

Readme – Server Side

פרויקט אישי- חלק ג'

1. מבנה תיקיית הפרויקט

בתוך תיקיית Project_Part_C נמצאים הקבצים הבאים:

- קובץ Index.js - זהו הקובץ של הסרבר.
- תיקיית db המכילה:
 - 3 קבצי אקסל המכילים את טבלאות בסיס הנתונים: DoctorsData, Searches, Users.
 - קובץ אקסל המכיל את המידע שמוצג בעמוד aboutUs - aboutUsText.
 - קובץ createdDB - מכיל בתוכו שאילתות של יצירה, הכנסה, הצגה ומחיקה של טבלאות בסיס הנתונים.
 - קובץ CRUD_functions - קובץ המכיל בתוכו פונקציות עבור פעולות המשתמש.
 - קבצי db ו db.config.
 - קובץ funcs - קובץ המכיל פונקציה אשר מחשבת מרחק בין שתי קואורדינטות.
- תיקיית public המכילה:
 - תיקיית views הכוללת בתוכה את כל קבצי ה-pug.
 - תיקיית static הכוללת בתוכה קובץ JavaScript.js עבור פונקציות וולצידיות אשר מתבצעות בצד של ה-client, קובץ style.css עבור העיצוב ותמונות.

2. טיפול בבקשות לקוח

הפונקציות המתאימות שיצרתי לכל בקשה שמגיעה מצד לקוח:

- עבור בקשת sign up - יצרתי את הפונקציה createNewCustomer.
- עבור בקשת sign in - יצרתי את הפונקציה findCustomer.
- עבור בקשת search - יצרתי את הפונקציה findDoctors.

כל הפונקציות כתובות בתוך קובץ CRUD_functions.

3. חיבור לבסיס נתונים ושאלתות SQL

החיבור לבסיס הנתונים מתבצע בקבצים db.config, תוך שימוש ב-mysql.

בתוך קובץ createDB כלולות כל השאלתות של יצירה, הכנסה, הצגה ומחיקה של טבלאות בסיס הנתונים.

הטבלאות שנוצרות:

- טבלת משתמשים אשר ביצעו רישום לאתר.
- טבלה עבור כל הרופאים הקיימים במאגר.
- טבלה המציגה את כל נתוני החיפוש שהתבצעו באתר.

כמו כן, בקובץ createDB מתבצעת הכנסה לטבלת הרופאים, ואילו לטבלת המשתמשים והחיפושים לא, לא מכיוון שהנתונים נוצרים כאשר משתמש יוצר חשבון חדש וכאשר משמשים מבצעים חיפושים.

בנוסף, בקובץ זה ישנן שאילתות אשר **מציגות ומוחקות** את כל אחת מהטבלאות.

להלן פירוט עבור ה-routes שיש להזין בכתובת ה-url על מנת לבצע כל אחת מהפעולות (יש לבצען על פי הסדר הרשום):

- localhost:3000/CreateDoctorsTable
- localhost:3000/CreateUsersTable
- localhost:3000/CreateSearchesTable
- localhost:3000/InsertDoctorsData
- localhost:3000/InsertDataToUsers
- localhost:3000/InsertDataToSearches
- localhost:3000/ShowTableDoctors
- localhost:3000/ShowTableSearches
- localhost:3000/ShowTableUsers
- localhost:3000/dropDoctorsData
- localhost:3000/DropTableSearches
- localhost:3000/DropTableUsers

4. מימוש הטפסים

- **טופס sign in** בעמוד הפתיחה- בקשת get (action = /findCustomer). על המשתמש להזין כתובת אימייל וסיסמה. במידה והמשתמש לא קיים בבסיס הנתונים תוקפץ לו הודעה מתאימה, אחרת הוא יועבר לעמוד החיפוש.
- **טופס sign up**- בקשת post (action = /newCustomer). על המשתמש להזין אימייל, סיסמה ואישור סיסמה. במידה והמשתמש קיים כבר בבסיס הנתונים תוקפץ לו הודעה מתאימה, אחרת הוא יתווסף לבסיס הנתונים ויועבר לעמוד ההתחברות (sign in).

5. מימוש פונקציונאליות עיבוד מידע

- **טופס search**- בקשת get (action = /findDoctors). לאחר שהמשתמש יזין את פרטי החיפוש יתבצע חיפוש ב-db עבור הרופאים המתאימים לנתוני החיפוש, במידה ונמצאו רופאים מתאימים המשתמש יועבר לעמוד בו יוצגו לו פרטי הרופאים הרלוונטיים, אחרת יועבר לעמוד שיופיע לו כי לא נמצא עבורו רופא מתאים.
- פרטי החיפוש הינם: סוג שירות מבוקש ומיקום, כאשר למשתמש יש 2 דרכים להזין את פרטי המיקום שלו:
 - אופציה ראשונה: להכניס מדינה ועיר. המערכת תחפש את הרופאים הרלוונטיים לאותו סוג שירות במדינה והעיר שהוזנו.
 - אופציה שנייה: המשתמש יוכל להשתמש במיקומו הנוכחי ולהזין את רדיוס החיפוש ביחס למיקומו הנוכחי. המערכת תבדוק את המרחק בין הקואורדינטות של המשתמש לבין הקואורדינטות של כל אחד מהרופאים הרלוונטיים (בעלי סוג השירות שהתבקש), ותשלח לו את הרופאים שהמרחק מהם קטן מהרדיוס שהזין המשתמש.

6. מימוש View Engine

כל עמודי ה-html תורגמו לעמודים מסוג pug. בדרך זו, במקום ליצור קבצי html עם אלמנטים ותוכן, יצרתי תבניות המכילות משתנים ומשתנים המכילים תוכן, כך שבכל פעם מוצג התוכן הרלוונטי.

המלצות לחיפוש:

- רופא עיניים בפלורידה, אורלנדו
- רופא משפחה בכל העולם- להשתמש במיקום הנוכחי ולשים רדיוס מקסימלי של 40,000 ק"מ.