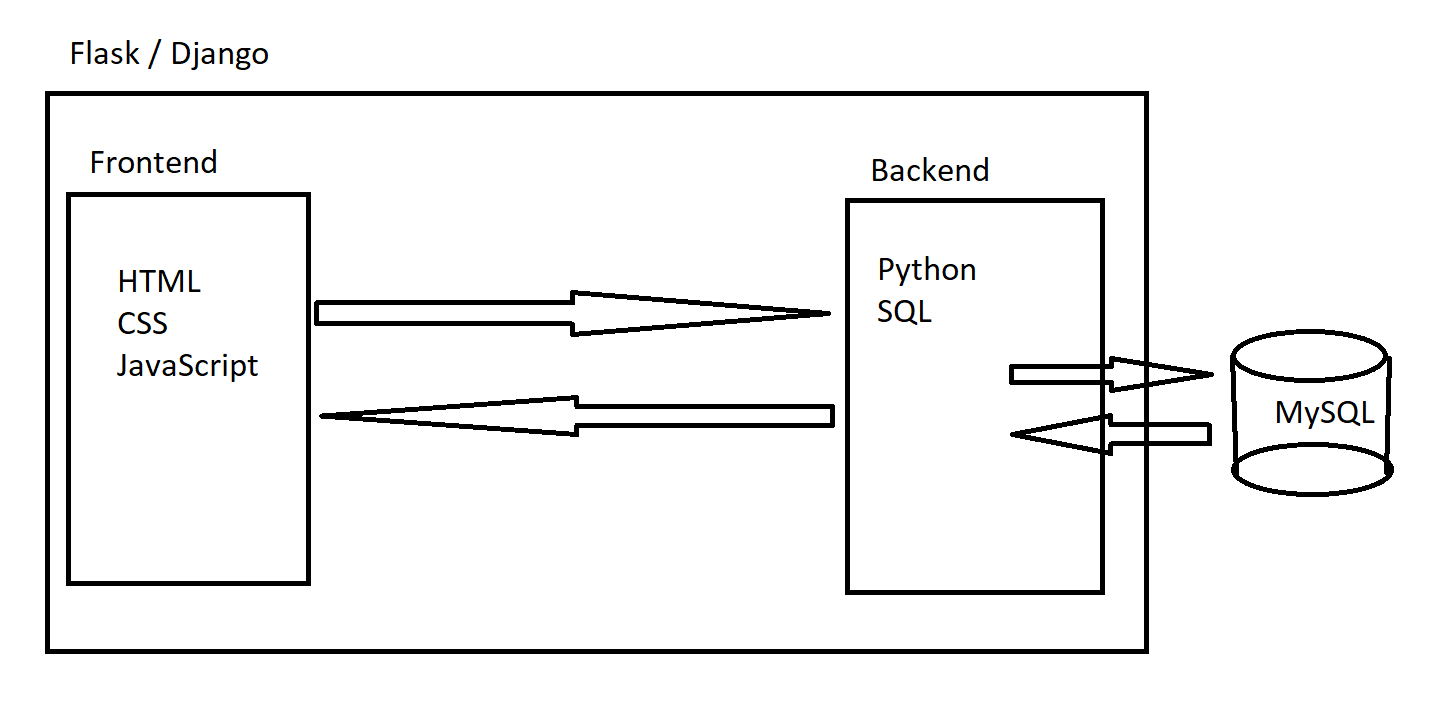
# **Full Stack**

**Framework**

מערכת מוכנה שאנו עובדים בתוכה ובמסגרת החוקים שלה והיא מאפשרת לבנות עבורנו חלק מהאתר או את כולו.

בפייתון יש שני Frameworks נפוצים:

1. Flask: מאפשרת לבנות בפייתון מערכת Full Stack, הינה מאוד Light, מיועדת לאתרים לא גדולים מידי, לא יותר מידי קובעת בשביל המתכנת איך לעבוד – יותר גמישה.
2. Django: מאפשרת לבנות בפייתון מערכת Full Stack, מיועדת לאתרים יותר גדולים, קובעת בשביל המתכנת איך לעבוד – פחות גמישה.



**Route**

זהו הניתוב הפנימי באתר שמוצג בדפדפן. לדוגמה: בכתובת הבאה: [http://mysite.com**/abc/def**](http://mysite.com/abc/def) ה-/abc/def זהו ה-Route

**View**

פונקציה בפייתון הנמצאת בצד השרת אליה הלקוח גולש ע"י תיאור Route בכתובת הדפדפן.

Blueprint זהו אובייקט המסוגל לייצר Views

**Model**

מחלקת מידע טהור

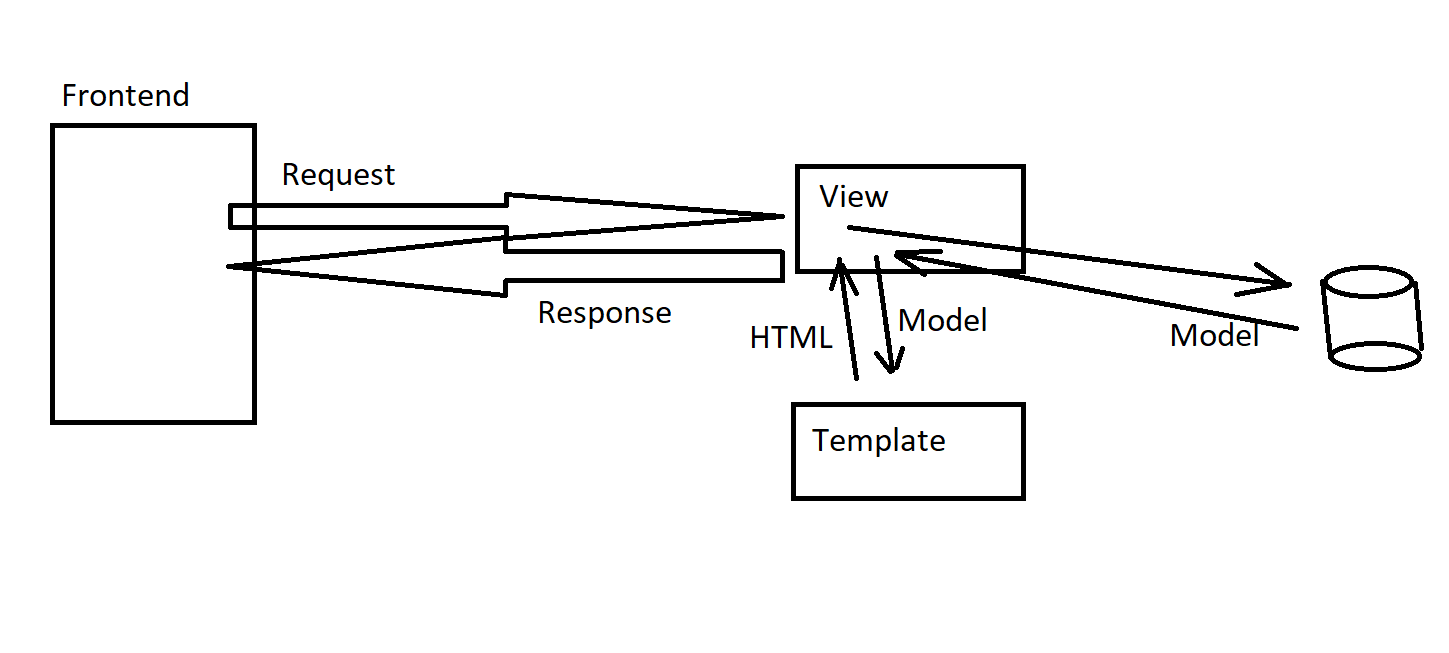
**Template**

זה דף HTML שפונקציית ה-View מחזירה ללקוח.

כלל השיטה הזו נקראת MVT (Model-View-Template)

**ה-Flow של Flask:**

1. משתמש גולש ע"י הדפדפן ל-Route ספציפי.
2. ה-Route מנתב לפונקציה ב-View.
3. פונקציית ה-View ניגשת ל-DB ומביאה מידע (Model).
4. הפונקציה שולחת את המידע ל-Template (קובץ HTML) לשילוב המידע
5. ה-Template מחזיר HTML ל-View
6. ה-View מחזיר את ה-HTML בחזרה ללקוח.



יצירת פרויקט Flask

1. יצירת Environment:

py -m venv env

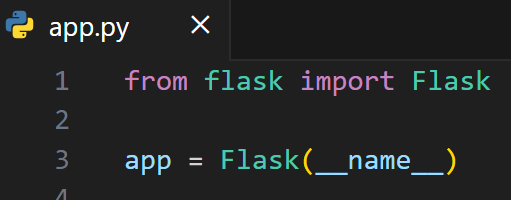
1. הרצת ה-Environment:

env/Scripts/activate

1. התקנת Flask פנימית בפרויקט:

pip install flask

1. יצירת תיקיית src ו-app.py בתוכה.
2. הגדרת אובייקט app:



1. הרצת האפליקציה ע"י פקודת הטרמינל:

flask --app src/app.py run --debug