

פיתוח Full Stack משלב שני עולמות:

צד שרת – שם מטפלים בלוגיקה, מסדי נתונים, אימות משתמשים.
צד לקוח – שם מציגים את המידע ועושים אינטראקציה עם המשתמש

HTTP – שכבות רשת

האינטרנט בנוי משכבות תקשורת – כל שכבה אחראית על חלק אחר בתהליך העברת המידע
שתי פרוטוקולים חשובים מאוד הם:

UDP – User Datagram Protocol

One to many broadcast

Small data packets

Non-indexed

TCP – Transmission Control Protocol

One to One / Peer to Peer

Complex and large data can be transmitted

Message can be broken to small parts

Each part has index

HTTP – IP Layer

- פרוטוקול HTTP יושב מעל TCP/IP – משתמש בשכבות האלו כדי להעביר מידע בצורה אמינה

HTTP – שימוש בטקסט

- משתמשים בפורמט **Unicode** – תקן בינלאומי לכתיבת טקסט.
- Unicode תומך בכל השפות בעולם – כולל עברית, ערבית, יפנית, אמוג'י ועוד.

מה זה Unicode?

- כל תו מיוצג ב-16 ביטים (2 בתים), מה שמאפשר אלפי תווים שונים.
- בזכות Unicode אפשר להעביר באינטרנט גם טקסטים בעברית וגם באנגלית וגם בסינית – בלי בעיות תאימות.

מה זה ביט (Bit)?

- ביט הוא יחידת המידע הכי קטנה במחשב.
- הוא יכול להיות רק אחד משני ערכים:
 - 0 (כבוי) אין זרם
 - 1 (דולק) יש זרם
- ביט אחד = 2 אפשרויות (0 או 1)
- 2 ביטים = 4 אפשרויות (00, 01, 10, 11)
- 3 ביטים = 8 אפשרויות
- 8 ביטים = **בייט** (Byte) = 256 אפשרויות (0–255)

דוגמה:

- האות **A** מיוצגת בבינארי כ-01000001 (8 ביטים).

איך הכל עובד?

הדפדפן שולח בקשת HTTP Request (HTTP) לשרת.

הדפדפן מתרגם את שם הדומיין (www.google.com) לכתובת IP.

השרת מקבל את הבקשה ומחזיר תגובה (HTTP Response).

אחרי שהדפדפן מקבל את המידע, הוא מפרש אותו ומציג אותו כאתר.

IP Address – כתובת IP

1. לכל אתר יש מזהה ייחודי (domain) ומאחוריו כתובת IP.

- כתובת IP היא מספר ייחודי שניתן לכל מכשיר שמחובר לאינטרנט
- היא משמשת כמו מספר טלפון – כדי שכל אחד יוכל למצוא אותך ולשלוח לך מידע.

השרתים משתמשים לרוב בכתובות סטטיות (לא משתנות).

המחשבים שלנו (לקוחות) מקבלים לרוב כתובת דינאמית (משתנה) בכל חיבור לאינטרנט.

IP – המחשב צריך אותו כדי להתחבר לשרת.

Domain – בני אדם צריכים אותו כדי לזכור את האתר.

מערכת בשם DNS (Domain Name System) מתרגמת בין השניים באופן אוטומטי.

URI – Uniform Resource Identifier

URI כולל את כל החלקים של הכתובת:

`http://www.google.com:80/index.html?q=url`

http = הפרוטוקול – scheme
www.google.com = שם הדומיין או הכתובת – domain
80 = פורט (יציאה) – port
index.html = הנתיב לקובץ או למשאב – path
q=url = פרמטרים נוספים שמועברים לשרת – query_string

HTTPS – Insecure Communication

פרוטוקול HTTPS – אותו HTTP, אבל עם שכבת הצפנה

ההצפנה מתבצעת באמצעות טכנולוגיה בשם SSL או הגרסה החדשה שלה TLS.

- SSL – Secure Sockets Layer (שכבת הצפנה ישנה)

- TLS – Transport Layer Security

כאשר HTTP "עובר דרך" SSL/TLS – הוא הופך ל-HTTPS
רוב ה-APIs היום מחייבים HTTPS כברירת מחדל

- הנתונים מוצפנים.
- השרת מזוהה על ידי תעודה (Certificate).
- אי אפשר לשנות את המידע בדרך

FTP – File Transfer Protocol

FTP = פרוטוקול תקשורת ייעודי להעלאה והורדה של קבצים בין מחשב מקומי לשרת

- העלאת קבצי אתר לשרת
- הורדת קבצים מהשרת לצורך גיבוי או עדכון
- מבוסס על TCP/IP – אמין
-

שרת FTP

- תוכנה שפועלת על השרת ושומרת את הקבצים

לקוח FTP

- תוכנה אצלך במחשב שמתחברת לשרת

XML – eXtensible Markup Language

- XML מטרתו לתאר נתונים ולא להציג נתונים כמו HTML
- שומר מידע גולמי

XML – מבנה

- Elements / Tags – תגיות שמקיפות את המידע (כמו `<name>David</name>`).
- Attributes – מידע נוסף על תגית (`<"person gender="M">`).
- CDATA – טקסט מיוחד שלא מתפרש כקוד.
- Comments – הערות בקובץ, לא חלק מהמידע.

XML – כללים

- כל תגית נפתחת ונסגרת
- ערכים במרכאות (`"id="123"`)
- אותיות גדולות וקטנות משמעותיות (Case Sensitive).
- תגיות מקוננות בצורה נכונה (אי אפשר לסגור תגית פנימית אחרי החיצונית).

XML – Validation

- **XSD – קובץ שמגדיר את החוקים של ה-XML**
- **איזה תגיות מותרות.**
- **איזה סוגי מידע מותר (מספר, תאריך וכו').**
- **אילו תגיות חובה ואילו אופציונליות.**
- **אם ה-XML לא תואם לסכימה – הוא נכשל בבדיקה.**

סיכום המצגת

מה זה אינטרנט?

- רשת עולמית של מחשבים ושרתים שמתקשרים דרך כבלים, אלחוט ועוד.
- ה-WWW (World Wide Web) הוא רק חלק מהאינטרנט – החלק שמציג דפי HTML.
- פרוטוקול התקשורת הנפוץ: **HTTP**.

שרתים (Servers)

- מחשבים ייעודיים שמאחסנים קבצי אתרים ומגיבים לבקשות מהלקוחות.
- דוגמאות לשרתים: Apache, IIS, Nginx.
- אחראיים לשלוח **תוכן סטטי** (HTML, תמונות) וגם **תוכן דינאמי** (PHP, Node.js).

לקוחות (Clients)

- מחשבים של המשתמשים – דפדפן הוא התוכנה שמבצעת את הבקשה ומציגה את התגובה.
- דפדפנים עיקריים: Chrome, Firefox, Safari, Opera.

HTTP – פרוטוקול בקשה/תגובה

- הלקוח שולח **HTTP Request**, השרת מחזיר **HTTP Response**.

- יושב מעל TCP/IP שמבטיח העברת נתונים אמינה.
 - כתובת אתר מורכבת מ-URI: scheme, domain, port, path, query string.
-

HTTPS – אבטחה

- HTTP עם שכבת SSL/TLS להצפנה.
 - משתמש במפתחות ציבוריים/פרטיים + מפתח סימטרי זמני.
 - מאובטח ע"י תעודות אבטחה מ-Certificate Authority (CA).
-

כתובת IP ודומיין

- לכל מכשיר יש IP ייחודי (IPv4 או IPv6).
 - דומיין הוא שם ידידותי לאדם שממופה ל-IP דרך DNS.
-

FTP – העברת קבצים

- פרוטוקול להעלאה/הורדה של קבצים בין מחשב מקומי לשרת.
 - מבוסס TCP/IP, תומך בהזדהות, לא מאובטח כברירת מחדל.
-

XML – מבנה נתונים

- שפה לתיאור מידע (לא להצגה).
 - מבנה עם תגיות, תכונות (Attributes), וכללים מחייבים.
 - ניתן לבדוק ולוודא נכונות בעזרת (XSD Schema).
-

מה לקחת מפה כמתכנת Full Stack?

- להבין איך דפדפן ושרת מדברים זה בסיס לכל מה שנעשה.
- לדעת להבדיל בין תוכן סטטי ודינאמי.
- להכיר פרוטוקולים מרכזיים: HTTP, HTTPS, FTP.
- להבין כתובות אינטרנט: IP, דומיין, URI.
- לקרוא XML כי עדיין תיתקל בזה במערכות ותיקות ו-APIs.