

גיבויים: גיבויים סדירים של הנתונים יאפשרו את שחזורם במקרה של אובדן הנתונים.

עסקאות: עסקאות (transactions) ב-DBMS הן מנגנון שהבטיח שכל הנתונים יעודכנו במקביל וללא שגיאות.

Journaling:

זהו מנגנון מרכזי להתאוששות ב MongoDB.

תיעוד של כל שינוי שעובר על הנתונים, כמו הוספה, עדכון או מחיקה.

במקרה של קריסה, MongoDB יכול להשתמש ביומן כדי לשחזר את הנתונים למצב עקבי לפני התקלה.

1. תרשימי מערכת מרכזיים

9.1. Use Case

9.2. Sequence Diagram

9.3. Data Flow

1. תיאור המרכיב האלגוריתמי - חישובי
   1. **הבעיה שבנו לפתור:?**
   2. **איסוף מידע וניתוחים סטטיסטיים**

**הישגי קורס מסוים - בדיקה של זמני הגשת מטלות/ פרויקטים לפני מועד ההגשה.**

**נוכחות קורסים בזמנים שונים בסמסטר**

**ממוצע כיתתי קורסי**

**ממוצע תלמידים פרטני**

**ממוצע זמן שימוש בטיימר פומודורו בזמן למידה כולל ביום**

**11. אבטחת מידע**

**אימות משתמשים (Authentication):**

* **המערכת משתמש בJWT (JSON Web Token) לאימות המשתמשים בגישה לנתוני המערכת.**
* **כל משתמש צריך להכנס עם שם משתמש וסיסמה ייחודיים לקבלת גישה למערכת.**

**הרשאות (Authorization):**

* **ממומשת במערכת הרשאות להגבלת גישה לנתונים ושירותים עבור משתמשים שונים.**
* **לדוגמה, מרצים יכלו לראות את הנתוני הקורסים שהם מובילים בלבד.**

**הצפנת נתונים:**

* **נתונים רגישים במערכת, כגון סיסמאות, יהיו מוצפנים באמצעות אלגוריתמי הצפנה מתקדמים.**

**12. משאבים הנדרשים לפרויקט**

**12.1. מספר שעות המוקדש לפרויקט, חלוקת עבודה בין חברי הצוות**

**500 שעות אדם:**

**50 שעות עפיון**

**50 שעות הקמת בסיסי נתונים**

**270 שעות פיתוח צד לכוח**

**130 שעות פיתוח צד שרת**

**12.2 ציוד נדרש:**

**טלפון, מחשב, אינטרנט**

**12.3 תוכנות נדרשות:**

**VSCode, Visual Studio, MSSQL, MongoShell, Expo**

**12.4 ידע חדש שנדרש ללמוד לצורך ביצוע הפרויקט:**

**פיתוח אפליקציות נטיביות ב React-Native**

**הקמת צד שרת בASP.NET**

**הקמת צד שרת בNode.js**

**12.5 ספרות ומקורות מידע**

https://stackoverflow.com

https://www.w3schools.com

https://dotnet.microsoft.com/en-us/apps/aspnet

https://chatgpt.com

https://www.mongodb.com

https://aistudio.google.com

https://reactnative.dev

https://expressjs.com

https://expo.dev

13. תכנית עבודה ושלבים למימוש הפרויקט

?

1. בדיקות

14.1. בדיקות תהליכיות

* אינטגרציה בין באתר בReact לשרת ASP.NET Core:  
  בדיקת התקשורת בין הclient ו הserver באמצעות REST API.
* אינטגרציה בין השרת ASP.NET Core לSQL Server:  
  בדיקת ההתקשורת בין הserver ל הDB SQL Server.

14.2. בדיקות יחידה

* בדיקת פונקציונאליות של Components בReact.
* בדיקת פונקציונאליות של Controllers בשרת ASP.NET Core.

1. בקרת גרסאות

?

1. קטעי קוד 