**סקירת ספרות**

**StudyBuddy**



**הסקירה נכתבה על ידי:**

**יבגני בלנקי 319323051**

**הדר עטיה 205518392**

תוכן

[סקירת ספרות 3](#_Toc60659619)

[*1.1 מבוא* 3](#_Toc60659620)

[*1.2 תיאור המערכת* 3](#_Toc60659621)

[1.2.1 כללי 3](#_Toc60659622)

[1.2.2 טכנולוגיות 4](#_Toc60659623)

[*1.3 תחרות* 5](#_Toc60659624)

[*1.4 אבטחת היישום* 7](#_Toc60659625)

[*1.5 מילון-מונחים* 7](#_Toc60659626)

[*1.6 ביבליוגרפיה* 7](#_Toc60659627)

# סקירת ספרות

## 1.1 מבוא

כיום עם התפתחות הטכנולוגיה, סטודנטים נעזרים באמצעים טכנולוגיים במהלך לימודיהם על מנת לשתף לרכוש ולנהל קבצים, בין אם מדובר בשיתוף סיכומים, עבודות, פתרונות לתרגילים, מבחנים, רשימת הרצאות ודרכי תקשורת עם מרצים. האמצעים שעומדים בפני הסטודנטים הם לרוב קבוצות וואצאפ, גוגל דרייב או כלי דומה לניהול קבצים ושיתופם ומודל. במהלך לימודינו, ובמיוחד בתקופת הקורונה, גילינו שהכלים האלה הרבה פעמים לא מספקים את כל צרכינו מכיוון שאינם ייעודיים לסטודנטים ובנוסף עלינו הסטודנטים ללמוד להשתמש בכל אותם כלים כדי שנוכל להשתמש ולנצל את שירותיהם ביעילות.

מכאן עלה הרעיון ל-StudyBuddy, יישום אינטרנטי מבוסס ענן עבור סטודנטים שייצור חווית למידה משותפת טובה ויעילה על ידי כך שירכז את הכלים הדרושים על מנת לשתף את כל המידע הדרוש לסטודנטים ובנוסף יאפשר להם לתקשר אחד עם השני.

בנוסף בגלל שהיישום מרכז את כל הכלים שמספקים יישומים אחרים והיישום מותאם עבור הסטודנטים מבחינה ויזואלית ופונקציונלית ,זה מאפשר לסטודנטים ללמוד בקלות ולהשתמש ביעילות בשירותי היישום שלנו ללא צורך בהפניות או

למידה נוספת של כלים נוספים.

## 1.2 תיאור המערכת

### 1.2.1 כללי

אתר הבית יכלול מידע כללי אודות היישום ואפשרות התחברות או הרשמה.

לאחר ההתחברות, משתמש יוכל לחפש משתמשים ע"י שם משתמש על מנת לצפות בפרופיל שלו, להוסיף אותו או לבקש להצטרף לקבוצה-ציבורית שפתח.

בנוסף יהיה ניתן לחפש קבוצות בעלי הרשאת - ציבורי(לדוגמא 'הנדסת תוכנה – סמי שמעון באר שבע')ולבקש להצטרף לקבוצה.

בקבוצה יהיה ניתן לקבל גישה לקבצים השייכים לקבוצה, לקבוצות הצ'אט ולסטודנטים שנמצאים בה.

הקבצים יכולים להיות תיקיות (למען סיווג לשנת לימוד או קורס ספציפי) וקבצי טקסט או תמונות. עבור כל קובץ יהיה שדה תגובות שבו ניתן יהיה לדון בתוכן הקובץ ומנהלי הקבוצה יהיו רשאים להעלות, למחוק ולשנות את הקבצים בקבוצה.

בעלי הרשאות יוכלו לפתוח קבוצות צ'אט(למשל לפי סיווג לשנת לימוד או קורס ספציפי)אליהם משתמשים יוכלו להצטרף.

בנוסף, משתמשים יהיו חשופים לחברי הקבוצה מה שיאפשר להם לצפות בפרופילים שלהם, להוסיף אותם ולשלוח הודעות אחד לשני וכך לתקשר בפרטיות במסגרת המערכת.

הקבצים שיועלו למערכת יהיו מאוחסנות בענן ולכן למשתמשים תהיה גישה אליהם מכל מקום, מה שיקנה לסטודנטים את הגמישות ללמוד גם כשאינם בבית או במוסד האקדמי.]דיאגרמה 1]

### 1.2.2 טכנולוגיות

Cloud:

לצורך הפרוייקט שלנו, אנו נשתמש בטכנולוגיית ענן.

'ענן' אומר שהמידע של המערכת ישב בשרת מרוחק, לכן משתמשים יוכלו לגשת

לכל המידע של המערכת מכל מקום ומבלי צורך בזיכרון מקומי ובמגבלה על שטח האחסון שברשותם.

הפרויקט יבנה באמצעות MERN Stack.

MERN – ראשי תיבות של **M**ongoDB, **E**xpress, **R**eact, **N**odeJS

הוא אוסף טכנולוגיות פופולרי לפיתוח בסביבת Web המספק את כל ה-Frameworks הנדרשים לפיתוח אפליקציית Web מודרנית.

יתרון גדול נוסף הוא שפיתוח בכל הטכנולוגיות מתבצע בשפה אחת, JavaScript,

מה שמקל ומזרז את תהליך פיתוח היישום על ידי צוות הפיתוח שלא נדרש ללמוד שפות נוספות דבר שיכול לקחת זמן ולעכב את התהליך. [1]

MongoDB:

MongoDB הוא בסיס נתונים מסוג NoSQL, זאת אומרת שבניגוד לבסיסי נתונים אחרים שעובדים עם טבלאות הוא נשען גם על פורמט מסוג JSON ובנוסף הוא מאפשר לשמור את המידע בענן.

MongoDB נותן פתרון סקאלבילי, מהיר ובעל יכולת לבצע שאילתות מורכבות על בסיס נתונים גדול.

בפרויקט שלנו נשתמש בכלי זה מפני שהמערכת אמורה לתת מענה למספר גדול של משתמשים ולשמור את הנתונים הרלוונטיים בצד שרת(בין היתר, פרטי משתמשים, רמת הגישה שלהם למערכת וקבצים), מה שייצור בסיס נתונים ענק הדורש פתרון סקלאבילי. [1] [4]

Express:

אקספרס היא Framework קוד פתוח המיועד לעבודה עם NodeJS המאפשרת פיתוח בצד שרת אשר חוסכת ומפשטת הרבה "עבודה שחורה", הופכת את הקוד להרבה יותר קריא ומובן וכתוצאה מכך משפרת את יכולת התחזוקה של האפליקציה. [1]

NodeJS:

NodeJS היא framework לפיתוח צד שרת בשפת JavaScript.

היא פופולרית מאד בשוק, יש לה מגוון גדול של ספריות עשירות עם דוקומנטציה טובה ולכן מאד נוחה למפתחים. בנוסף, יש לה קהילת משתמשים גדולה ופעילה שעוזרת אחד לשני, דבר המקל הפיתוח.

בפרויקט שלנו נשתמש בכלי זה כדי שניצור שרת משלנו ועל ידי כך נוכל לשלוט בבסיס הנתונים אליו יהיה מחובר, במידע שנמצא אצלו ובהודעות שמקבל מהלקוח ושולח ואליו. [1]

React:

ספריית קוד פתוח JavaScript שפותחה על ידי פייסבוק ונועדה לספק ממשקי משתמש.

מכיוון שאנו רוצים אפליקציה אינטראקטיבית, ריאקט היא המועמדת האולטימטיבית לפיתוח בצד לקוח. בזכות העובדה שניתן לעדכן קומפוננטות בודדות בניגוד לרינדור כל העמוד וספריות צד לקוח עשירות, ריאקט נותנת חוויה מעולה גם למשתמש וגם למפתח.

ולכן בפרויקט שלנו נשתמש בריאקט כדי לבנות את כל צד הלקוח. [1]

בנוסף נשתמש בצד הלקוח ב css ,react bootstrap לצורך עיצוב המערכת.

react bootstrap:

זוהי ספריה שמאגדת בתוכה קומפוננטות React על מנת לחסוך זמן הכרוך בעיצוב ותכנון של React Components.

CSS:

CSS הוא אוסף כללים שמגדיר מראה של אלמנטים ב-HTML. שימוש בכללים אלה עוזר לנראות קונסיסטנטית של האלמנטים המוצגים בדף.

## 1.3 תחרות

התחרות שלנו כרגע היא בעיקר מול שירותי ענן מובילים כמו Google Drive, , MEGA,Dropbox.

ושירותים כמו WhatsApp ו Moodle

Google Drive:

אמנם הדרייב של גוגל מאפשר למשתמשים לשתף קבצים, אבל אין בו את האפשרות ליצור דיון על מנת ליטוש ושיפור התוכן ולכן, לפחות מהניסיון שלנו בתור סטודנטים, הסיכומים בו הרבה פעמים מלאים בטעויות, הפתרונות לתרגילים חלקיים או לא נכונים ולכן הצורך ביצירת דיון לדעתנו הוא חשוב ביותר.

בנוסף, ברגע שבעל הדרייב מסיים את לימודיו, אין ביכולו "להעביר את השרביט" לדור הבא של הסטודנטים מבלי לוותר על חשבון הגוגל שלו, דבר אשר מקטין את תוחלת החיים של הדרייב בצורה משמעותית.

דבר נוסף הוא שבדרייב של גוגל ישנה בעיה של פרטיות.

כחלק מתנאי השימוש בגוגל דרייב, אנחנו נותנים לגוגל את ההרשאה לסרוק ולנתח את הקבצים שאנו מעלים כדי ליצור עבורנו "פרופיל משתמש" ובנוסף גוגל רשאי, בין היתר, "לפרסם בפומבי" כל דבר שעולה לדרייב .וכמשתמשים אנו מעדיפים לשמור על פרטיותנו .

Other Drives:

מלבד Google Drive, קיימים בשוק עשרות שירותי Cloud נוספים.

שמבחינת פונקציונליות זהים מאוד ל Google drive אבל קיימים כמה הבדלים ביניהם.

נציג כאן כמה שירותים מובילים ואת ההבדלים הבולטים ביניהם:

DropBox – בהשוואה לגוגל דרייב, מציעים נפח אחסון נמוך יחסית (2 GB). בנוסף, DropBox סובלת מבעיית אבטחה כי אינה מיישמת הצפנת E2E.

MEGA – שירות CLOUD שנותן נפח אחסון גדול (50 GB), הצפנת E2E, וגם Cross Platform (Windows, Linux and MAC). למרות היתרונות, השירות איטי, סובל מבאגים ועדיין לא עונה על הדרישה לשיפור תוכן ע"י יצירת דיון סביב התוכן והגדרת הרשאות שונות למשתמשים.

Microsoft OneDrive – השירות של מיקרוסופט דומה מאד לשירות של גוגל, אבל סובל מאותם בעיות כמו MEGA.

Moodle:

המודל מאפשר שיתוף קבצים ויצירת 'פורום' לדיון, אבל הכוח הזה לא נמצא אצל הסטודנטים מכיוון שרק סגל ההוראה מחליט על התוכן באתר.

WhatsApp:

אפליקציית WhatsApp מאפשרת לסטודנטים לתקשר אחד עם השני באופן מידי ולחלוק רעיונות וידע בכתב ובנוסף לשתף קבצים. אולם זה עוד כלי נפרד וניהול הקבצים בו מאוד לא נוח לסטודנטים.

הפתרון שלנו StudyBuddy יעניק לסטודנטים את הכוח לשלוט בתוכן בעצמם בניגוד ל Moodle ,ובכך לספוג גם את הידע שצברו סטודנטים משנים קודמות ולבנות עליו על מנת לשפר את איכות המידע ממנו הם לומדים.

לגבי שירותי ה cloud שפורטו מעל ואלו שלא, לכולם יש יתרונות וחסרונות משלהם שבעיקר מתרכזים בנפח אחסון, אבטחה, שירות וביצועים. אבל מהסקר שערכנו לא מצאנו אף אחד שנותן את האפשרות לתת הרשאות מותאמות ולפתוח דיון בין משתמשי הענן על התוכן הקיים. ולכן ב StudyBuddy אנחנו נותנים את המענה לכך.

לגבי הבעייתיות עם WhatsApp אנו נותנים שירות בעל פונקציונליות זהה ועל ידי כך שמיושם במערכת שלנו עם עוד שירתים שונים שהכנסנו אנו מייצרים מערכת אחת ויחידה שעונה על צרכיהם של הסטודנטים.

## 1.4 אבטחת היישום

bcrypt – אלגוריתם הצפנה (hash) מבוסס על Blowfish Algorithm והינו אחת האופציות הטובות ביותר להצפנת סיסמאות כיום. [3]

אנו משתמשים בפונקציית bycrypt לצורך הצפנת סיסמאות המשתמשים ב database .

וזאת כדי שגם אם ייפרצו למערכת ,לתוקפים לא יהיו את הסיסמאות האמתיות של המשתמשים של המערכת דבר שמגן עליהם.

JWT – מכיוון שאנו משתמשים ב-MongoDB כבסיס הנתונים שלנו, JSON Web Token נראה כמו הפתרון המתאים ביותר היות ו-MongoDB מתקשר ע"י שליחת הודעות בפורמט JSON.

אנחנו משתמשים בJWT עבור ההזדהות במערכת מה שמאפשר לנו לשמור session התחברות בטוח ומימוש קל עבור ה-Single Sign-On, שבעזרתו ניתן להתחבר פעם אחת ולשמור על ה-session "פתוח" כדי שלא יהיה צורך להתחבר כל פעם מחדש. [2]

אבל עדיין אינינו יודעים באילו כלים ושיטות נוספות נשתמש לצורך אבטחה מיטבית למערכת.

## 1.5 מילון-מונחים

|  |  |  |
| --- | --- | --- |
|  | מונח | הסבר |
| 1 | בסיס נתונים | מערכת לשמירת נתונים במחשב |
| 2 | JSON | פורמט טקסטואלי להעברת נתונים הבנוי בצורת מפתח-ערך (כמו מילון).  הפורמט קריא על ידי בני אדם. |
| 3 | Framework- תשתית תוכנה | אוסף ספריות שמיועדות למטרה מסוימת.  שימוש ב-frameworks מקל על המפתח לביצוע המשימה. |
| 4 | קוד פתוח | תוכנה שהקוד שלה פתוח לקהל הרחב, מה שנותן למפתחים את האפשרות לשנות את התנהגות התוכנה שתתאים להם יותר, לתקן באגים ולכתוב פיצ'רים חדשים ולשתף את שאר קהילת המפתחים בהם. |
| 5 | קומפוננטות | קומפוננטה הוא רכיב, פיסת קוד בJavaScript האחראי על התנהגות ספציפית בקוד. |
| 6 | רינדור | הפיכת טקסט (קוד) לתמונה. |

## 1.6 ביבליוגרפיה

1. Vasan Subramanian - “Pro MERN Stack”
2. I Putu Arie Pratama\*, Linawati, Nyoman Putra Sastra - Token-based Single Sign-on with JWT as Information System Dashboard for Government
3. Information and Password Attacks on Social Networks:: An Argument for Cryptography - Enrico Franchi, Michele Tomaiuolo, Agostino Poggi
4. Hema Krishnan, M.Sudheep Elayidom, T.Santhanakrishnan - MongoDB – a comparison with NoSQL databases