הפרויקט:

סוגיה

רשת חברתית שעוסקת בשאלות ודיונים ("סוגיות") אודות עולמות תוכן עכשוויים וסוגיות שעל סדר היום הציבורי.

הרשת מורכבת מרשימה ארוכה של "בתי מדרש",

לכל בית מדרש קבוצת משתמשים וסוגיות מוגדרות

בד בבד הפלטפורמה פתוחה גם לשימוש ציבורי תחת הפלטפורמה של בית המדרש הכללי

הפלטפורמה מופעלת על ידי עמותת "גשר מפעלים חינוכיים" כפרויקט יזמי, חברתי, רעיוני שאיננו מסחר במהותו אך עשוי להתפתח לכיוון הזה בעתיד.

הפלטפורמה פועלת כרגע תחת הדומיין "sugia.net" ועוד סאב-דומיינים של מתי מדרש נוספים למשל eco.sugia.net.

הפלטפורמה הנוכחית פותחה בסביבת קוד פתוח, PHP במערכת Wordpress, והגיעה לידי סף הכלה מקסימלי שלא מאפשר פיתוחים ייחודים בשל אילוצי המערכת.

דוגמאות :

~~~~~~~

א. לאחר בדיקה מעמיקה התברר כי אינגרציה בין בתי מדרש שונים דורשת הגדרה מחדשת של ארכיטקטורת בסיס הנתונים וגוררת גם שינויים בקבצי מערכת (CORE)

שמובילים לסיבוכיות גבוהה וחסרי יעילות.

ב. פיתוחים קלים יחסית כמו הטמעת POP-UP דורשים שינויים אגרסיביים בקוד שאינם מתאימים למערכת הנוכחית

ג. מערכת WORDPRESS אינה עובדת לפי עקרון MVC דהיינו, הפרדה בין תצוגה, בסיסי נתונים וקוד צד שרת ולכן התאמה של הפרויקט למכשירי Mobile לכדי אפליקציה היברידית, תדרוש שכתוב מחדש של המערכת

ד. בעיות PERFORMANCE (ביצועי מערכת) רבות נגרמות כתוצאה מהתערבות בלתי מבוקרת של רכיבים חיצוניים (PLUGINS) במערכת היוצרים הן עומס על שאילתות בסיס הנתונים והן על שירשור קבצי Media בצד הלקוח (קבצי js, css, תמונה וכד')

הפרויקט הנוכחי שאנו מציעות, הוא למעשה מעבר טכנולוגי בין פיתוח בסביבת LAMP

(Linux , Apache , MySQL , PHP)

לסביבת MEAN Stack -

(MongoDB , Express, Angular , Node.js )

מאפייני המערכת הטכנולוגית החדשה:

a. Server side based on node.js as a clean RESTFull API service, that includes:

\* Full user authentication and permissions controller includes access throgh external providers : Facebook, GoogleAuth API Service.

\* Set of database models based on mongoose (Object relational mapping service) that connects to either MySQL or MongoDB no-sql service (will be defined upon customer's neccesities)

\* lightweight http server based on node.js

b. client side based on angular 1.0 which consists of the following:

\* Module-based programming

\* Working with Declarite type methodology (directives)

\* Working with services and promise objects

\* Stricted Dependency Injection approach

\* Angular controllers for each view

\* Single Page App - route system with client side

אתגרים במערכת החדשה: