# flask-UPDATE ACCOUNT

בפרק הקודם ראינו כיצד לבצע פונקציית hashing על סיסמאות, איך לשמור אותן במסד ולאמת שאכן הוכנסה הסיסמא והמייל הנכונים. ראינו איך יוצרים validationError מתואם אישית, ואיך להחתחבר ולהתנתק בפועל למערכת או מהמערכת. בסוף הפרק התחלנו לבנות עמוד account שמאפשר למשתמש לשנות פרטים אישיים.

בפרק זה נמשיך לממש את העמוד- נראה איך אפשר לשנות נתונים מהמסד.

## טופס UpdateAccountForm-

אם אנחנו רוצים לשנות את הפרטים של המשתמש, צריך לבנות איזשהו טופס שעליו נבצע את השינויים.   
נכנס ל-forms.py ונוסיף טופס חדש שישמש לעדכון החשבון. נשים לב שהטופס יהיה יחסית זהה לטופס RegiserationForm למעט אולי הסיסמא ואימות הסיסמא, אבל זה תלוי אם נרצה לשנות את זה בטופס או במייל, כרגע נשאיר בלי האפשרות לשנות סיסמא.

עכשיו אם המשתמש לא ירצה לשנות את שם המשתמש שלו אלא רק את המייל, או להפך, הרי שבכל מקרה צריך לבצע בדיקה שהשם או המייל שהוא הכניס לא שמור כבר במסד, ואם הוא לא שינה את שם המשתמש שלו המערכת תזהה את זה כניסיון להשתמש בשם משתמש תפוס כי היא תקבל את השם המקורי שלו ממתודת ה-post של הטופס.

בשביל זה נצטרך לוודא שהמשתמש שמחובר כרגע הכניס פרטים ששונים מהפרטים הנוכחים שלו, אחרת לא נבצע בדיקה (כי הוא הכניס את השם של עצמו). כדי לדעת פרטים על המשתמש שמחובר כעת נשתמש ב-current\_user של flask\_login, אז לא לשכוח לייבא אותו:

from flask\_login import current\_user

...

class UpdateAccountForm(FlaskForm):

    username = StringField('Username',

                           validators=[DataRequired(), Length(min=2, max=20)])

    email = StringField('Email',

                        validators=[DataRequired(), Email()])

    submit = SubmitField('Update')

    def validate\_username(self, username):

        if username.data != current\_user.username:

            user = User.query.filter\_by(username=username.data).first()

            if user:

                raise ValidationError('That username is taken. Please choose a different one.')

    def validate\_email(self, email):

        if email.data != current\_user.email:

            user = User.query.filter\_by(email=email.data).first()

            if user:

                raise ValidationError('That email is taken. Please choose a different one.')

עכשיו נרצה לעשות כמה שינויים קוסמטיים.   
דבר ראשון נרצה להוסיף תיקייה חדשה, בתוך static שתכיל תמונת פרופיל, נקרא לה profile\_pics.  
לקחנו איזושהי תמונה שמצאנו באינטרנט שתייצג תמונת פרופיל דיפולטיבית.

**עדכון השם והמייל של המשתמש-**

שינוי קוסמטי נוסף יהיה להוסיף את התמונת פרופיל לעמוד account, כאן נשתמש בקטע קוד html מוכן מראש כדי שיתאים את התמונה בצורה יפה לראש העמוד, נלך ל-account.html ונשנה אותו לזה:

{% extends "layout.html" %}

{% block content %}

    <div class="content-section">

      <div class="media">

        <img class="rounded-circle account-img" src="{{ image\_file }}">

        <div class="media-body">

          <h2 class="account-heading">{{ current\_user.username }}</h2>

          <p class="text-secondary">{{ current\_user.email }}</p>

        </div>

      </div>

<!--CONTENT-->

</div>

בנוסף נרצה להציג את מבנה הטופס, לכן נעתיק את המבנה (ה-html) של דף הרישום בחלקים של הטופס, נדביק במקום <!--CONTENT--> ונמחק את השדות שלא רלוונטים אלינו (שהם הסיסמא ואימות הסיסמא).  
כפי שחלקכם שמו לב אנחנו משתמשים במשתנה שנקרא image\_file, אי לכך נצטרך לשלוח אותו בפונקציית הניתוב.   
אז נעבור ל-routes.py, נייבא את המחלקה UpdateAcountForm ונלך לפונקציה account() .  
דבר ראשון ניצור משתנה עבור תמונת הפרופיל. כרגע התמונה היחידה שיש לנו היא התמונה הדיפולטיבית, ועדיין אין אפשרות לשנות אותה. נשתמש בפונקציה url\_for כדי לקבל את כתובת התמונה. לפונקציה נכניס את 'static' כפרמטר ראשון והפרמטר השני יהיה התיקייה של התמונה שרשור עם שם התמונה של המשתמש ,בהמשך נשמור את כל התמונות של המשתמשים בתיקייה הזאת. ונעביר את התמונה כפרמטר לפונקציית הרינדור render\_template().   
וכמובן ניצור אינסטנס חדש לאובייקט UpdateAccountForm ונעביר אותו לפונקציית הרינדור גם כפרמטר:

...  
from flask\_example.forms import ... , UpdateAccountForm  
...  
@app.route("/account")

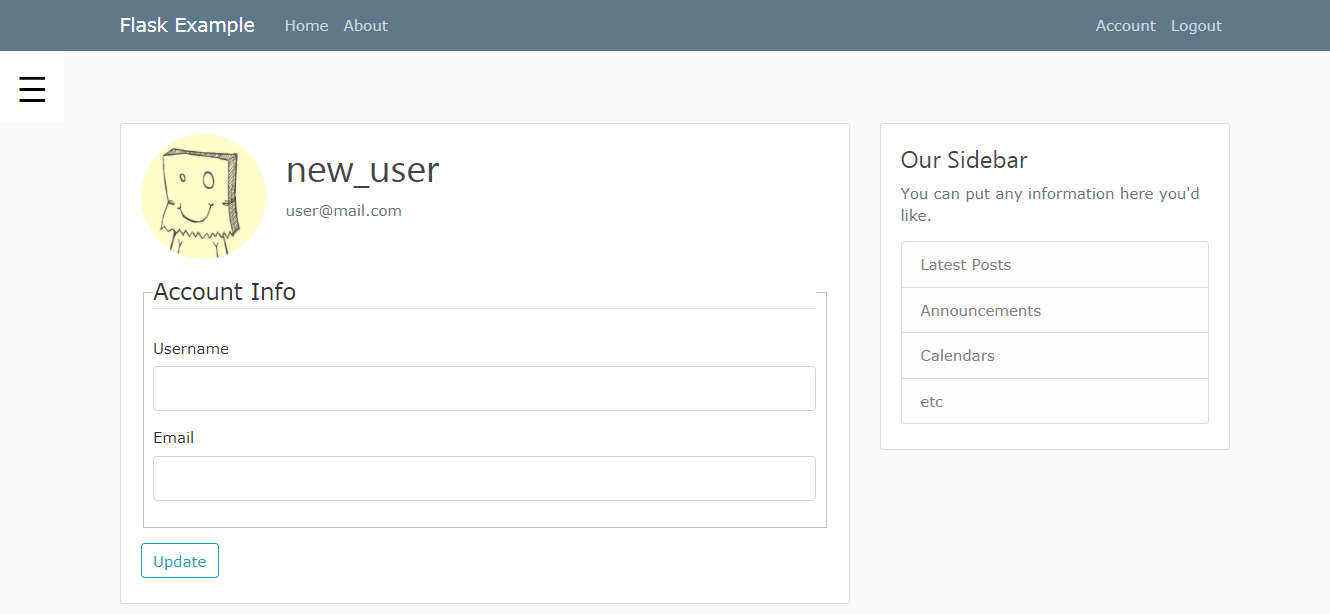
@login\_required

def account():

    form = UpdateAccountForm()

    image\_file = url\_for('static' , filename= f'profile\_pics/{current\_user.profile\_img}')

    return render\_template('account.html', title='Account', image\_file = image\_file , form = form)

נשמור, נריץ ואם אין משתמש מחובר ניכנס לאיזשהו משתמש שהגדרנו. אנחנו אמורים לקבל משהו כזה בערך: 

שני דברים שצריך להוסיף: 1. שהשינויים ישמרו במסד הנתונים. 2.שנוכל להטעין תמונה.

אבל לפני זה, משהו נחמד שכדאי להוסיף זה, שכאשר המשתמש נכנס למערכת, כדאי שבתיבת הטקסט ב-account יופיעו השם והמייל הנוכחים שלו. אז דבר ראשון שחשוב שנוסיף לפונקציה זה מתודות GET ו-SET. אח"כ נבצע בדיקה האם השליחה של ההודעה הייתה תקנית עם הפונקציה validate\_on\_submit(), אם כן נשמור את השינויים במסד נתונים.   
אחד היתרונות הגדולים של sqlalchemy זה הפשטות של ביצוע פעולות שכיחות על מסדי נתונים.  
אם נרצה לשנות את השם של המשתמש או המייל שלו, פשוט נשנה את השדה המתאים ב-current\_user ונבצע commit() בסוף. למשל כדי לשנות את השם של המשתמש: current\_user.username = form.username.data  
אחרי שנשנה את שם המשתמש והסיסמא שלו נשלח גם הודעת flash שמודיעה שהשינויים בוצעו כהלכה.

אמרנו גם שיהיה נחמד לו התיבות טקסט היו מגיעות עם שם המשתמש והמייל הנוכחים כשנכנסים לעמוד account, כדי לעשות את זה נשים לב לדבר הבא: כשאנחנו מבקשים מהשרת לעבור לדף מסויים אנחנו מבצעים מתודת get, ומתי ששולחים הודעה משתמשים במתודת post. נרצה לבדוק האם ההודעה שקראה לפונקציה היא מסוג get ואם כן לאכלס את התיבות טקסט של הדף עם הנתונים הנוכחיים של המשתמש. כדי לעשות את זה נצטרך רק לשים ערכים למשתנים form.username.data ו- form.email.data ולתת להם את הערכים של current\_user.username ו-current\_user.email בהתאמה והם יופיעו אוטומטית בעמוד:

...  
@app.route("/account" ,methods=['GET', 'POST'])

@login\_required

def account():

    form = UpdateAccountForm()

    if form.validate\_on\_submit():

        current\_user.username = form.username.data

        current\_user.email = form.email.data

        db.session.commit()

        flash('Your account has been updated!', 'success')

        return redirect(url\_for('account'))

    elif request.method == 'GET':

        form.username.data = current\_user.username

        form.email.data = current\_user.email

    image\_file = url\_for('static' , filename= f'profile\_pics/{current\_user.profile\_img}')

    return render\_template('account.html', title='Account', image\_file = image\_file , form = form)

נשמור, נריץ, ואנחנו אמורים לראות את שמות השדות של המשתמש מופעים כבר התיבות הטקסט של דף account, ואם ננסה לשנות את שם המשתמש או המייל הפעולה אמורה לעבוד כמו שצריך, ואפילו אנחנו מקבלים הודעה אינפורמטיבית שהחשבון עודכן.