**בעיה ופיתרון:**

כשחשבנו על נושא לפרוייקט עלתה לנו הבעיה הבאה: אם אני רוצה לדעת על ספר היכן הסיפור מתקיים בעולם, האם יש כלי שיכול לעשות זאת? אנחנו רצינו לראות על מפה היכן הספרים האהובים עלינו מתקיימים, כלומר היכן הסיפור קורה גיאוגרפית!

לכן **החלטנו ליצור כלי** שלוקח ספרים, מחלץ מהטקסט שלהם את המקומות הגיאוגרפיים ומציג את זה בצורה נוחה ונעימה לעין על מפה. נושא זה נכנס לשדה **מדעי הרוח הדיגיטליים** בכך שהוא לוקח כמות מידע גדולה שקשורה ישירות למקצועות הומניים מסורתיים (ספרות, היסטוריה, מדע, שפה וכו..) , מעבד את מידע באמצעים טכנולוגיים של שיטות מחקר מתחום המחשבים (במקרה שלנו עיבוד שפה, ניתוח טקסט למיקום גאוגרפי וחילוץ מידע בעזרת מילות מפתח) ולאחר מכן מציגים את המידע לאחר עיבוד בצורה גרפית שיכולה לתת רבדים חדשים למידע ולמחקר, ובנוסף בחרנו לתת דגש על הצגה גרפית נעימה לעין ומושכת שגורמת למשתמש לרצות "לשחק" עם הכלי ובכך להגיע למסקנות בעצמו. בעזרת המפה שיצרנו והפילטרים ניתן להגיע למסקנות מעניינות שלא היה ניתן להגיע אליהם עד היום - כמו למשל על איזה מקומות כתבו מתחילת הספרות הכתובה והאם בתקופות מסויימות בהיסטוריה היו מקומות יותר "פופולריים" שכתבו עליהם, ולאיזה ז'אנרים הם שייכים.

**פרויקטים דומים:**

לאחר שיטוט באינטרנט מצאנו כמה בלוגים ישנים שניסו לעשות זאת וכבר לא פעילים (למשל http://www.placingliterature.com/), ועוד שירות שנותן אפשרות לחיפוש ספרים לפי מיקום אך מתייחס למיקום כללי של ספר, כלומר נותן את שם העיר או המדינה שעיקר הסיפור מתקיים בו, ולא לוקח אותך למסע סובב עולם אם הגיבור בסיפור מחליט באמצע לבקר במדינה אחרת.

לאחר שראינו שפתרונות אחרים לא עושים את מה שאנחנו רוצים, ואם כן אז עושים זאת בצורה מאוד צולעת, התחלנו לעבוד.

**מקורות מידע, אופן העבודה ועיבוד המידע בעזרת כלים (נכתב באנגלית כי משולב באתר ההצגה הסופית):**

**Data:**

Using Project Gutenberg data scraping we fetch a full book text and Author name. With this information we use OpenLibrary API to fetch additional attributes like subject, cover and author photos and more.

**Python:**

We analyze the full text of each book using 'spacy' NLP algorithm to get the named entities of each word according to its context in the text. Afterwards we are using 'geopy' geographical package to get the coordination of each given geographic named entities and decide whether its a genuine location according to the confidence of the the geographic package and some black list we created of 'spacy' common mistakes.

**JavaScript:**

Using MapBox service - that lets us upload the database into a live map and together with JavaScript coding manipulate the map using our JSON file and self-coded filters.

**DataBase:**

After fetching, scrapping and cleaning the information we made a readable .db available to download and added to the project folder.