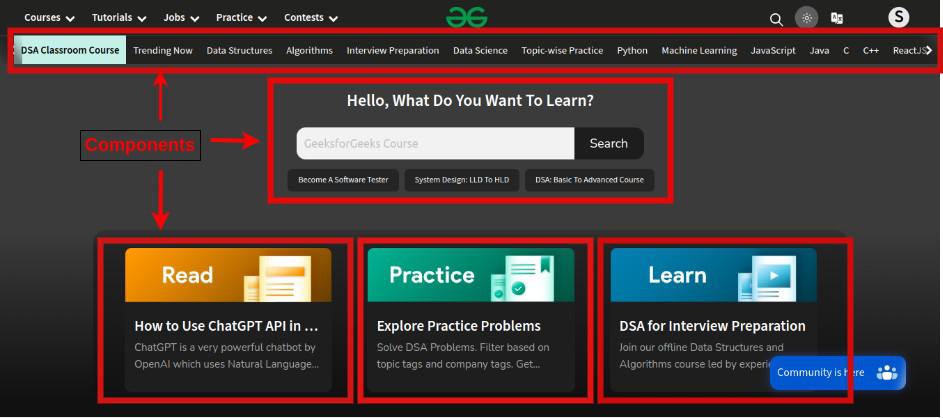
# **React**

ספרייה – Library: קובץ המכיל פקודות, פונקציות, מחלקות וכו'. ניתן להשתמש בו בפרויקט שאנו בונים לפי ראות עינינו. לדוגמה: jQuery זה ספרייה.

Framework: מערכת בתוכה אנו עובדים עם החוקים משלה והיא לוקחת את הקוד שלנו ומשנה אותו באיזשהו אופן, בכדי להביא לתוצר הסופי. לדוגמה: React זה Framework.

Component – רכיב או קומפוננטה: אבן הבניין של ריאקט. קומפוננטה זה יחידת UI נפרדת.



ניתן לבנות קומפוננטה ע"י מחלקה. זה נקרא Class Component או CC. מיושן ולא מומלץ.

ניתן לבנות קומפוננטה ע"י פונקציה. זה נקרא Functional Component או FC. זו השיטה המומלצת.

MPA: Multi-Page Application: זהו אתר אינטרנט הבנוי ממספר דפים. משתמש גולש לדף ע"י תפריט, הדפדפן גולש לשרת ומביא את הדף הדרוש. עקב כך האתר נטען מחדש. כל משתנה שיצרתי בזיכרון – נעלם.

SPA: Single-Page Application: זהו אתר אינטרנט הבנוי מדף index.html אחד. משתמש גולש לדף אחר ע"י תפריט, הדפדפן לא גולש לשרת! אלא משנה את ה-index.html בצורה דינאמית "מזריק HTML דינאמי ל-index.html). אין גלישות לשרת עבור קבלת דפים, אלא רק עבור קבלת מידע טהור.

React: Framework לבניית אתר SPA מבוסס קומפוננטות.

התקנות – מבוצעות ע"י npm:

1. התקנה של Node.js במחשב.
2. התקנת React פעם אחת ברמת המחשב: npm i -g create-react-app
3. יצירת פרויקט חדש: ב-Command-Line או טרמינל חיצוני, יש להגיע לתיקייה בה ברצוננו לבנות את הפרויקט ואז לכתוב:

create-react-app project-name --template typescript

CSS-Leaking: זליגת CSS מקומפוננטה אחת לאחרת.

שתי שיטות למנוע זאת:

1. עטיפת קוד הקומפוננטה ב-Class בעל שם הקומפוננטה, וביצוע עיצוב ב-CSS של הקומפוננטה הנמצא בתוך Class זה.
2. שימוש ב-CSS-Modules (נראה בהמשך מה זה)

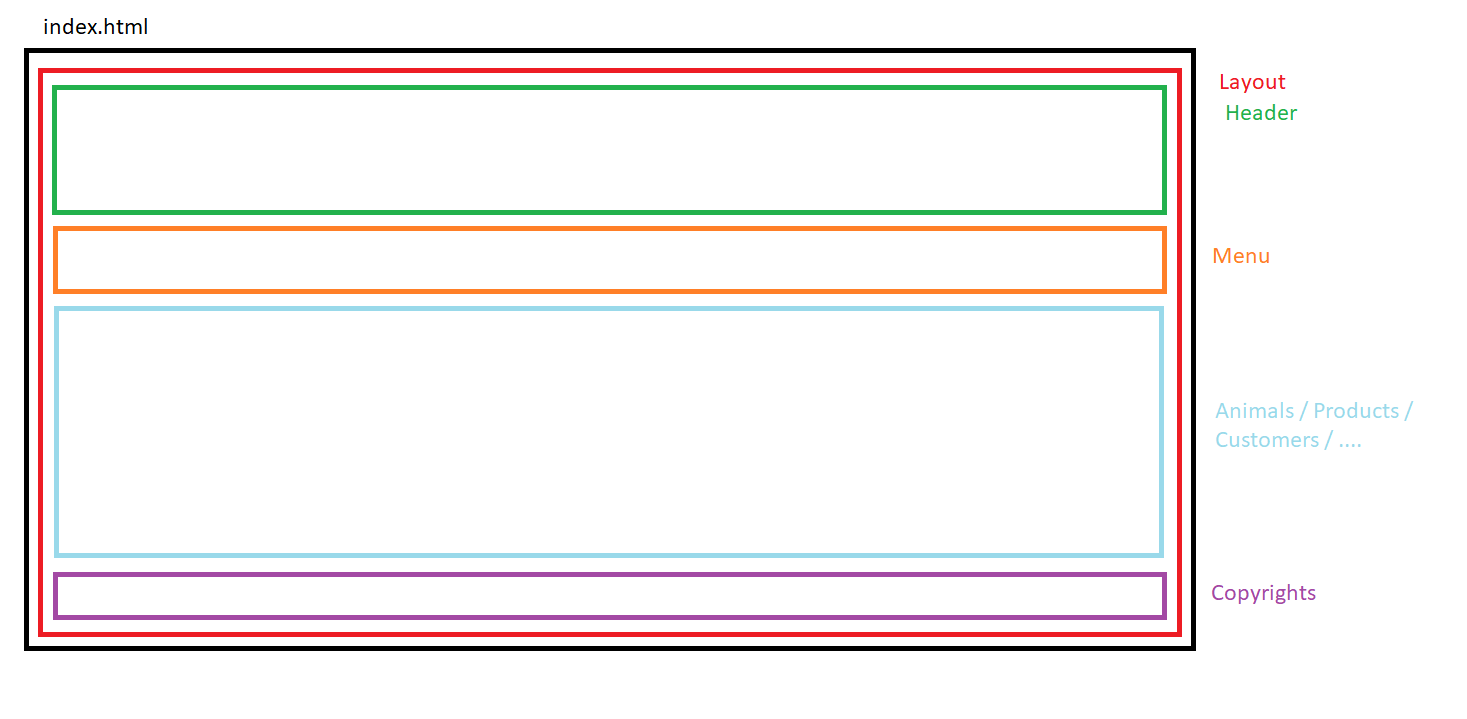
Props: זהו אובייקט ש-Parent Component יכול להעביר ל-Child Component (הוקמפוננטה שהוא מציג).

אובייקט זה נבנה ב-Child, לרוב ע"י type.

Render: הפעולה של תרגום הקוד ל-UI. הצגת ה-UI.

Conditional Rendering: הצגת UI כתלות בתנאי מסוים.

Layout: הסידור הכללי של רכיבי האתר. נבנה ע"י קומפוננטה ייעודית.



Directory Structure: זהו מבנה קבצים ותיקיות בפרויקט. בריאקט מקובל לבצע את המבנה הבא:

ProjectName

src

Assets – תמונות, סרטונים, גופנים

Images

Fonts

Components – כלל הקומפוננטות, מחולקות לאזורים

LayoutArea

Layout

Layout.css

Layout.tsx

Header

Header.css

Header.tsx

Footer

Footer.css

Footer.tsx

AnimalArea

Kitten

Kitten.css

Kitten.tsx

Puppy

Puppy.css

Puppy.tsx

ProductArea

CustomerArea

Models – מחלקות מידע טהור

KittenModel.ts

Services – מחלקות חישוביות

MonetaryService.ts

ApiService.ts

CLI: Command-Line Interface: קבוצת פקודות עבור ה-Command-Line, לרוב כולן מתחילות באותה המילה, ומותקנות ע"י ספרייה גלובלית.

Snippet: קטע קוד מוכן מראש.

ספרייה ליצירת קומפוננטות (התקנה גלובלית ברמת המחשב): npm i -g react-cli-snippets

Route: ניתוב של הכתובת הפנימית המוצגת בדפדפן לקומפוננטה שברצוננו להציג.

כתובת פנימית:

[http://www.mysite.com**/news/sports/football**](http://www.mysite.com/news/sports/football)

מנגנון Routing נבנה ע"י קומפוננטה שיודעת להציג Component כתלות ב-Route.

דרושה התקנה של ספרייה פנימית בפרויקט של ריאקט (בטרמינל הפנימי של VS CODE):

npm i react-router-dom @types/react-router-dom

אנו מגדירים איזה Route (נתיב פנימי ב-URL) גורם להצגה של איזו קומפוננטה.