**תיעוד לקוד**

**Models.py:**

class Specialization(Enum):

נותן לנו את סוגי בעלי המקצוע

class Customer(AbstractBaseUser):

מחלקה של הלקוח יוכל לשמש גם כמשתמש

class Professional(AbstractBaseUser):

מחלקה של בעל מקצוע יוכל לשמש גם כמשתמש

def update\_average\_rating:

מעדכנת את הדירוג של בעל מקצוע אחרי שפרויקט שהשתתף בו הסתיים

def money\_for\_pro:

כאן נחשב על פי מיון התמחות של בעלי מקצוע שנקבל כפרמטר כמה אחוז מהתקציב ילך להם לשכר

class Question(models.Model):

כאן נשמור את השאלות(יהיה לה גם שדה של התחום של בעל המקצוע שנדע לאיזה להפנות את השאלה)

class Answer(models.Model):

כאן נשמור את התשובות כל תשובה ציון בין 1 ל 5

class JobDetail(models.Model:

כאן נשמור את פרטי העבודה שהלקוח יצר

def check\_project\_status:

בודקת אם פרויקט הסתיים

def determine\_professional\_priority:

כאן נחזיר רשימה של סוגי ההתמחות לפי סדר עדיפות שלהם לפי האפיון שקיבלנו מהלקוח לגבי הפרויקט שלו

class AnswerJob(models.Model):

כאן נשמור את התשובות לשאלות ששאלנו את הלקוח ביצירת הפרויקט

class JobApproval(models.Model):

זו מחלקה שמחברת בין הבעל מקצוע לפרטי העבודה שהלקוח יצר כאן יהיה לנו 4 שדות: תז של העבודה החדשה שנוצרה, תז של בעל המקצוע שנבחר ,ערך בוליאני שיגיד אם בעל המקצוע אישר את העבודה והמשכורת שמוצעת לו

**views.py:**

def Home\_new(request):

מחבר לעמוד הבית של html

def home(request):

דרך דף הראשון שיוצג נבחר בו את המשתמש ומשם נעבור להרשמה או להתחברות

def customer\_register(request):

דרך עמוד הרשמה ללקוח כאן הלקוח יוכל להירשם נוודא תקינות ושמור במאגר הנתונים

def customer\_login(request):

התחברות הלקוח נבדוק אם השם משתמש והסיסמא תואמים ומשם נשלח אותו לדף הבית של הלקוח

def professional\_register(request):

דרך לעמוד הרשמה לבעל המקצוע כאן הבעל מקצוע יוכל להירשם נוודא תקינות ושמור במאגר הנתונים

def professional\_login(request):

התחברות הבעל מקצוע נבדוק אם השם משתמש והסיסמא תואמים ומשם נשלח אותו לדף הבית של הבעל מקצוע

def user\_type(request):

בדף הראשון נוודא איזה סוג משתמש בחרנו בעזרת הפונקציה הזו

def customer\_home(request):

עמוד הבית של הלקוח נוודא שהלקוח שהתחבר הוא הלקוח שנמצא בעמוד הבית יהיו לו 2 אופציות:1.לעקוב אחרי עבודות קיימות שיצר 2. ליצור עבודה חדשה

def get\_best\_pro(professional, ans\_job\_special):

זו פונקציה שמקבלת בעל מקצוע מסוים ואת התשובות שהלקוח מילא לעבודה מסוימת הפונקציה תשווה בין התשובות של הבעל מקצוע בהתאם לתשובות של הלקוח ביצירת העבודה המסוימת לפי התז של השאלה ותחזיר ציון לפי מספר התשובות התואמות

def get\_matching\_professionals(new\_job\_detail):

זו פונקציה שמקבלת את פרטי העבודה החדשים שהלקוח רוצה ליצור ומחזירה מערך שמורכב מהבעלי המקצוע שמצאנו מתאימים לעבודה שהלקוח רוצה. בשלב הראשוני אנחנו נביא את כל הבעלי מקצוע שפנויים ולאחר מכן נביא את כל התשובות שהלקוח מילא לאחר מכן נרוץ בלולאה על כל הסוגי התמחות ואז לכל סוג התמחות נעבור על הבעלי מקצוע שנמצאים בתחום הזה ונמצא את הבעל המקצוע המתאים לאותו תחום ונוסיף למערך

def job\_create(request):

כאן ניצור עבודה חדשה בהתחלה יופיעו פרטי העבודה שהלקוח ימלא ולאחר מכן סט שאלות ללקוח שאחר כך ננתח אותם כדי למצוא את הבעלי מקצוע המתאימים ואז נעבור לדף ששם נבחר אם אנחנו רוצים את הבעלי מקצוע שקיבלנו

def job\_detail(request):

זו פונקציה שתחזיר את העבודות שהלקוח יצר. לכל עבודה יופיעו הבעלי מקצוע שאישרו את העבודה ואלה שלא אישרו

def get\_detail\_professional(request):

כאן יופיעו שאלות לבעל מקצוע לפי התחום התמחות שלו כדי שנוכל לנתח את אותו העל מקצוע

def professional\_home(request):

דף הבית של בעל המקצוע כאן יופיעו לו שני אופציות:1: לענות שאלות שיאפיינו אותו 2: לראות אם קיבל הצעות עבודה ולבחור אם לאשר אותם

def choose\_pro(request):

פונקציה זו תראה את המערך של הבעלי מקצוע שקיבלנו לפי היצרת עבודה של הלקוח וניתן לו אפשרות לבחור האם לשלוח להם הצעה או לא לשלוח להם

def job\_offers(request):

בפונקציה זו נראה לבעל המקצוע האם יש לו הצעות עבודה ואם יבחר לאשר אותם נשנה את השדה שלו ללא פנוי

def get\_total\_score(new\_job\_detail):

כאן נסכום את סך התשובות של הלקוח לשאלות האפיון של הפרויקט זאת בשביל להבין לאיזה בעל מקצוע לתת עדיפות בשכר

def rate\_professionals(request):

כאן נשלח לעמוד של דירוג בעלי מקצוע ונשמור את הדירוג שהלקוח נתן להם