



פרויקט גמר הנדסאי תוכנה

שם הפרויקט: "Coupon Club"

שמות הסטודנטים:

данיה דבש

עайд חלייל

שם מנהה:

שגב טוביה

הצהרת סטודנט**שם הסטודנט: דאניה דבש ת.ז. 214257412****שם הסטודנט: עайд חיליל ת.ז. 325870079**

שם המכללה בה לומד הסטודנט: מכללת אורט ירושלים.

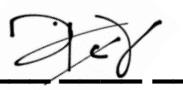
אני הח"מ, מצהיר בזאת כי פרויקט הגמר וספר הפרויקט המcz"ב נעשו על ידי ידינו בלבד.

פרויקט הגמר נעשה על סמך הנושאים שלמדתי במכללה ובאופן עצמאי.

פרויקט הגמר וספר הפרויקט נעשה על בסיס הנחייתו של המנחה האישי.

מקורות המידע בהם השתמשתי לביצוע פרויקט הגמר מצוינים ברשימת המקורות המצוינים בספר הפרויקט.

אני מודע לאחריות שהנני מקבל על עצמי על ידי חתימתה על הצהרה זו שכל הנאמר בה אמת ורק אמת.

חתימת הסטודנט:  תאריך: 21-7-2024

חתימת הסטודנט:  תאריך: 21-7-2024

חו"ר מנהל מה"ט – 11-4-51 נספח מס) 1 הצעה לפרויקט גמר

תאריך 21-07-2024

לכבוד
יחידת הפרויקטים
מה"ט

הצעה לפרויקט גמר

1. פרטי הסטודנטים

שם הסטודנט	ת.ז.	כתובת	טלפון נייד	ת.ס.ום לימודים
עайдן חלייל	325870079	הר הזיתים	0543177169	
דانيا דבז	214257412	בית צפפה	0542376671	

.2

שם המכללה: מכללת אורט ירושלים סמל המכללה:

מסלול הכשרה: הנדסאים.

מגמת לימוד: תוכנה

מקום ביצוע הפרויקט: מכללת אורט ירושלים

2. פרטי המנהה האישית

שם המנהה	כתובת	טלפון נייד	תואר	מקום עבודה
שאיב טוביה	הבדולח 6, מעלה אדומים	054-6782784	B.Sc. M.B.A.	עצמאי

.3

חתימת המנהה האישית

חתימת סטודנט

אישור המנהה האישי

הריני מאשר שהפרויקט בוצע בהנחייתך, קראתך את ספר הפרויקט ומצאת
כי הוא מוכן לצורך הגשת הסטודנט להגנה על פרויקט גמר.

שם המנהה: שגיב טוביה חתימה: תאריך:

אישור ראש המגמה

הריני מאשר שספר הפרויקט מוכן לצורך הגשת הסטודנט להגנה על
פרויקט הגמר.

שם ראש המגמה: שגיב טוביה חתימה: תאריך:

תודות פרויקט זה היוו עבורי ההזדמנות נדירה לבחון ולישם את הידע הנרחב והמגוון, אותו רכשנו במהלך כל שנות לימודנו במחלקה התוכנה שבמכללת אורט ירושלים.
חשוב לנו, להביא את תודתנו הרבה והערכתנו לכל אלו אשר תמכו, סייעו ועזרו לנו לכל מהלך הדרכ:

ברצוננו להודות:

למנהל האקדמי שלנו, למר טוביה שגיב - על הנחיתת הפרויקט, והעזרה המקצועית לאורך הדרכ. תודה על הסבלנות, ותודה על השקעת הזמן, תוך דגש על הקידום המקצועי והעשרהינו על הכל, תודה.
דانيا דבוס ועaid חלי.

תוכן עניינים

1. שם הפרויקט

2. תמצית הפרויקט

3. מצב קיים בשוק

4. בעיות ופתרונות

5. משאבים נדרשים להציג הפרויקט

6. דרישות מערכת

7. תוכנית עבודה GANT

8. ניטוביים

9. למה בחרת בטכנולוגיות אלו

10. עץ התיקיות

11. מה הפרויקט הקנה לנו

12. מסכימים

13. מודלים

14. קוד

1. שם הפרויקט :-

אפליקציה אפליקציית קופונים - **Coupon Club**

2. תמצית הפרויקט:-

- **רקע:**

אפליקציה אפליקציית קופונים , האפליקציה נותנת שירות הרשמה, צפייה בחברות שהיא תומכת בהם, צפייה בקופהים של כל חברה, הוספה קופון חדש. אפשרות לשמור את הקופונים לצרכים בדף פרופיל של המשתמש. כמו כן אפשר לעדכן אימייל ושם משתמש. האתר מאד נגיש ונוח, עם מגוון טכנולוגיות קצה.

- **שפות וכליים לבניית הפרויקט:**

האפליקציה נבנתה בסביבת עבודה של **visual studio code**,
טכנולוגיות בשימוש:

בצד שרת : Node.js, Type Script , Express

בצד לקוח : React Native, Type Script

ושומר את הנתונים בעזרת **MongoDB** ונבנה באמצעות מערכת הפעלה **Windows**.

3. מצב קיימם בשוק:

רצינו לקנות מתנה לאחד החברים ואז חיפשנו קופוניים להנחהות והיה קשה מאוד להשיג קופוניים.
אפשרו לא ניתן לפרסם קופוניים חדשים, נניח שאחד המستخدمים רוצה לשיתף קופון, הוא לא יכול, لكن DIGSHNU שגם המשתמש יכול לצרף לאפליקציה את הקופוניים שהוא רוצה.

החליטנו לבנות אפליקציה שאוספת מגוון קופוניים מחברות רבות ומציגה אותם ללא עלות כדי לעזור לאנשים להשיג קופוניים בקלות.

בדקנו אם כבר קיימות אפליקציות שהם דומות, ראיינו איך הן עובדות, אבל רואים שהם מציגים על השיווק ולא על הקופוניים.

אפליקציות דומות:

Coupert
Rekuten
AlMufer
Groupon

4. בעיות ופתרונות:

היו לנו הרבה תקלות שישבנו עליהם במהלך זמן רב, תקלות קשור בין צד שרת לצד לקוח, הוצאת נתונים מבסיס נתונים, הנראות היכי נוכה לעין המשתמש מבחינת צבע ועיצוב.

ישבנו עם המנהה שלנו, קיבלנו ממנו את הכלים הנכונים זה עזר לנו מאוד, אז התשםנו ב Stack Overflow | GetHup ועוד כמה אתרים שהיה לאנשים אותם תקלות, והצלחנו לפתרם אותם והעלו את הפתרונות שלהם לאתר. ובעזרת כמה חברים, הצלחנו לתקן ולעboro את כל התקנות שהו.

5. משאבים נדרשים להצגת הפרויקט:

-: Frontend •

React Native: לפיתוח האפליקציה בטלפון.

TypeScript: להוספת סוגיות סטטיות לשפות JavaScript.

React Navigation: לניהול ניווט בתחום האפליקציה.

Axios: לביצוע בקשות HTTP לשרת.

React Native StyleSheet: לעיצוב האפליקציה.

-:Backend •

Node.js: להרצת צד השרת.

Express: לפיתוח ה-Api.

Mongoose: לתקשורת עם MongoDB.

MongoDB: למסד נתונים NoSQL.

מסד נתונים:

MongoDB מאפשר להשתמש בשירותים כמו MongoDB Atlas לניהול פשוט יותר של מסד הנתונים.

6. דרישות מערכת:

- **מכשירים ניידים:** טלפונים חכמים וטאבלטים עם מערכת הפעלה 8 Android ומעלה או 12 OS ומעלה, לצורך בדיקות והפצה של האפליקציה.
- **דרישות לקוחות:**
 - חיבור אינטרנט מהיר ויציב לצורך גישה למערכת ולהורדת האפליקציה.
 - גישה למייל.

7. תכנית עבודה GANT:

חשבנו על כמה אפליקציות, אבל ברגע שנתקלנו עם הבעיה של הקופונים החלטנו לעשות אפליקציה קופוניים כמו שאמרנו לעיל
ותאריך הבחירה היה ב 1.02.2024.

ולאחר מכן בחרנו את התכנולוגיה הci מתאימות לשוק והci ייעילות
ונוחות לעבודה זהה היה בתאריך 16.02.2024

כמו שאמרנו היו לנו תקלות עד שהבנו את התכנולוגיות איך צריך
לעבוד ומה צריך כדי ליצור את האפליקציה ועבדנו וחיפשנו עד שהגענו
למסקנה והבנו איך עובדת אפליקציה זהה היה ב 7.03.2024

התחנו לחשב על צד השרת, ציירנו איך הוא אמור לעבוד עם הבסיס
נתונים, אז התחילנו להכין בקשות CRUD-API ולבדק מול PostMan
שהיה מעביר את הנתונים לבסיס נתונים, התחילנו לעשות עיצוב של
צד לקוח, וחברנו אותו ביחד וזה היה ב 28.03.2024

היי כמה חברים שבדקו והשתמשו באפליקציה והיה להם נעים
 מבחינות עיצוב וצבע, ואהבו את האפליקציה, היה להם כמה הערות
 שעוזרו לנו לשפר אותה וזה היה ב 27.06.2024
 סיימנו לעבוד על הפליקציה בתאריך 20.07.2024

8. ניטוביים :-

:User's •

Post	/auth/ create	קיבל נתונים מהלוקות וייצור משתמש חדש בסיס נתונים
Post	/auth /verify-email	על מנת לאמת את המail שהזנת
Post	/auth/re-verify-email	הוא מוחדש את מס' האimoto
Post	/auth/forget-password	הוא מייצר קישור לאיפוס סיסמה ושולח אותו למיל של המשתמש
Post	/auth /verify-pass-reset-token	הוא בודק את תוקף אסימון איפוס הסיסמה ומאשר אם הוא תקין
Post	/auth /update-password	הוא מאמת את תקינות אסימון איפוס הסיסמה ומעדכן את הסיסמה החדש
Post	/auth /sign-in	הוא מאמת את פרטי ההתחברות ומחזיר אסימון גישה ומידע על הprofil
Get	/auth /is-auth	הוא מאמת את האסימון כדי לבדוק אם המשתמש מחובר ושולח את פרויף המשתמש אם האimoto הצליל
Post	/auth /update-profile	הוא מאמת את האסימון, מעבד קבצים מ--form, data, ומעדכן את פרויף המשתמש
Post	/auth /log-out	הוא מאמת את האסימון, ו מבצע יציאה מהמערכת, עם אפשרות לצאת מכל המכניםים

:Coupon's •

Post	/coupons/create	הוא יוצר קופונים חדשים ושולח תשובה על הצלחה או שגיאה
Get	/coupons/GetCoupon	הוא שולף את הקופונים מהמסד נתונים ושולח את התוצאה
Get	/coupons/:CouponId	הוא מוחש קופון לפי מזהה ומחזיר את התוצאה או שגיאה אם לא נמצא
Patch	/coupons/:CouponId	הוא מעדכן קופון לפי מזהה ומבודא שאין כפילות בקוד הקופון
Delete	/coupons/:CouponId	הוא מוחק קופון לפי מזהה ומחזיר הודעה על הצלחה או שגיאה

9. למה בחרת בטכנולוגיות אלו :-

• : React Native

כי היא משלבת בין שני מערכות הפעלה גם **IOS** וגם **ANDROID** חלק גדול מהקוד ניתן לשתף בין פלטפורמות שונות, מה שמאפשר את תחזוקת האפליקציה ומודול עליונות פיתוח.

שיכולות לעזור לך להתמודד עם אटגרים שונים **Libraries** יש קהילה פעילה ותוספות רבות וליעיל את הפיתוח.

מציע ביצועים טובים ואחדים יחסית על פני פלטפורמות שונות, מה שחשוב לחוויית משתמש טובה.

• : Node.js & Express

כי היא נחשבת לפתרון מעולה ומהיר לתקשרות עם צד שרת ובסיס הנתונים ובנניה לצורך בקוד פתוח.

mbioso על אירוחים ואסינכרוניות, מה שמאפשר לו לטפל בכמות גבוהה של בקשות **Node.js** במקביל, מה שחשוב עבור אפליקציות.

הוא פרימורך קל ונitin להתקאה אישית, שקל על פיתוח והקמה מהירה של **Express Api**.

גם בצד הלקוח וגם בצד השרת, מה-**TypeScript/JavaScript** מאפשר לך להשתמש בשמfuck את ניהול הקוד ואת התקשרות בין החלקים השונים של המערכת.

• : MongoDB

עזרה לגילות שגיאות בזמן הקומpileציה, מה שמחזית את מספר השגיאות בתוכנית ושיפור איכות הקוד.

ספקת ביצועים טובים במיוחד כשמדבר בניהול נתונים לא מובנים כמו תמונות או תכנים נוספים.

מתאימה מאוד לאפליקציות עם דרישות עלות של נתונים, מה שמשמע
לך להתמודד עם גידול בכמות המשתמשים והנתונים.

- גישה קלה ומהירה למידע.
- נטמרק בשפות רבות.
- יכולת אינדוקס גמישה.
- זמינות גבוהה וскопול הנתונים השומר על שלמות המידע.
- אחסון ביןארי הגורם להגדלת המהירות.

:TypeScript •

עזרת לגנות שגיאות בזמן הקומpileציה, מה שמחית את מספר השגיאות בתוכנית ושיפור איכות הקוד.

נותה להיות יותר קרייא ומסודר, מה שמקנה יתרון בפרויקטים גדולים ומשמע **TypeScript** קוד בביטחון ושיתוף פעולה בצוות.

תמיכה טובה עם **React Native & Node.js**: משלב היברידי עם **Native & TypeScript ,Node.js** מה שמאפשר לך ליהנות מה יתרונות של טיפוסיות בזמן הפיתוח.

.10. עכנתונות:

- בצד שרת:

```

    <div style="display: flex; justify-content: space-between; align-items: center;">
      <div style="flex-grow: 1;">
        <div style="background-color: #f0f0f0; padding: 10px; border-radius: 5px;">
          <ul style="list-style-type: none; padding-left: 0;">
            <li>> Backend
            <li style="margin-top: 5px;">> node_modules
            <li style="margin-top: 5px;"><div style="border-bottom: 1px solid #ccc; padding-bottom: 5px; margin-bottom: 5px;"><div style="display: flex; justify-content: space-between; align-items: center;">
              <div style="flex-grow: 1;">
                <div style="background-color: #f0f0f0; padding: 10px; border-radius: 5px;">
                  <ul style="list-style-type: none; padding-left: 0;">
                    <li>> src
                    <li style="margin-top: 5px;"><div style="display: flex; justify-content: space-between; align-items: center;">
                      <div style="flex-grow: 1;">
                        <div style="background-color: #f0f0f0; padding: 10px; border-radius: 5px;">
                          <ul style="list-style-type: none; padding-left: 0;">
                            <li>> @types
                            <li style="margin-top: 5px;">TS user.ts
                          
                        </div>
                      <div style="font-size: 1.5em; color: #ccc;">>
```

```

Frontend
  src
    @types
      navigation.ts
    api
      client.ts
    assets
    components
      form
        AddCoupon.tsx
        AppCotnainer.tsx
        AuthFormContainer.tsx
        Category.tsx
        CouponCard.tsx
        EditBar.tsx
        Header.tsx
        LogoutButton.tsx
    navigation
      AuthNavigator.tsx
      index.tsx
      TapNavigator.tsx
    store
      auth.ts
      coupon.ts
      index.ts
    ui
      AppButton.tsx
      AppInput.tsx
      AppLink.tsx
      CircleUi.tsx

```

• בצד לקוח :

```

  Loader.tsx
  OTPField.tsx
  PasswordVisiblityIcon.tsx
  utils
    asyncStorage.ts
    AuthContext.tsx
    colors.ts
    CopunContext.tsx
    imageMap.ts
  views\auth
    CouponPage.tsx
    EditProfile.tsx
    HomePage.tsx
    LostPassword.tsx
    ProfilePage.tsx
    SignIn.tsx
    SignUp.tsx
    Verification.tsx

```

11. מה הפרויקט הקנה לנו :-

ניסיון טכנולוגי מעשי :

בעבודה עם טכנולוגיות כמו React Native, Node.js, TypeScript ו MongoDB חיזק את הידע שלנו בתחוםים אלו, מה שיכל לשפר את יכולתנו לפתח פתרונות טכנולוגיים מתקדמים. ההתמודדות עם בעיות ותקלות בזמן הפיתוח פיתחה את כישוריتنا בפתרון בעיות והתאמת טכנולוגיות לצרכים ספציפיים.

כישורים בפיתוח פרויקטים :

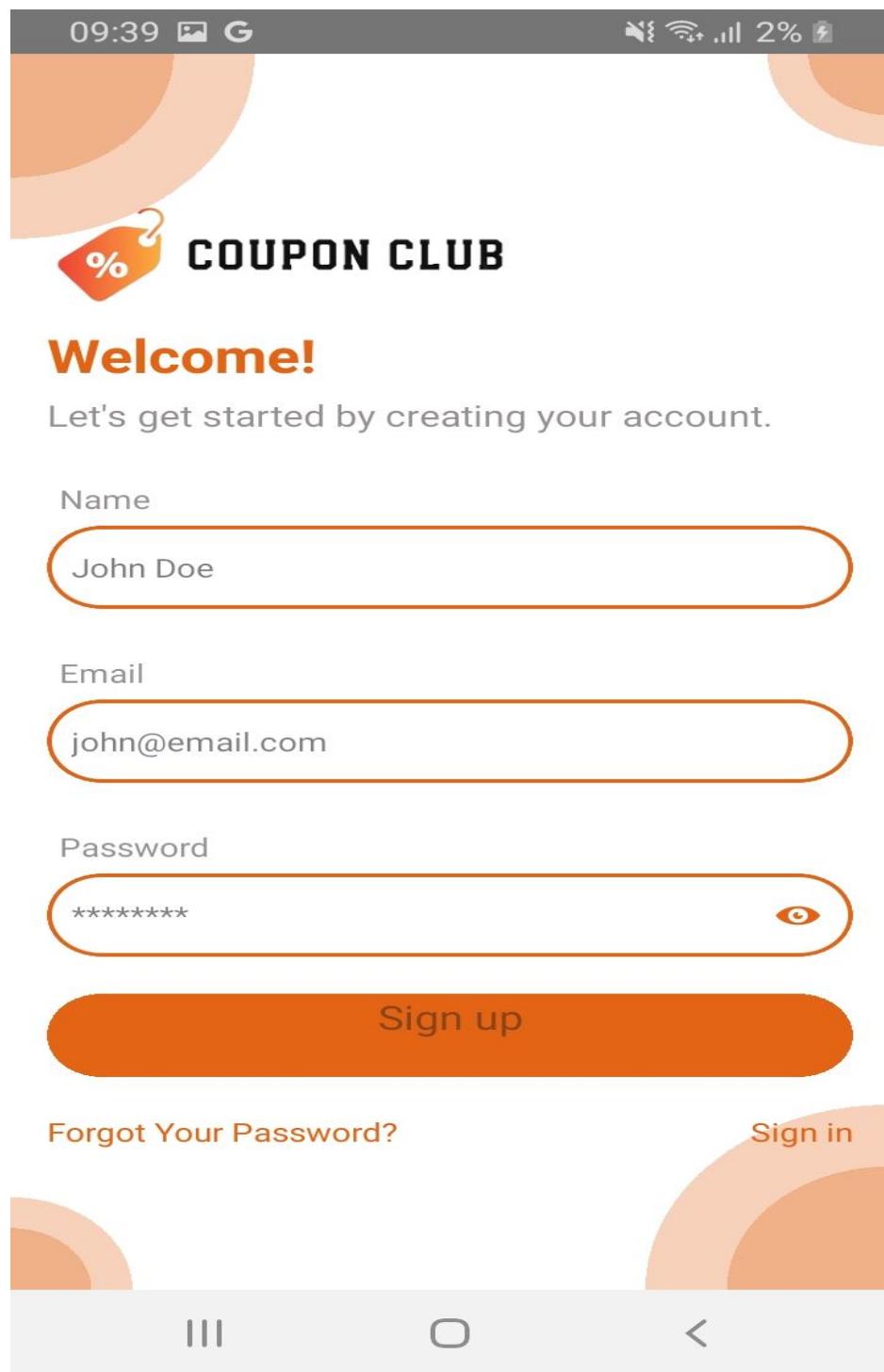
ניסיון בפיתוח אפליקציות באמצעות **React Native**, המאפשרת לך לעבוד על פרויקטים עבור Android ו OS-iOS-动员. עבודה עם Express ו -js.Node לשלם פיתוח ויישומים מבוססי שרת ניהול נתונים גמיש וקלאלבי, עם ניסיון ב- MongoDB .

12. מוסכים:

1. **הרשמה** – אפשרות להרשמה , קבלת פרטיים אישיים של המשתמש ושמירתם בבסיס הנתונים.
2. **אימות** – לאמת את האמייל של המשתמש.
3. **שחזור סיסמה** – אם המשתמש שכח את הסיסמה, הוא יכול לעדכן אותה.
4. **התחברות** – משתמש רשום יוכל להתחבר לאפליקציה.
5. **מספר בית** – מספר הבית של האפליקציה .
6. **מספר קופוניים** - מספר קופוניים של האפליקציה .
7. **מספר הוסף קופוניים** - מספר הוסף קופוניים להאפליקציה .
8. **עמוד פרופיל** - עמוד שבו אתה רואה את שמו, התמונה והקופוניים ששמרת.
9. **מספר עדקון פרופיל** - על מנת לשנות את השם, האמייל והתמונה האישית.

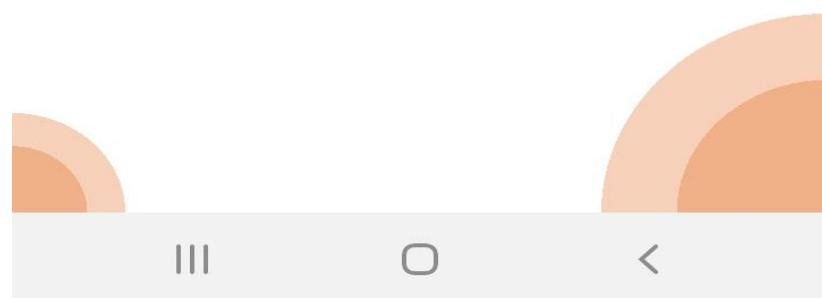
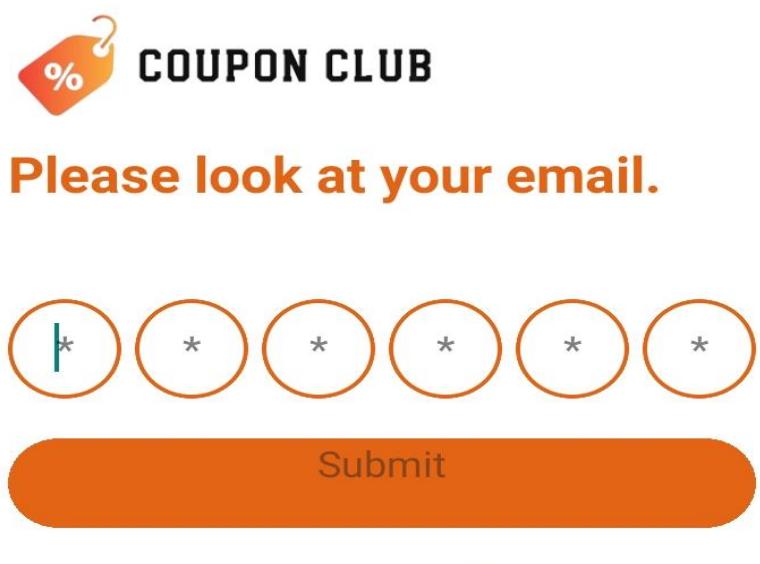
1. דף הרשמה:

כآن המשתמש מתחבר פעם ראשונה לאפליקציה ונרשם עם שם, איימל, וסיסמה.
לאחר לחיצה על כפתור הכניסה, הבקשה עוברת לשרת ושמורת את הנתונים ב MongoDB.

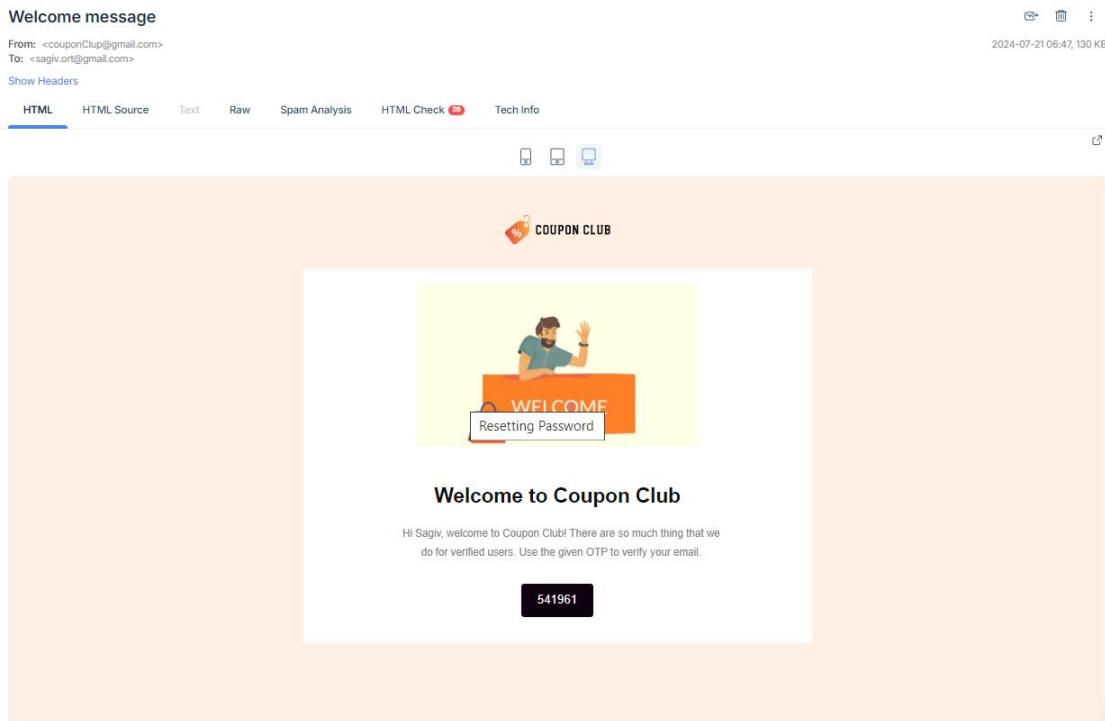


2. אימות מייל:

אחרי שהמשתמש נרשם, הוא צריך לאמת שזה המייל שלו, וזה חד פעמי, רק בהרשמה.

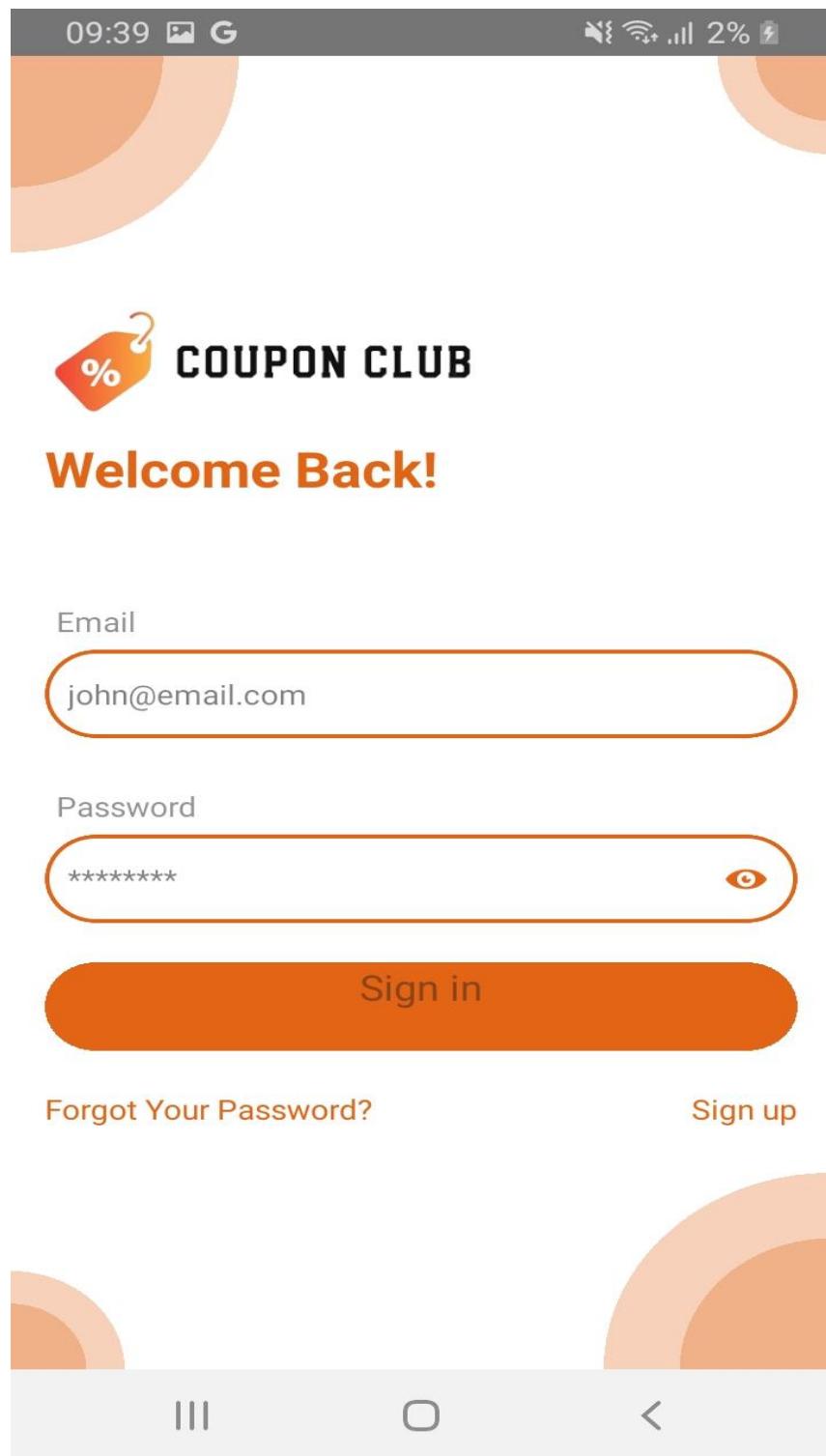


הוא שולח את הקוד ל-mailtap רק לצורך בדיקה, אחר כך אנו נשנה אותו לאימייל של המשתמשים.



3. דף התחברות :

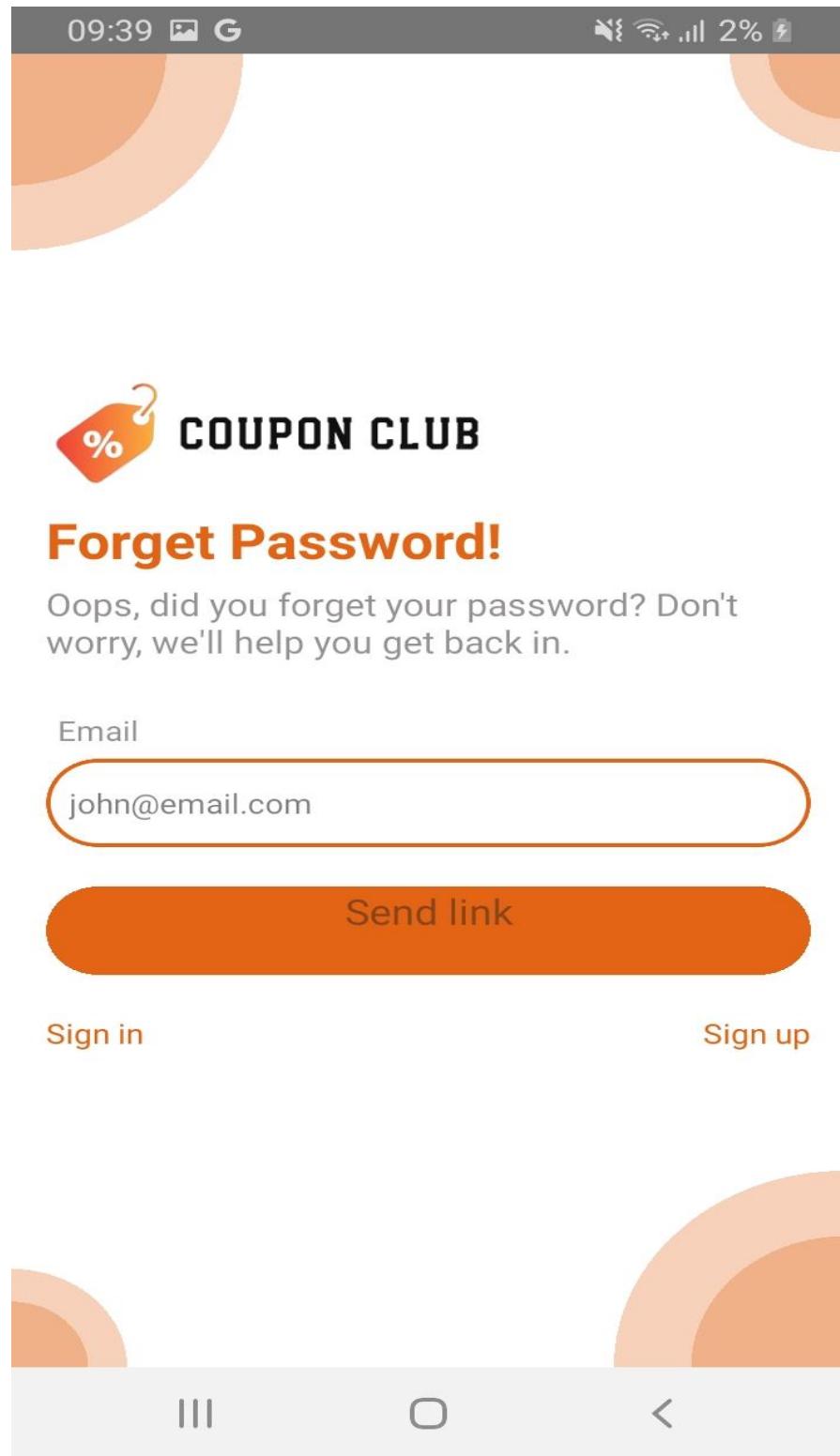
כאן המשתמש נכנס לחשבון שלו, בתנאי שיש לו כבר חשבון קיימ בבסיס נתוניים.



4. דף שחזור סיסמה :

אם המשתמש שכח את הסיסמה שלו, הוא יכול לשחזר אותה דרך האימייל.

הוא מחפש את המail אם הוא קיים, ואם הוא מוצא אותו, הוא שולח הודעה למייל לחידוש הסיסמה .



5. דף הבית:



COUPON CLUB

UP TO
25%

GET CODE

Top Coupons

5% Dior Luxury

Valid till Wed Dec 25 2024

COPY

SEE ALL

Company



iHerb

UP TO
20%

GET CODE

<

ROLEX

GET A DISCOUNT FROM
BEST COMPANY

UP TO
20%

More Coupons

MANGO

15% Mango Sale

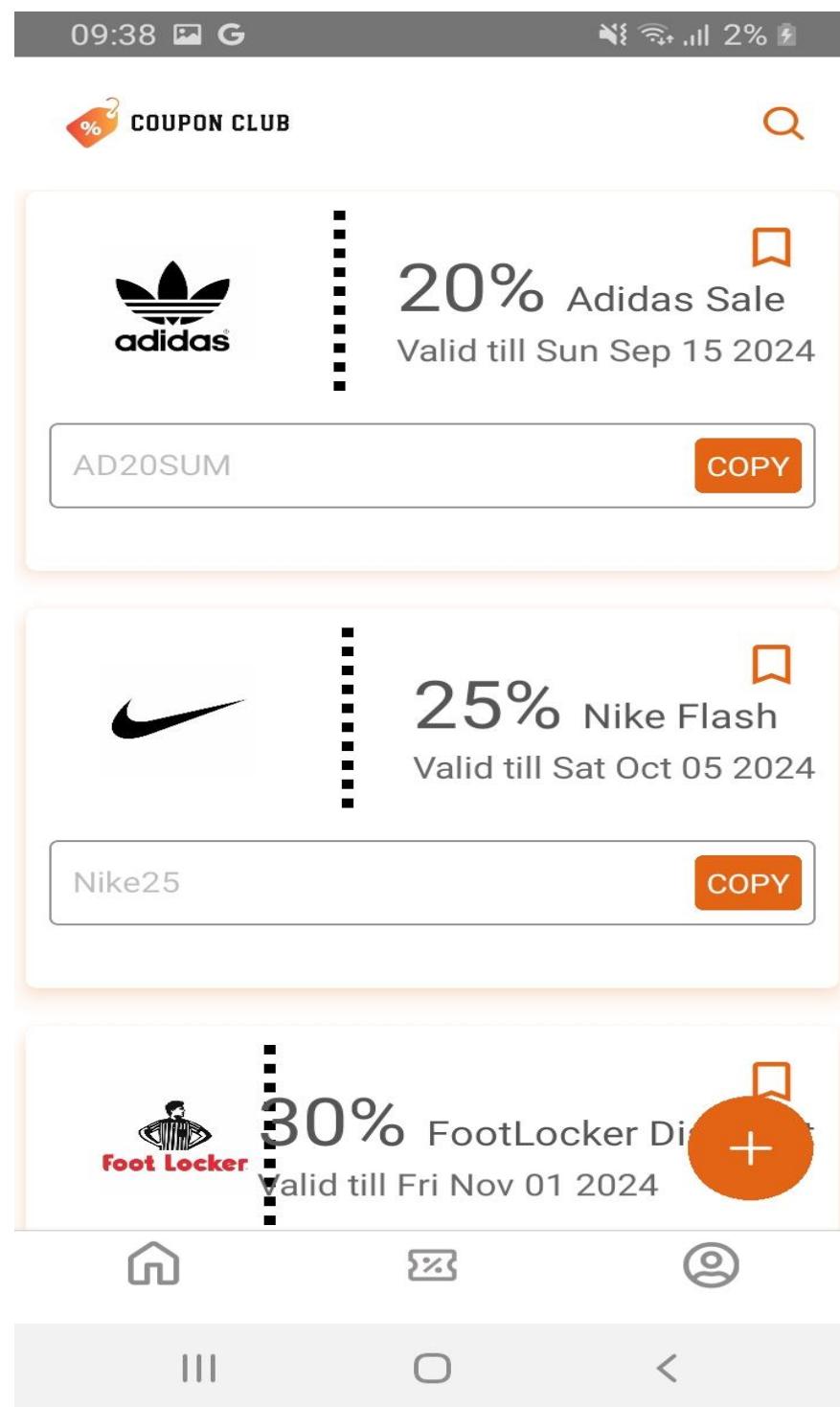
<

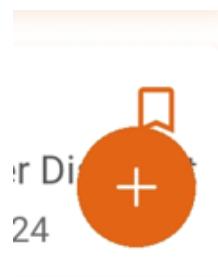
אחרי שהמשתמש מתחבר, הוא עובר לדף הבית הראשי של האפליקציה, הוא יכול לבצע מגוון פעולות, כמו כניסה לאתרי החברות, או כניסה לכל הקופונים של החברה שפורסםה בפרסום. יוכל לשמר את הקופון בדף הפרופיל שלו.

6. דף קופונים:

כאן מיצגים כל הקופונים שיש בתחום הבסיס נתונים המשתמש יכול להעתיק את הקוד קופון שירצה, ויכול לשמר אותו לתוכן ה פרופיל שלו.

וגם אפשר לחפש קופונים לפי שם החברה שהוא רוצה דרך החיפוש בדופן הדף.





אחרי לחיצה על הלחצן זהה , נפתח דף הוסף קופון שמאפשר למשתמשים לפרסום קופונים שרכשו או שהשיגו אותם במקום אחר.

7. דף ייצור קופונים:

10:03 5%

Coupon Name:

Discount Percentage:

Coupon Code:

Company Name:

Valid Date (YYYY-MM-DD):

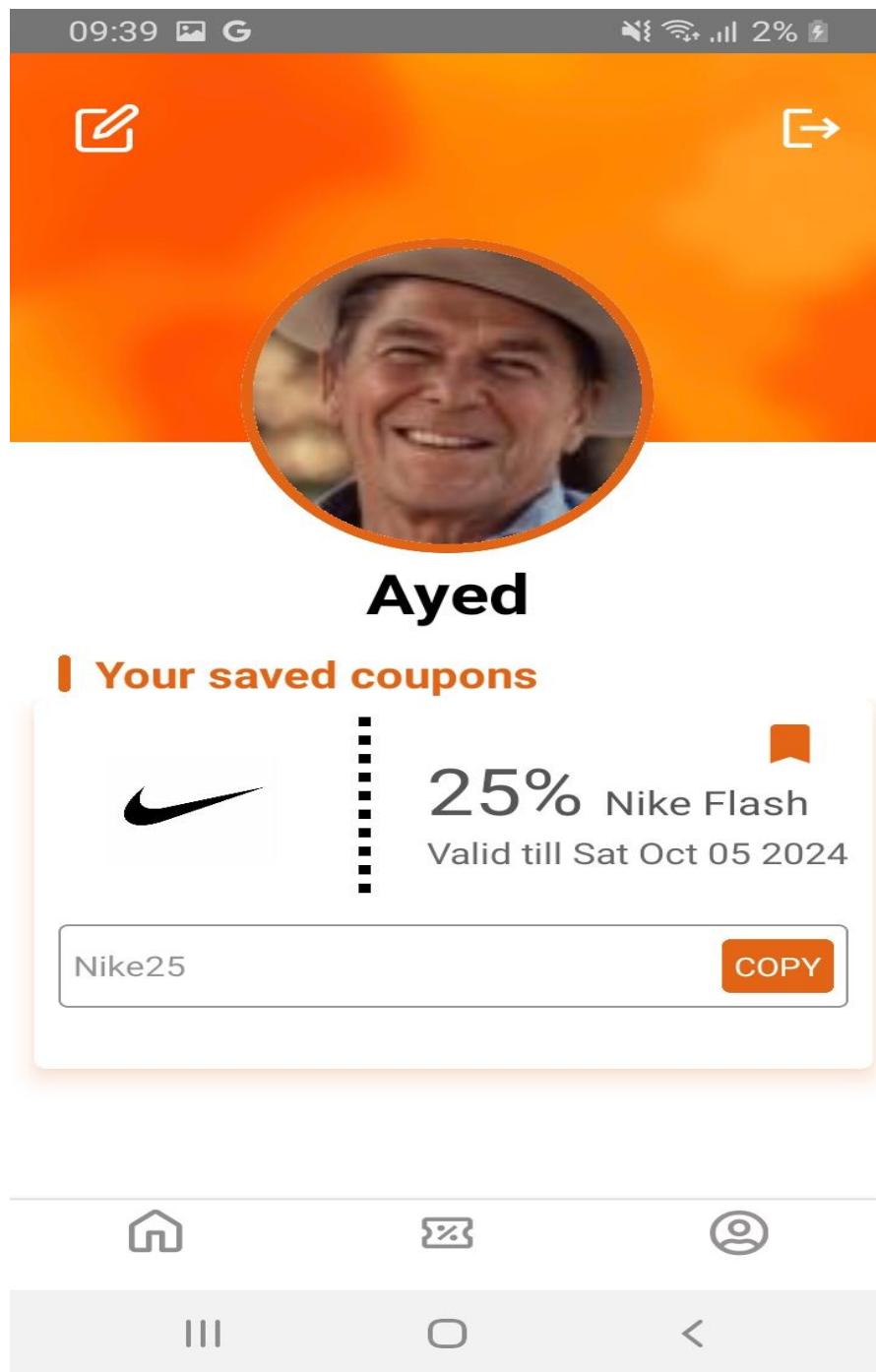
Brand: Adidas

Add Coupon

8. דף פרופיל:

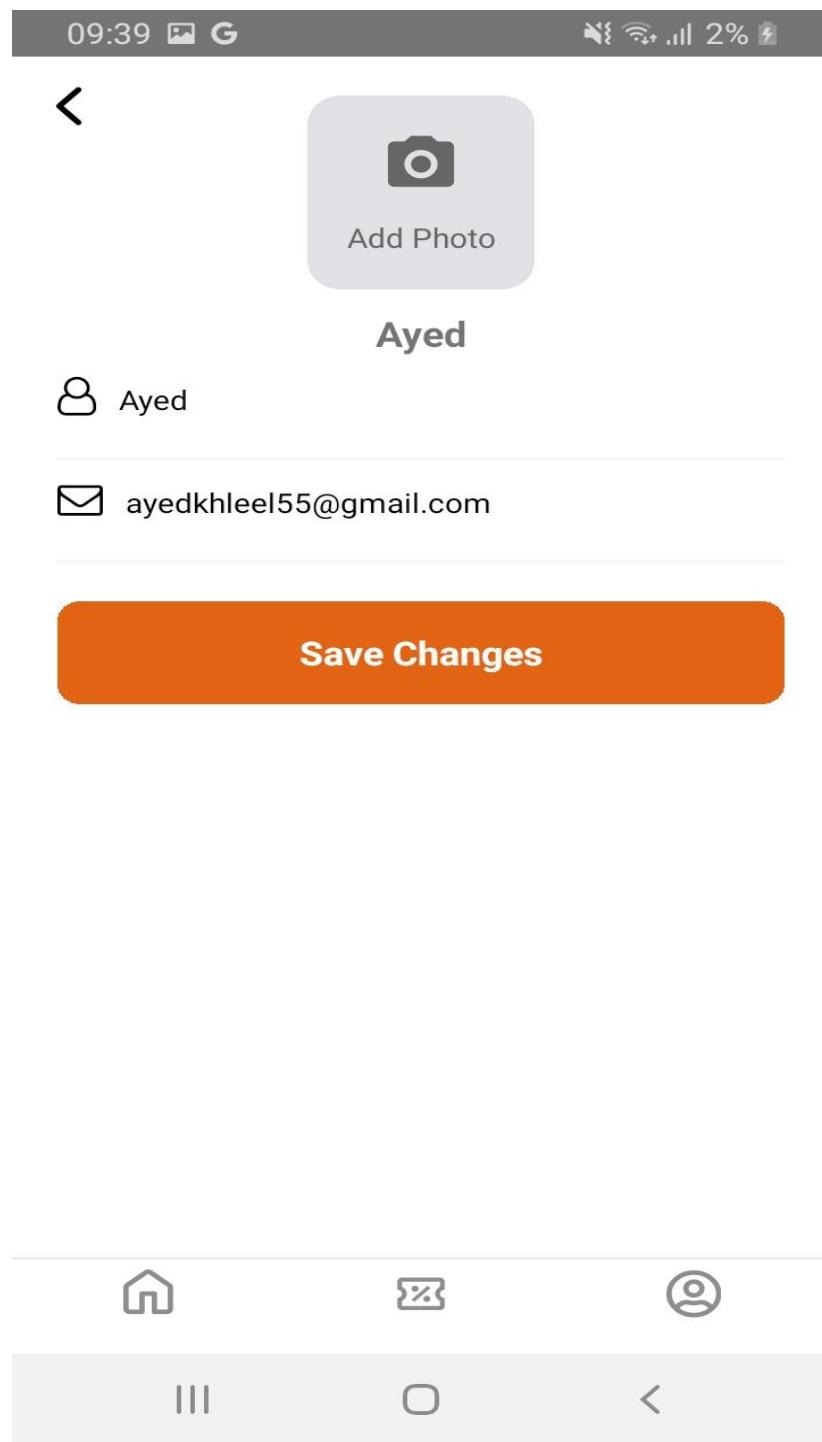
כאן מיוצג את המידע של המשתמש, וגם הקופונים שהוא שמר.

ניתן לעשות התנתקות או שינוי שם/מייל או שמירת קופון.



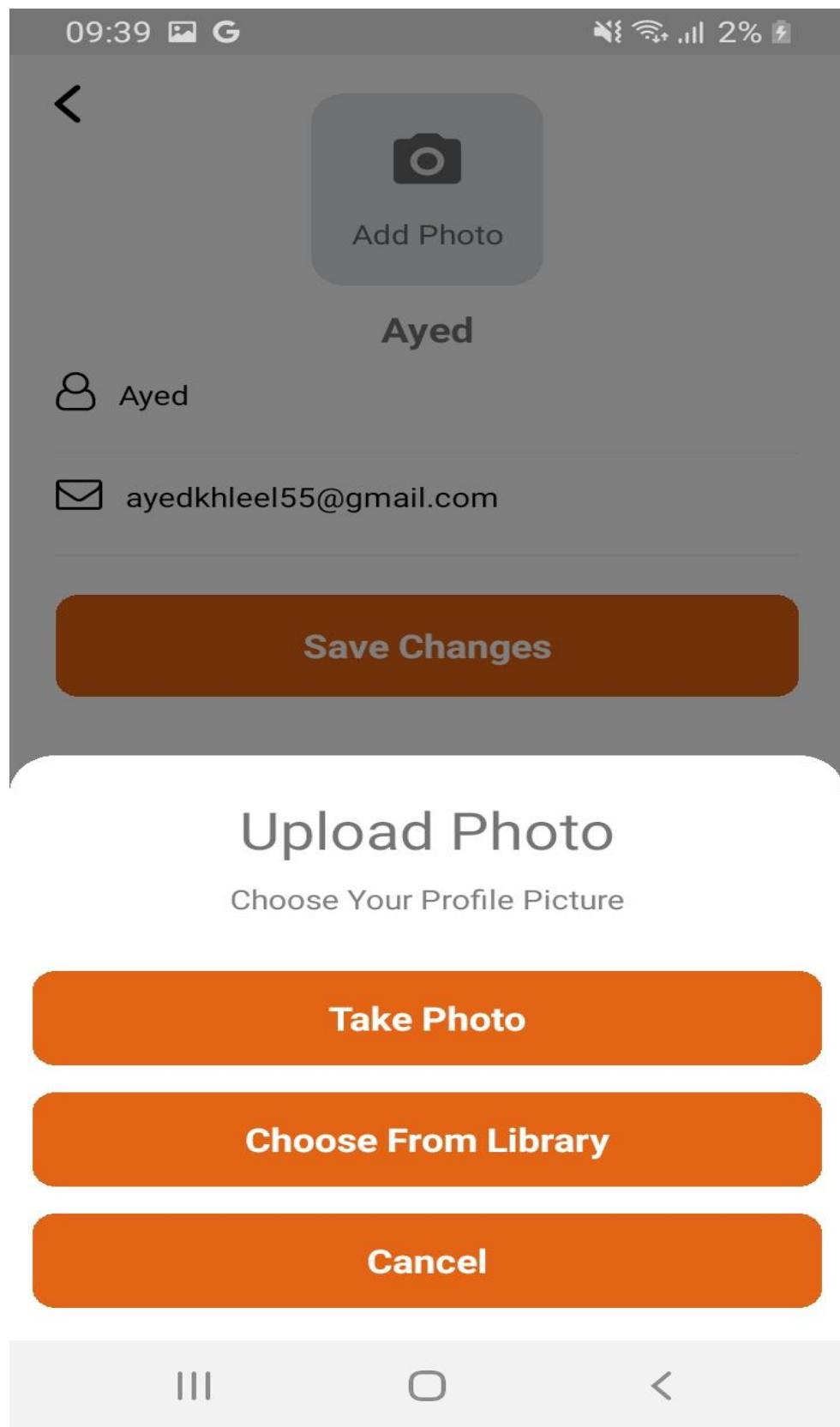
9. דף עדכון פרטיים :

אפשר להציג אליו מדף פרופיל , אם המשתמש רוצה לשנות פרטיים, כל עדכון שקורחה כאן קוררה גם בסיס נתונים.



העלאת תמונה:

המשתמש יכול לבחור, או לצלם תמונה נוכחית, או לעלות תמונה.



מודלים: 13.

זה הבסיס נתונים שומרה את השדות
כואן מופיע את השדה של הקופונים כקובץ json.

Discounts :

name:
count:
CouponsCode:
Clicked:
companyId
validDate:

Document ID	Name	Count	Coupon Code	Company ID	Valid Date	Select Brand	Image URL
_id: ObjectId('669e07b966ed2e98ca6f50ef')	Adidas Sale	20	AD20SUM	Adidas	2024-09-15T00:00:00Z	Adidas	Adidas.jpg
_id: ObjectId('669e07b966ed2e98ca6f50f0')	Nike Flash	25	Nike25	Nike	2024-10-05T00:00:00Z	Nike	Nike.jpg
_id: ObjectId('669e07b966ed2e98ca6f50f1')	FootLocker Discount	30	FL30OFF	FootLocker	2024-11-01T00:00:00Z	FootLocker	footlocker.jpg

כאן מופיע כל משתמש שנרשם לאתר ומראה האם הוא אימט את החשבון שלו או לא, לפי השדה Verified והтокין והסימנה מצופנים.

collections::

Users:

FirstName
LastName
ID
dateOfBirth
phone
email
pass

The screenshot shows the MongoDB Compass interface connected to localhost:27017. The left sidebar shows databases: admin, config, couponClub (selected), coupons, emailverificationtokens, passwordresettokens, and users. The main area shows the 'users' collection under the 'couponClub' database. There are 6 documents listed. The first document is expanded, showing its fields: _id, name, email, password, verified, favorites, points, tokens, createdAt, updatedAt, and __v. The second document is partially visible below it.

```

_id: ObjectId('66655ae212e7621cb5bd70e9')
name: "Ayed"
email: "ayedkhleel55@gmail.com"
password: "$2b$10$/q4QG7HiSJTuZzXkq9tk5.0Y/2wV4h2lcYxh7CG7rK.4A0raUU77W"
verified: true
favorites: Array (empty)
points: Array (empty)
tokens: Array (108)
createdAt: 2024-06-09T07:33:54.781+00:00
updatedAt: 2024-07-25T07:03:12.687+00:00
__v: 108

_id: ObjectId('6690520da24ebef0d06012dd')
name: "Jaaaaaaaa Doe"
email: "john.doe@example.com"
password: "$2b$10$ee22T87nfvKdNcLwAgj05u.tRpg6v6T0zqKUnML27svH.wzdGVYi2"
verified: false
favorites: Array (empty)
points: Array (empty)
tokens: Array (empty)
createdAt: 2024-07-11T21:43:41.026+00:00
updatedAt: 2024-07-11T21:43:41.026+00:00
__v: 0

_id: ObjectId('669cac9301bba4969fe6731b')
name: "Mav...g"

```

בבסיס נתונים נראה שאימל ומספר מזהה חייבים להיות
מיוחדים, וכל משתמש מסטר אמייל זה יהיה רק לו.

The screenshot shows the MongoDB Atlas interface for the 'users' collection. The 'Indexes' tab is selected, displaying two entries:

Name and Definition	Type	Size	Usage	Properties
id	REGULAR ⓘ	36.9 KB	11 (since Mon Jul 22 2024)	UNIQUE ⓘ
email_1	REGULAR ⓘ	36.9 KB	14 (since Mon Jul 22 2024)	UNIQUE ⓘ

• צד ללקוח:

types
Navigations

```

} interface NewUserResponse
;id: string
;name: string
;email: string
{

} = export type AuthStackParamList
;SignIn: undefined
;SignUp: undefined
;LostPassword: undefined
;HomePage: undefined
;CouponPage: undefined
;Verification: {userInfo: NewUserResponse}
;ProfilePage: undefined
;{

```

Api
`;import axios from 'axios'`

```

})const client = axios.create
,'baseURL: 'http://192.168.96.1:2717
;({
```

`;export default client`
Components
`;import {FC, ReactNode, useEffect} from 'react'`
`;import AppInput from '../ui/AppInput'`
`;import colors from '../utils/colors'`
`;import {useFormikContext} from 'formik'`
`} import`
`,StyleSheet`
`,View`

```

,Text
,Keyboard
,TextInputProps
,StyleProp
,ViewStyle
,Pressable
;'from 'react-native {
} ,import Animated
,useAnimatedStyle
,useSharedValue
,withSequence
,withSpring
,withTiming
;'from 'react-native-reanimated {

} interface Props
;name: string
;label?: string
;placeholder?: string
;keyboardType?: TextInputProps['keyboardType']
;autoCapitalize?: TextInputProps['autoCapitalize']
;secureTextEntry?: boolean
;<containerStyle?: StyleProp<ViewStyle
;rightIcon?: ReactNode
;onRightIconPress?(): void
{



} <= const AuthInputField: FC<Props> = props
;(0)const inputStyleTransformValue = useSharedValue
const {handleChange, values, errors, handleBlur,
}>touched} = useFormikContext
;[key: string]: string
;()<

} const
,label
,placeholder
,autoCapitalize
,keyboardType
,secureTextEntry
,containerStyle
,name
,rightIcon
,onRightIconPress
;props = {

```

```

const errorMsg = touched[name] && errors[name] ?
;" : errors[name]

} <= () = const shakeUI
)inputStyleTransFormValue.value = withSequence
,withTiming(-10, {duration: 50})
} ,withSpring(0
,damping: 8
,mass: 0.5
,stiffness: 1000
,restDisplacementThreshold: 0.1
,({
;((
;{

} <= ()const inputStyle = useAnimatedStyle
} return
transform: [{translateX:
,inputStyleTransFormValue.value}]
;{
;({


} <= ())useEffect
;()if (errorMsg) shakeUI
;([errorMsg],{


) return
<Animated.View style={[containerStyle, inputStyle]}>
<View style={styles.labelContainer}>
<Text style={styles.label}>{label}</Text>
<Text style={styles.errorMsg}>{errorMsg}</Text>
<View/>
<View>
AppInput>
placeholder={placeholder}
keyboardType={keyboardType}
autoCapitalize={autoCapitalize}
secureTextEntry={secureTextEntry}
onChangeText={handleChange(name)}
value={values[name]}
onBlur={handleBlur(name)}
</
) ? rightIcon}
Pressable onPress={onRightIconPress} >
<style={styles.rightIcon}>{rightIcon}</Pressable

```

```

{null : (
<View/>
<Animated.View/>
;(
;{

})const styles = StyleSheet.create
} :labelContainer
,'flexDirection: 'row
,'alignItems: 'center
,'justifyContent: 'space-between
,padding: 5
,{
} :label
,color: colors.CONTRAST
,{
} :errorMsg
,color: colors.ERROR
,{
} :rightIcon
,width: 45
,height: 45
,'position: 'absolute
,top: 0
,right: 0
,'justifyContent: 'center
,'alignItems: 'center
,{
;({


;export default AuthInputField

;'import React, { ReactNode } from 'react'
;'import { Formik, FormikHelpers } from 'formik'
;'import { StyleSheet, View } from 'react-native'

} <interface Props<T
;initialValues: any
;validationSchema: any
onSubmit(values: T, formikHelpers: FormikHelpers<T>):
;void
;children: ReactNode
{

} <= const Form = <T extends object>(props: Props<T>)

```

```

) return
Formik>
onSubmit={props.onSubmit}
initialValues={props.initialValues}
<validationSchema={props.validationSchema}>
{props.children}
</Formik>
;((
;{

;export default Form

;import { useFormikContext } from 'formik'
;import { FC } from 'react'
;import { StyleSheet } from 'react-native'
;import AppButton from '../ui/AppButton

} interface Props
:title: string

{

} <= const SubmitBtn: FC<Props> = props
;()const {handleSubmit, isSubmitting} = useFormikContext
return <AppButton busy={isSubmitting}>
;</onPress={handleSubmit} title={props.title}>
;{

})const styles = StyleSheet.create
,{} :container
;({

;export default SubmitBtn

;import React, {useState} from 'react'
;import {Picker} from '@react-native-picker/picker

} import
,View
,Text
,TextInput
,TouchableOpacity

```

```

,StyleSheet
,Alert
;'from 'react-native {
;'import client from '../api/client'
;'import colors from '../utils/colors

} interface AddCouponFormProps
;onCouponAdded: (coupon: any) => void
{

const AddCouponForm: React.FC<AddCouponFormProps>
}<= ({onCouponAdded})
;(")const [name, setName] = useState
;(")const [count, setCount] = useState
;(")const [couponCode, setCouponCode] = useState
;(")const [CompanyId, setCompanyId] = useState
;(")const [ValidDate, setValidDate] = useState
;(")const [selectBrand, setSelectedBrand] = useState

}<= () const handleAddCoupon = async
) if
|| name!
|| count!
|| couponCode!
|| CompanyId!
|| ValidDate!
selectBrand!
)(
;Alert.alert('Error', 'Please fill out all fields')
;return
{

} = const coupon
,name
,count: parseInt(count)
,couponCode
,CompanyId
,ValidDate
,selectBrand
;{

} try
const {data, status} = await
;client.post('/coupons/create', coupon)

} if (status === 200)

```

```

;Alert.alert('Success', 'Coupon added successfully')
;onCouponAdded(data)
Clear form fields //
;(")setName
;(")setCount
;(")setCouponCode
;(")setCompanyId
;(")setValidDate
} else {
;console.error('Unexpected response:', status, data)
;Alert.alert('Error', 'Failed to add coupon')
{
} (catch (error: any {
)console.error
,:Error adding coupon'
,error.response?.data || error.message
;(
)Alert.alert
,'Error'
} ${ :Failed to add coupon`}
error.response?.data.message || error.message
,{{
;(
{
;
;

) return
<View style={styles.card}>
<Text style={styles.label}>Coupon Name:</Text>
TextInput style={styles.input} value={name} >
</ onChangeText={setName}

<Text style={styles.label}>Discount Percentage:</Text>
TextInput>
style={styles.input}
value={count}
onChangeText={setCount}
"keyboardType="numeric
</

<Text style={styles.label}>Coupon Code:</Text>
TextInput>
style={styles.input}
value={couponCode}
onChangeText={setCouponCode}
</

```

```

<Text style={styles.label}>Company Name:</Text>
<TextInput>
  style={styles.input}
  value={CompanyId}
  onChangeText={setCompanyId}
</>

Text style={styles.label}>Valid Date (YYYY-MM->
<DD):</Text>
<TextInput>
  style={styles.input}
  value={ValidDate}
  onChangeText={setValidDate}
</>

<Text style={styles.label}>Brand:</Text>
<Picker>
  selectedValue={selectBrand}
  onValueChange={itemValue =>
    setSelectedBrand(itemValue)}
  <style={styles.picker}>
    </ "Picker.Item label="Adidas" value="Adidas">
    </ "Picker.Item label="Nike" value="Nike">
    </ "Picker.Item label="FootLocker" value="FootLocker">
    </ "Picker.Item label="Timberland" value="Timberland">
    </ "Picker.Item label="Dior" value="Dior">
    Picker.Item label="Giorgio Armani" value="Giorgio ">
    </ "Armani
    </ "Picker.Item label="H&M" value="H&M">
    </ "Picker.Item label="iHerb" value="iHerb">
    </ "Picker.Item label="Lacoste" value="Lacoste">
    </ "Picker.Item label="Levis" value="Levis">
    </ "Picker.Item label="Mango" value="Mango">
    </ "Picker.Item label="Polo" value="Polo">
    </ "Picker.Item label="Rolex" value="Rolex">
    </ "Picker.Item label="Sacara" value="Sacara">
    </ "Picker.Item label="Tommy" value="Tommy">
    Picker.Item label="Victoria's Secret" value="Victoria's ">
    </ "Secret
    </ "Picker.Item label="Zara" value="Zara">
  <Picker/>

  TouchableOpacity style={styles.button} >
    <onPress={handleAddCoupon}>
      <Text style={styles.buttonText}>Add Coupon</Text>

```

```
<TouchableOpacity>
<View/>
;(
;{

})const styles = StyleSheet.create
} :card
, width: 350
, padding: 20
, borderRadius: 5
, backgroundColor: colors.PRIMARY
, shadowColor: colors.SECONDARY
, shadowOffset: {width: 0, height: 4}
, shadowOpacity: 0.2
, shadowRadius: 6
, elevation: 8
, marginBottom: 20
,{ 
} :label
, fontSize: 16
, color: colors.CONTRAST
, marginBottom: 8
,{ 
} :input
, height: 40
, borderColor: colors.SECONDARY
, borderWidth: 1
, marginBottom: 20
, paddingHorizontal: 10
, borderRadius: 5
,{ 
} :button
, backgroundColor: colors.SECONDARY
, paddingVertical: 10
, 'alignItems': 'center'
, borderRadius: 5
,{ 
} :buttonText
, 'color': '#fff
, fontSize: 18
,{ 
} :picker
, height: 40
, borderColor: colors.SECONDARY
, borderWidth: 1
, marginBottom: 20
```

```

,borderRadius: 5
,{
;({}

;export default AddCouponForm

;'import {FC, ReactNode} from 'react'
;'import {SafeAreaView, StyleSheet} from 'react-native'

} interface Props
;children: ReactNode
{

} <= const AppCotnainer: FC<Props> = ({children})
return <SafeAreaView
;<style={styles.container}>{children}</SafeAreaView
;{

})const styles = StyleSheet.create
} :container
,flex: 1
,{
;({

;export default AppCotnainer

;'import React, {Children, FC, ReactNode} from 'react'
;'import {StyleSheet, View, Image, Text} from 'react-native'
;'import CircleUi from '../ui/CircleUi'
;'import colors from '../utils/colors

} interface Props
;children: ReactNode
;heading?: string
;subHeading?: string
{

const AuthFormContainer: FC<Props> =
} <= ({children,heading,subHeading})
) return
<View style={styles.container}>
</ {200}=CircleUi position="top-left" size>
</ {100}=CircleUi position="top-right" size>
</ {100}=CircleUi position="bottom-left" size>

```

```
</ {200}=CircleUi position="bottom-right" size>

<View style={styles.headerContainer}>
  Image style={styles.logo} >
    </ source={require('../assets/logo.png')}>
  Text>
  <style={styles.heading}>
    {heading}
  <Text/>
  <Text style={styles.subHeading}>
    {subHeading}
  <Text/>
  <View/>

  {children}
  <View/>
  ;(
  ;{

})const styles = StyleSheet.create
}:container
,flex: 1
,backgroundColor: colors.PRIMARY
,'alignItems: 'center
,'justifyContent: 'center
,paddingHorizontal: 15
,{
} :headerContainer
,'width: '100%
,marginBottom: 20
,{
}:heading
,color: colors.SECONDARY
,fontSize: 25
,'fontWeight: 'bold
,paddingVertical: 5
,{
} :logo
,width: 190
,height: 65
,{
}:subHeading
,color: colors.CONTRAST
,fontSize: 16
,{
};({
```

```

;export default AuthFormContainer

;'import React from 'react
import { View, FlatList, Image, StyleSheet,
;'TouchableOpacity, Text, Linking } from 'react-native'
;'import colors from './utils/colors

Define the Category interface with the link property //
} interface Category
:id: string
image: any; // Using 'any' type here for the source due to
dynamic import handling
;link: string
{

Sample categories data with links //
]= []const categories: Category
id: '1', image: require('../assets/tommy.jpg'), link: }
,{ "https://www.tommy.com
id: '2', image: require('../assets/dior.jpg'), link: }
,{ "https://www.dior.com
id: '3', image: require('../assets/Adidas.jpg'), link: }
,{ "https://www.adidas.com
id: '4', image: require('../assets/footLocker.jpg'), link: }
,{ "https://www.footlocker.com
id: '5', image: require('../assets/GA.jpg'), link: }
,{ "https://www.armani.com
id: '6', image: require('../assets/iHerb.jpg'), link: }
,{ "https://www.iherb.com
id: '7', image: require('../assets/lacoste.jpg'), link: }
,{ "https://www.lacoste.com
id: '8', image: require('../assets/mango.jpg'), link: }
,{ "https://www.mango.com
id: '9', image: require('../assets/Nike.jpg'), link: }
,{ "https://www.nike.com
id: '10', image: require('../assets/H&M.jpg'), link: }
,{ "https://www.hm.com
id: '11', image: require('../assets/polo.jpg'), link: }
,{ "https://www.ralphlauren.com
id: '12', image: require('../assets/sacara.jpg'), link: }
,{ '/https://www.sacara.co.il
id: '13', image: require('../assets/Timberland.jpg'), link: }
,{ "https://www.timberland.com

```

```

id: '14', image: require('../assets/levis.jpg'), link: }
,{ "https://www.levi.com
id: '15', image: require('../assets/VS.jpg'), link: }
,{ "https://www.victoriassecret.com
id: '16', image: require('../assets/zara.jpg'), link: }
,{ "https://www.zara.com
;[

```

Function to open the URL //

```

} <= const openLink = (url: string)
Linking.openURL(url).catch(err => console.error("Couldn't
load page", err))
;{

```

Component to render each category item //

```

const CategoryItem: React.FC<{ item: Category }> = ({ item
) <= })
<TouchableOpacity>
  style={styles.itemContainer}
  onPress={() => openLink(item.link)} // Handle press to
  open the link
<
</Image source={item.image} style={styles.image}>
<TouchableOpacity>
;(

```

Main component to render the category list //

```

} <= () = const CategoryList: React.FC
) return
<View style={styles.container}>
<Text style={styles.title}>| Company</Text>
<View>
<FlatList>
  data={categories}
  renderItem={({ item }) => <CategoryItem item={item}
/>}
  keyExtractor={item => item.id}
  horizontal
  showsHorizontalScrollIndicator={false}
  contentContainerStyle={styles.listContainer}
</
<View/>
<View/>
;(
;{

```

```

Styles for the component //
})const styles = StyleSheet.create
} :container
,paddingVertical: 20
,backgroundColor: colors.PRIMARY
,{
} :listContainer
,paddingHorizontal: 10
,{
} :itemContainer
,borderRadius: 5
,backgroundColor: colors.PRIMARY
,padding: 10
,'position: 'relative
,shadowColor: colors.SECONDARY
,shadowOffset: { width: 0, height: 4 }
,shadowOpacity: 0.2
,shadowRadius: 6
,elevation: 8
,'alignItems: 'center
,'justifyContent: 'center
,marginRight: 10
,{
} :image
,width: 100
,height: 70
borderRadius: 7, // Optional: Add border radius for
rounded corners
,{
} :title
,fontSize: 24
,'fontWeight: 'bold
,paddingHorizontal: 10
,marginBottom: 10
,'color: 'black
,{
};({

;export default CategoryList

;'import React, {FC, useState} from 'react
} import
,View
,Text
,Image

```

```

,TextInput
,TouchableOpacity
,StyleSheet
,Alert
,ImageSourcePropTypes
;'from 'react-native {
;import Clipboard from '@react-native-clipboard/clipboard'
import AsyncStorage from '@react-native-async-
;storage/async-storage
;import colors from '../utils/colors'
;import Icon from 'react-native-vector-icons/MaterialIcons'
;import imageMap from '../utils/imageMap

} interface Coupon
:id: string_
;name: string
;count: number
;couponCode: string
;CompanyId: string
;ValidDate: Date
;selectBrand: string
;imageUrl: string
{

} interface CouponCardProps
;coupon: Coupon
{

<= const CouponCard: FC<CouponCardProps> = ({coupon})
}
;const [buttonText, setButtonText] = useState('COPY')
const [showSavedIcon, setShowSavedIcon] =
;useState(false)
= const imageUrl: ImageSourcePropTypes | undefined
;imageMap[coupon.imageUrl]

} <= const copyIt = (code: string)
;Clipboard.setString(code)
;setButtonText('COPIED')
)Alert.alert
,'Copied to Clipboard'
,!Coupon code ${code} copied to clipboard'
;(
;{

} <= () const saveCoupon = async

```

```
    } try
  const savedCoupons = await
  AsyncStorage.getItem('savedCoupons')
  const coupons = savedCoupons ?
  [] : JSON.parse(savedCoupons)
  coupons.push(coupon)
  await AsyncStorage.setItem('savedCoupons',
  JSON.stringify(coupons))
  setShowSavedIcon(true); // Show bookmark icon
  setTimeout(() => setShowSavedIcon(false), 3000); //
  Hide bookmark icon after 2 seconds
} (catch (error {
  Alert.alert('Error', 'Failed to save the coupon.')
{
;
;

) return
<View style={styles.cardWrapper}>
<View style={styles.card}>
  TouchableOpacity onPress={saveCoupon} >
    <style={styles.saveBar}
      ) ? showSavedIcon}
      Icon name="bookmark" size={28} >
      </ color={colors.SECONDARY}
    ) : (
      Icon name="bookmark-border" size={28} >
      </ color={colors.SECONDARY}
    {
      <TouchableOpacity>
      <View style={styles.main}>
        <View style={styles.colImg}>
          Image source={imageSource} style={styles.image}>
        </
        <View/>
        </ View style={styles.vertical}>
        <View style={styles.content}>
          <Text style={styles.h1}>
            {coupon.count}% <Text
            <style={styles.span}>{coupon.name}</Text
            <Text/>
            <Text style={styles.p}>
              Valid till {new
              }()Date(coupon.ValidDate).toDateString
              <Text/>
            <View/>
            <View/>
```

```
<View style={styles.copyButton}>
  <TextInput>
    style={styles.input}
    value={coupon.couponCode}
    editable={false}
  </>
  <TouchableOpacity>
    onPress={() => copyIt(coupon.couponCode)}
    style={styles.button}
    <Text style={styles.buttonText}>{buttonText}</Text>
    <TouchableOpacity/>
  <View/>
  <View/>
  <View/>
  ;(
  ;{

})const styles = StyleSheet.create
} :cardWrapper
,marginBottom: 20
,{
} :card
, width: 350
, height: 200
, borderRadius: 5
, backgroundColor: colors.PRIMARY
, padding: 10
, 'position': 'relative'
, shadowColor: colors.SECONDARY
, shadowOffset: {width: 0, height: 4}
, shadowOpacity: 0.2
, shadowRadius: 6
, elevation: 8
,{
} :saveBar
, 'position': 'absolute'
, top: 10
, right: 10
, padding: 5
,{
} :main
, 'flexDirection': 'row'
, 'justifyContent': 'space-between'
, 'alignItems': 'center'
,{
} :colmg
```

```
,marginRight: 10
,{
} :image
,width: 70
,height: 60
,left: 20
,{
} :vertical
,borderLeftWidth: 5
,'borderLeftColor: 'black
,'borderStyle: 'dotted
,height: 100
,{
} :content
,marginLeft: -20
,{
} :h1
,fontSize: 35
,'color: '#565656
,{
} :span
,fontSize: 18
,{
} :p
,fontSize: 16
,'color: '#696969
,{
} :copyButton
,'flexDirection: 'row
,'justifyContent: 'space-between
,'alignItems: 'center
,marginTop: 12
,height: 45
,borderRadius: 4
,borderWidth: 1
,borderColor: colors.CONTRAST
,paddingHorizontal: 5
,{
} :input
,flex: 1
,'height: '100%
,fontSize: 15
,{
} :button
,padding: 5
,backgroundColor: colors.SECONDARY
```

```
,borderRadius: 4
,'justifyContent: 'center
,'alignItems: 'center
,{
} :buttonText
,'color: '#fff
,{
;({

;export default CouponCard

;'import React from 'react
;'import { TouchableOpacity, StyleSheet } from 'react-native'
;'import { useNavigation } from '@react-navigation/native'
;'import Icon from 'react-native-vector-icons/Feather'
;'import colors from '../utils/colors

} <= () = const EditBar: React.FC
;()const navigation = useNavigation

} <= () = const handleEditProfile
navigation.navigate('EditProfile'); // Navigate to the edit
profile page
;{

) return
TouchableOpacity style={styles.editButton} >
<onPress={handleEditProfile}
</ Icon name="edit" size={28} color={colors.PRIMARY}>
<TouchableOpacity>
;(
;{

})const styles = StyleSheet.create
} :editButton
,'position: 'absolute
,top: 20
,left: 20
,padding: 6
,{
;({

;export default EditBar
```

```
Header.tsx //  
;import React, { useState } from 'react'  
import { View, TextInput, Image, StyleSheet,  
;TouchableWithoutFeedback } from 'react-native'  
;import Icon from 'react-native-vector-icons/Octicons'  
;import colors from '../utils/colors'  
  
} interface HeaderProps  
;search: string  
;updateSearch: (search: string) => void  
{  
  
const Header: React.FC<HeaderProps> = ({ search,  
} <= updateSearch })  
const [showearchBar, setShowearchBar] =  
;useState<boolean>(false)  
  
} <= () = const toggleSearchBar  
;setShowearchBar(!showearchBar)  
;{  
  
) return  
<View style={styles.logoBar}>  
Image source={require('../assets/logo.png')} >  
</ "style={styles.logolImage} resizeMode="contain  
</ View style={styles.spacer}>  
TouchableWithoutFeedback >  
<onPress={toggleSearchBar}  
Icon name="search" size={20} color="#E36414" >  
</ style={styles.searchIcon}  
<TouchableWithoutFeedback/>  
) && showearchBar}  
<View style={styles.searchBar}>  
TextInput>  
style={styles.searchInput}  
"placeholder="Search  
onChangeText={updateSearch}  
value={search}  
"placeholderTextColor="gray  
</  
<View/>  
{(  
<View/>  
;(  
;{
```

```

})const styles = StyleSheet.create
} :logoBar
,'flexDirection: 'row
,'alignItems: 'center
,'backgroundColor: '#fff
,paddingVertical: 10
,paddingHorizontal: 20
,'width: '100%
,{

} :logolImage
,width: 100
,height: 50
,{

} :spacer
,flex: 1
,{

} :searchBar
,'flexDirection: 'row
,'alignItems: 'center
,'backgroundColor: '#F0F0F0
,borderWidth: 2
,height: 40
,borderRadius: 16
,paddingVertical: 5
,paddingHorizontal: 10
,borderColor: colors.SECONDARY
,flex: 2
,{

} :searchIcon
,marginLeft: 10
,{

} :searchInput
,flex: 1
,height: 40
,'color: 'black
,{

};({


;export default Header

```

LogoutButton.tsx //

```

;'import React from 'react'
;'import {TouchableOpacity, StyleSheet} from 'react-native'
;'import Icon from 'react-native-vector-icons/MaterialIcons

```

```

;'import colors from '../utils/colors'
;'import {useAuth} from '../utils/AuthContext'
import {NavigationProp, useNavigation} from '@react-
;navigation/native
;'import {AuthStackParamList} from '../@types/navigation
;'import {updateLoggedInState} from '../store/auth
;'import {useDispatch} from 'react-redux
} interface LogoutButtonProps
style?: object; // Add more specific styles if needed
{

const LogoutButton: React.FC<LogoutButtonProps> =
} <= ({style})
const navigation =
;()<<useNavigation<NavigationProp<AuthStackParamList
;()const {signOut} = useAuth
;()const dispatch = useDispatch

} <= () = const handleSignOut
;()signOut
;dispatch(updateLoggedInState(false))

})navigation.reset
,index: 0
,routes: [{name: 'SignIn'}]
;(
;{

) return
TouchableOpacity>
onPress={handleSignOut}
<style={[styles.logoutButton, style]}>
</Icon name="logout" size={28} color={'white'}>
<TouchableOpacity>
;(
;{

})const styles = StyleSheet.create
} :logoutButton
,'position: 'absolute
,top: 20
,right: 10
,padding: 6
,{
;(

```

```

;export default LogoutButton

navigation

import {createNativeStackNavigator} from '@react-
;navigation/native-stack
;import SignIn from '../views/auth/SignIn
;import SignUp from '../views/auth/SignUp
;import LostPassword from '../views/auth/LostPassword
;import Verification from '../views/auth/Verification
;import {AuthStackParamList} from '../@types/navigation
;import {useSelector} from 'react-redux
;import {getAuthState} from '../store/auth
;import HomePage from '../views/auth/HomePage
;import EditProfileScreen from '../views/auth/EditProfile
;import ProfilePage from '../views/auth/ProfilePage
;import CouponPage from '../views/auth/CouponPage

const Stack =
;()<createNativeStackNavigator<AuthStackParamList

} <= () = const AuthNavigator
;const authState = useSelector(getAuthState)
) return
Stack.Navigator screenOptions={{headerShown: >
<false}}
</ Stack.Screen name="SignIn" component={SignIn}>
</ Stack.Screen name="SignUp" component={SignUp}>
Stack.Screen name="LostPassword" >
</ component={LostPassword}
Stack.Screen name="Verification" >
</ component={Verification}
Stack.Screen name="ProfilePage" >
</ component={ProfilePage}
Stack.Screen name="HomePage" >
</ component={HomePage}
Stack.Screen name="CouponPage" >
</component={CouponPage}
<Stack.Navigator>
;(
;{

;export default AuthNavigator

```

```

import {DefaultTheme, NavigationContainer} from '@react-
;navigation/native
;import {FC, useEffect} from 'react
;import {useDispatch, useSelector} from 'react-redux
} import
,getAuthState
,updateBusyState
,updateLoggedInState
,updateProfile
:'from '../store/auth {
;import AuthNavigator from './AuthNavigator
;import TabNavigator from './TapNavigator
import {Keys, getFromAsyncStorage} from
;"./utils/asyncStorage
;import client from './api/client
;import {View} from 'react-native
;import Loader from '../ui/Loader
;import colors from '../utils/colors

{} interface Props

} = const AppTheme
,DefaultTheme...
} :colors
,DefaultTheme.colors...
,background: colors.PRIMARY
primary:colors.CONTRAST
,{

;{

} <= const AppNavigator: FC<Props> = props
;const {loggedIn, busy} = useSelector(getAuthState)
;()const dispatch = useDispatch

} <= ()useEffect
} <= () const fetchAuthInfo = async
;dispatch(updateBusyState(true))
} try
const token = await
;getFromAsyncStorage(Keys.AUTH_TOKE)
} if (!token)
;return dispatch(updateBusyState(false))
{

} ,const {data} = await client.get('/auth/is-auth
} :headers

```

```

,Authorization: 'Bearer ' + token
,{(
;({(
;dispatch(updateProfile(data.profile))
;dispatch(updateLoggedInState(true))
} (catch (error {
;console.log('Auth error: ', error)
{
;dispatch(updateBusyState(false))
;{
;()fetchAuthInfo
;[],{
) return
<NavigationContainer theme={AppTheme}>
) ? busy}
View>
})=style
,'position: 'absolute
,top: 0
,bottom: 0
,left: 0
,right: 0
,backgroundColor: colors.OVERLAY
,'justifyContent: 'center
,'alignItems: 'center
,zIndex: 1
<{
</ Loader>
<View/>
{null : (
{</ loggedIn ? <TabNavigator /> : <AuthNavigator>
<NavigationContainer/>
;(
;{
;export default AppNavigator

```

```

import {createBottomTabNavigator} from '@react-
;navigation/bottom-tabs
;'import HomePage from '../views/auth/HomePage

```

```

;'import ProfilePage from '../views/auth/ProfilePage
import EditProfile from '../views/auth/EditProfile'; // Import
EditProfile
;'import CouponPage from '../views/auth/CouponPage
;'import colors from '../utils/colors
import AntDesign from 'react-native-vector-
;icons/AntDesign
import MaterialComIcon from 'react-native-vector-
;icons/MaterialCommunityIcons
import FontAwesome5 from 'react-native-vector-
;icons/FontAwesome5
;'import React from 'react
import {createNativeStackNavigator} from '@react-
;navigation/native-stack
;'import Octicons from 'react-native-vector-icons/Octicons
;'import SignIn from '../views/auth/SignIn
;'import SignUp from '../views/auth/SignUp
;'import LostPassword from '../views/auth/LostPassword
;'import Verification from '../views/auth/Verification

;()const CouponStack = createStackNavigator
;()const Tab = createBottomTabNavigator

} <= () = const TabNavigator
) return
<>
Tab.Navigator>
} }=screenOptions
,headerShown: false
} :tabBarStyle
,backgroundColor: colors.PRIMARY
,{
<{
Tab.Screen>
"name="Home
component={HomePage}
} }=options
) <= tabBarIcon: ({color, size})
Octicons name="home" size={size} color={color}>
</
,(

," :tabBarLabel
},{
</
Tab.Screen>
"name="Coupon

```

```
component={CouponPage}
}=options
) <= tabBarIcon: ({color, size})
MaterialComIcon>
"name=ticket-percent-outline
size={size}
color={color}
</
,( 
," :tabBarLabel
{{ 
</
Tab.Screen>
"name=Profile
component={ProfilePage}
}=options
) <= tabBarIcon: ({color, size})
FontAwesome5 name="user-circle" size={size} >
</ color={color}
,( 
," :tabBarLabel
{{ 
</
Tab.Screen>
"name>EditProfile
component={EditProfile}
}=options
,tabBarButton: () => null
{{ 
</
Tab.Screen>
"name=SignIn
component={SignIn}
}=options
,tabBarButton: () => null
{{ 
</
Tab.Screen>
"name=SignUp
component={SignUp}
}=options
,tabBarButton: () => null
{{ 
</
Tab.Screen>
"name=LostPassword
```

```

component={LostPassword}
}=options
,tabBarButton: () => null
){}
</
Tab.Screen>
"name="ProfilePage
component={ProfilePage}
}=options
,tabBarButton: () => null
){}
</
Tab.Screen>
"name="HomePage
component={HomePage}
}=options
,tabBarButton: () => null
){}
</
Tab.Screen>
"name="CouponPage
component={CouponPage}
}=options
,tabBarButton: () => null
){}
</
<Tab.Navigator/>
</>
;((
;{
;{
;export default TabNavigator

```

Store

```

import {PayloadAction, createSelector, createSlice} from
;"@reduxjs/toolkit
;'. import {RootState} from

} export interface UserProfile
:id: string
:name: string
@email: string
;verified: boolean
;avatar?: string

```

```

;points: number
;favorites: number
{

} interface AuthState
;notification: any
;profile: UserProfile | null
;loggedIn: boolean
;busy: boolean
{

}= const initialState: AuthState
,profile: null
,loggedIn: false
,busy: false
notification: undefined
;{

})const slice = createSlice
,'name: 'auth
,initialState
} :reducers
updateProfile(authState, {payload}:
) PayloadAction<UserProfile | null>
;authState.profile = payload
,{
} updateLoggedInState(authState, {payload})
;authState.loggedIn = payload
,{
updateBusyState(authState, {payload}:
) PayloadAction<boolean>
;authState.busy = payload
,{
,
;(
;

export const {updateLoggedInState, updateProfile,
= updateBusyState}
;slice.actions

)export const getAuthState = createSelector
,(state: RootState) => state
,authState => authState
;(

;export default slice.reducer

```

```

;import { createStore, combineReducers } from 'redux'
;import { createSelector } from 'reselect'

Define Todo type //
} export interface Todo
:id: number
;text: string
;completed: boolean
{

Reducer for todos //
} <= const todosReducer = (state: Todo[] = [], action: any)
} switch (action.type)
:'case 'ADD_TODO
;return [...state, action.payload]
:default
;return state
{
;

Root reducer combining todosReducer //
})const rootReducer = combineReducers
,todos: todosReducer
;({


Sample initial state //
} = const initialState
] :todos
,{ id: 1, text: 'Learn React', completed: false }
,{ id: 2, text: 'Write Redux code', completed: true }
,{ id: 3, text: 'Build a Redux app', completed: false }
,[]as Todo [
{



Create store with root reducer and initial state //
;const store = createStore(rootReducer, initialState)

Input selectors //
}) => []const selectTodos = (state: { todos: Todo
;state.todos


Main selector with transformation logic //
)const selectCompletedTodos = createSelector
,[selectTodos]
todos => todos.filter(todo => todo.completed)

```

```
;()
```

```
Test the selectors //  
;console.log('All todos:', selectTodos(store.getState())) //  
;console.log('Completed todos:', //  
;selectCompletedTodos(store.getState()))
```

```
Dispatch an action to add a new todo //  
})store.dispatch  
, 'type: 'ADD_TODO  
payload: { id: 4, text: 'Refactor selectors', completed: false  
,}  
;({
```

```
Test the selectors after dispatching the action //  
console.log('All todos after adding a new todo:',  
;selectTodos(store.getState()))  
console.log('Completed todos after adding a new todo:',  
;selectCompletedTodos(store.getState()))
```

```
;"import { configureStore } from "@reduxjs/toolkit"  
;"import authReducer from "./auth

})const store =configureStore  
,reducer: authReducer  
{

<export type RootState = ReturnType<typeof store.getState

;export default store

UI
;import {FC} from 'react'  
;import {Pressable, StyleSheet, Text} from 'react-native'  
;import colors from '../utils/colors'  
;import Loader from './Loader

} interface Props  
;title: string  
;onPress?: void  
;busy?: boolean  
{

} <= const AppButton: FC<Props> = ({title, busy, onPress})  
 ) return
```

```

<Pressable onPress={onPress} style={styles.container}>
  busy ? <Text style={styles.title}>{title}</Text> : !}
  </> <Loader
  <Pressable/>
  ;(
  ;{

})const styles = StyleSheet.create
} :container
,'width: '100%
,height: 45
,backgroundColor: colors.SECONDARY
,'alignItems: 'center
,borderRadius: 25
,{
} :title
,color: colors.CONTRAST
,fontSize: 18
,{(
;({

;export default AppButton

;'import {FC} from 'react
import {TextInputProps, StyleSheet, TextInput, Text} from
;"react-native
;'import colors from '..utils/colors
{} interface Props extends TextInputProps

} <= const AppInput: FC<Props> = props
) return
<TextInput>
{props...}
placeholderTextColor={colors.INACTIVE_CONTRAST}
style={[styles.input , props.style]}
</
;(
;{

})const styles = StyleSheet.create
} :input
,borderWidth: 2
,borderColor: colors.SECONDARY
,height: 45
,borderRadius: 25
,color: colors.CONTRAST

```

```

,padding: 10
,{
;(
;export default AppInput

;'import {FC} from 'react
;'import {StyleSheet, Pressable, Text} from 'react-native
;'import colors from '../utils/colors

} interface Props
:title: string
;onPress?(): void
;active?: boolean
{

const AppLink: FC<Props> = ({title, active = true, onPress})
} <=
) return
Pressable>
onPress={active ? onPress : null}
<style={{opacity: active ? 1 : 0.4}}>
<Text style={styles.title}>{title}</Text>
<Pressable/>
;(
;{

})const styles = StyleSheet.create
} :title
,color: colors.SECONDARY
,{

;(

;export default AppLink

;'import colors from '../utils/colors
;'import {FC} from 'react
;'import {View, StyleSheet, FlexStyle} from 'react-native

} interface Props
:size: number
;position: 'top-left' | 'top-right' | 'bottom-left' | 'bottom-right
{

} <= const CircleUi: FC<Props> = ({size, position})
;{} = let viewPostion: FlexStyle

```

```

} switch (position)
:'case 'top-left
;viewPostion = {top: -size / 2, left: -size / 2}
;break
:'case 'top-right
;viewPostion = {top: -size / 2, right: -size / 2}
;break
:'case 'bottom-right
;viewPostion = {bottom: -size / 2, right: -size / 2}
;break
:'case 'bottom-left
;viewPostion = {bottom: -size / 2, left: -size / 2}
;break
{



) return
View>
})=style
,width: size
,height: size
,'position: 'absolute
,viewPostion...
<{{{
View>
})=style
,width: size
,height: size
,borderRadius: size / 2
,backgroundColor: colors.SECONDARY
,opacity: 0.3
{{{
</



View>
})=style
,width: size / 1.5
,height: size / 1.5
,borderRadius: size / 2
,backgroundColor: colors.SECONDARY
,opacity: 0.3
,'position: 'absolute
,'top: '50%
,'left: '50%
transform: [{translateX: -size / 3}, {translateY: -size /
,3}]

```

```

{{  

</  

<View/>  

;(  

;{  

})const styles = StyleSheet.create  

,{} :container  

;({  

;export default CircleUi  

;'import colors from './utils/colors  

;'import {FC, useEffect} from 'react  

},import Animated  

,useAnimatedStyle  

,useSharedValue  

,withRepeat  

,withTiming  

;'from 'react-native-reanimated {  

import AntDesign from 'react-native-vector-  

;'icons/AntDesign  

} interface Props  

;color?: string  

{  

const Loader: FC<Props> = ({color = colors.CONTRAST})  

} <=  

;(0)const initialRotation = useSharedValue  

} <= ()const transform = useAnimatedStyle  

} return  

,transform: [{rotate: `${initialRotation.value}deg`}]  

;{  

;({  

} <= ()useEffect  

;initialRotation.value = withRepeat(withTiming(360), -1)  

;({  

}) return  

<Animated.View style={transform}>  

</ AntDesign name="loading1" size={24} color={color}>  

<Animated.View/>

```

```
;(
;{

;export default Loader

;

;

;'import colors from './utils/colors
;'import {FC, forwardRef} from 'react
import {View, StyleSheet, TextInput, TextInputProps} from
;"react-native
} interface Props extends TextInputProps
;ref: any
{
  ;(


  (const OTPField = forwardRef<TextInput, Props>((props, ref
} <=
) return
TextInput>
{props...}
ref={ref}
style={[styles.input, props.style]}
placeholderTextColor={colors.INACTIVE_CONTRAST}
</
;(
;({


})const styles = StyleSheet.create
} :input
, width: 50
, height: 50
, borderRadius: 25
, borderColor: colors.SECONDARY
, borderWidth: 2
,'textAlign: 'center
,color: colors.CONTRAST
,fontSize: 18
,'fontWeight: 'bold
,lineHeight: 0
,{

;({


;export default OTPField

;'import {FC} from 'react
```

```

;import {StyleSheet, View} from 'react-native'
;import Icon from 'react-native-vector-icons/Entypo'
;import colors from '../utils/colors

} interface Props
;privateIcon: boolean
{

const PasswordVisibilityIcon: FC<Props> = ({privateIcon})
} <=
) ? return privateIcon
{18}=Icon name="eye" color={colors.SECONDARY} size>
</
) : (
Icon name="eye-with-line" color={colors.SECONDARY} >
</ {18}=size
;(
;{
})const styles = StyleSheet.create
,{} :container
;({

;export default PasswordVisibilityIcon

```

Views

```

;import React, {FC, useEffect, useState} from 'react'
} import
,StyleSheet
,Text
,View
,ScrollView
,Alert
,TouchableOpacity
,Modal
,RefreshControl
;'from 'react-native {
;import client from '../api/client'
;import colors from '../utils/colors'
;import CouponCard from '../components/CouponCard'
;import Header from '../components/Header'
import AddCouponForm from
;"../components/AddCoupon
import AntDesign from 'react-native-vector-
;icons/AntDesign

```

```

;'import {useRoute} from '@react-navigation/native

} interface Coupon
;id: string_
;name: string
;count: number
;couponCode: string
CompanyId?: string; // Make CompanyId optional
;ValidDate: Date
;selectBrand: string
;imageUrl: string
{

} <= () = const CouponPage: FC
;()const route = useRoute
;([])<[]const [coupons, setCoupons] = useState<Coupon>
;const [loading, setLoading] = useState(true)
;("")<const [search, setSearch] = useState<string>
const [showAddCoupon, setShowAddCoupon] =
;useState<boolean>(false)
;const [refreshing, setRefreshing] = useState(false)

const companyId = route.params?.companyId; // Get the
companyId from navigation params

} <= const updateSearch = (search: string)
;setSearch(search)
;{

} <= const handleCouponAdded = (newCoupon: Coupon)
setCoupons(prevCoupons => [newCoupon,
;...prevCoupons])
;{

} <= () const fetchCoupons = async
} try
const {data} = await client.get<{foundCoupons:
}>[]Coupon
,'coupons/GetCoupon/'
;(
;setCoupons(data.foundCoupons)
} (catch (error {
;console.error('Error fetching coupons:', error)
Alert.alert('Error', 'Failed to fetch coupons. Please try
;again later.')
} finally {

```

```

; setLoading(false)
setRefreshing(false); // Make sure to set refreshing to
false after fetch completes
{
;{

} <= ()useEffect
;()fetchCoupons
;[],{

} <= ()const onRefresh = React.useCallback
;setRefreshing(true)
;()fetchCoupons
;[],{



Filter coupons based on companyId if it's provided //
const filteredCoupons = companyId
coupons.filter(coupon => coupon.CompanyId === ?
companyId)
)coupons.filter :
<= coupon
&& coupon.CompanyId

coupon.CompanyId.toLowerCase().includes(search.toLowerCase())
;(

} if (loading)
) return
<View style={styles.loadingContainer}>
<Text>Loading...</Text>
<View/>
;(
{

) return
<View style={styles.container}>
Header search={search} updateSearch={updateSearch}>
</
ScrollView>
contentContainerStyle={styles.scrollViewContent}
}=refreshControl
RefreshControl refreshing={refreshing} >
</ onRefresh={onRefresh}
<{
) <= filteredCoupons.map(coupon)

```

```

</ CouponCard key={coupon._id} coupon={coupon}>
{(
<ScrollView/>
TouchableOpacity>
style={styles.fab}
<onPress={() => setShowAddCoupon(true)}>
< "AntDesign name="plus" size={24} color="white">
<TouchableOpacity/>
Modal visible={showAddCoupon} >
<animationType="slide" transparent={true}>
<View style={styles.modalContainer}>
<View style={styles.modalContent}>
TouchableOpacity>
style={styles.modalCloseButton}
<onPress={() => setShowAddCoupon(false)}>
"AntDesign name="close" size={24} color="black">
</
<TouchableOpacity/>
ScrollView >
<contentContainerStyle={styles.scrollViewContent}>
AddCouponForm >
</ onCouponAdded={handleCouponAdded}>
<ScrollView/>
<View/>
<View/>
<Modal/>
<View/>
;( 
;{ 

})const styles = StyleSheet.create
} :fab
,'position: 'absolute
,margin: 16
,right: 0
,bottom: 0
,backgroundColor: colors.SECONDARY
,borderRadius: 30
,padding: 15
,{ 
} :scrollViewContent
,paddingVertical: 1
,padding: 5
,'alignItems: 'center
,{ 
} :container

```

```
,flex: 1
,backgroundColor: colors.PRIMARY
,{
} :loadingContainer
,flex: 1
,'justifyContent: 'center
,'alignItems: 'center
,{
} :modalContainer
,flex: 1
,'(0.5 ,0 ,0 ,0)backgroundColor: 'rgba
,'justifyContent: 'center
,'alignItems: 'center
,{
} :modalContent
,'backgroundColor: 'white
,borderRadius: 10
,padding: 20
,'width: '100%
,'height: '100%
,{
} :modalCloseButton
,'position: 'absolute
,top: 30
,right: 30
,zIndex: 1
,{
,:({
```

;export default CouponPage

```
;import React, { useState } from 'react'
} import
,View
,Text
,TouchableOpacity
,ImageBackground
,TextInput
,StyleSheet
,Modal
,Platform
,ActivityIndicator
;'from 'react-native {
import Icon from 'react-native-vector-
;'icons/MaterialCommunityIcons
```

```
import FontAwesome from 'react-native-vector-
;icons/FontAwesome
;import ImagePicker from 'react-native-image-crop-picker
;import Ionicons from 'react-native-vector-icons/Ionicons
;import { useAuth } from '../utils/AuthContext
;import client from '../api/client
;import colors from '../utils/colors

} <= const EditProfileScreen: React.FC = ({ navigation })
;()const { user, setUser } = useAuth
const [image, setImage] = useState<string | 
;null>(user?.avatar?.url || null)
const [userName, setUserName] = useState<string>(user 
;? user.name : "")
const [email, setEmail] = useState<string>(user ?
;user.email : "")
const [modalVisible, setModalVisible] = 
;useState<boolean>(false)
;const [loading, setLoading] = useState<boolean>(false)

} <= () = const takePhotoFromCamera
})ImagePicker.openCamera
,compressImageMaxWidth: 300
,compressImageMaxHeight: 300
,cropping: true
,compressImageQuality: 0.7
({
} <= then(image.
;setImage(image.path)
;setModalVisible(false)
({
} <= catch(error.
;console.log('Error taking photo from camera:', error)
;({{
;{

} <= () = const choosePhotoFromLibrary
})ImagePicker.openPicker
,width: 300
,height: 300
,cropping: true
,compressImageQuality: 0.7
({
} <= then(image.
;setImage(image.path)
;setModalVisible(false)
;{
```

```

({  

} <= catch(error.  

;console.log('Error choosing photo from library:', error)  

;({  

;{  

} <= () const handleSave = async  

; setLoading(true)  

} try  

;()const formData = new FormData  

;formData.append('name', userName)  

;formData.append('email', email)  

} if (image)  

},'formData.append('avatar  

,uri: image  

,'type: 'image/jpeg  

,'name: 'avatar.jpg  

;({  

{  

const response = await client.post('/auth/update-  

},profile', formData  

} :headers  

,'Content-Type': 'multipart/form-data'  

,`Authorization: `Bearer ${user?.token}  

,{  

;({  

} if (response.data)  

setUser(response.data.profile); // Assuming setUser  

updates your context/state with the updated profile  

{  

;setLoading(false)  

;console.log('User data saved:', response.data)  

} (catch (error {  

;setLoading(false)  

;console.log('Error updating profile:', error)  

{  

;{  

}) return  

<View style={styles.container}>

```

```

TouchableOpacity style={styles.backButton} >
<onPress={() => navigation.navigate('ProfilePage')}>
  Ionicons name="chevron-back-outline" size={30} >
</ >
<TouchableOpacity/>
<Modal>
  "animationType="slide
  transparent={true}
  visible={modalVisible}
  onRequestClose={() => setModalVisible(false)}
<
  <View style={styles.modalContainer}>
    <View style={styles.innerContainer}>
      <View style={{ alignItems: 'center' }}>
        Text style={styles.modalTitle}>Upload >
        <Photo</Text
        Text style={styles.modalSubtitle}>Choose Your >
        <Profile Picture</Text
        <View/>
        TouchableOpacity style={styles.modalButton} >
          <onPress={takePhotoFromCamera}>
            Text style={styles.modalButtonTitle}>Take >
            <Photo</Text
            <TouchableOpacity/>
            TouchableOpacity style={styles.modalButton} >
              <onPress={choosePhotoFromLibrary}>
                Text style={styles.modalButtonTitle}>Choose >
                <From Library</Text
                <TouchableOpacity/>
                TouchableOpacity style={styles.modalButton} >
                  <onPress={() => setModalVisible(false)}>
                    Text >
                    <style={styles.modalButtonTitle}>Cancel</Text
                    <TouchableOpacity/>
                    <View/>
                    <View/>
                    <Modal/>

<View style={{ margin: 20 }}>
<View style={{ alignItems: 'center' }}>
  TouchableOpacity onPress={() => >
    <setModalVisible(true)>
    <View style={styles.dpContainer}>
      ) ? image}
      ImageBackground style={styles.profileImage} >
        <source={{ uri: image }}>

```

```

<View>
  Icon>
    "name='camera'
    {35}=size
    "color="#fff
  }=style
  ,opacity: 0.7
  ,'alignItems: 'center
  ,'justifyContent: 'center
  ,borderWidth: 1
  ,'borderColor: '#fff
  ,borderRadius: 10
  {{
  </
<View/>
<ImageBackground/>
) : (
<View style={styles.profileImagePlaceholder}>
</ "Icon name="camera" size={35} color="#666>
Text style={{ color: '#666', marginTop: 10 >
<}}>Add Photo</Text
<View/>
{(
<View/>
<TouchableOpacity>
Text style={{ marginTop: 10, fontSize: 18, >
<fontWeight: 'bold' >}{user_name}</Text
<View/>

<View style={styles.action}>
FontAwesome name="user-o" color={'black'} >
</ {20}=size
TextInput>
"placeholder="User Name
"placeholderTextColor="#666666
autoCorrect={false}
style={[styles.textInput, { color: 'black' }]}}
value={user_name}
onChangeText={text => setUserName(text)}
</
<View/>
<View style={styles.action}>
FontAwesome name="envelope-o" color={'black'} >
</ {20}=size
TextInput>
"placeholder="Email

```

```
"placeholderTextColor="#666666
"keyboardType="email-address
autoCorrect={false}
style={[styles.textInput, { color: 'black' }]}
value={email}
onChangeText={text => setEmail(text)}
</
<View/>

TouchableOpacity style={styles.commandButton} >
<onPress={handleSave} disabled={loading}
) ? loading}
</ "ActivityIndicator color="white">
) : (
Text style={styles.modalButtonTitle}>Save >
<Changes</Text
{(
<TouchableOpacity>
<View/>
<View/>
;((
;{

;export default EditProfileScreen

})const styles = StyleSheet.create
} :container
,flex: 1
,{
} :backButton
,'position: 'absolute
,top: 10
,left: 10
,zIndex: 1
,{
} :commandButton
,padding: 15
,borderRadius: 10
,backgroundColor: colors.SECONDARY
,'alignItems: 'center
,marginTop: 20
,{
} :action
,'flexDirection: 'row
,marginTop: 10
,borderBottomWidth: 1
```

```
, 'borderBottomColor: '#f2f2f2
, 'paddingBottom: 5
, {
} :textInput
, 'flex: 1
, 'marginTop: Platform.OS === 'ios' ? 0 : -12
, 'paddingLeft: 10
, 'color: '#05375a
, {
} :modalContainer
, 'flex: 1
, 'justifyContent: 'flex-end
, '(0.5 ,0 ,0 ,0)backgroundColor: 'rgba
, {
} :innerContainer
, 'backgroundColor: '#FFFFFF
, 'paddingTop: 20
, 'borderTopLeftRadius: 20
, 'borderTopRightRadius: 20
, 'paddingHorizontal: 10
, 'paddingBottom: 10
, {
} :modalTitle
, 'fontSize: 27
, 'height: 35
, 'marginBottom: 10
, 'textAlign: 'center
, {
} :modalSubtitle
, 'fontSize: 14
, 'color: 'gray
, 'height: 30
, 'marginBottom: 10
, 'textAlign: 'center
, {
} :modalButton
, 'padding: 13
, 'borderRadius: 10
, 'backgroundColor: colors.SECONDARY
, 'alignItems: 'center
, 'marginVertical: 7
, {
} :modalButtonTitle
, 'fontSize: 17
, 'fontWeight: 'bold
, 'color: 'white
```

```
,{
} :dpContainer
,height: 100
,width: 100
,borderRadius: 15
,'justifyContent: 'center
,'alignItems: 'center
,{
} :profileImage
,height: 100
,width: 100
,borderRadius: 15
,'overflow: 'hidden
,'justifyContent: 'center
,'alignItems: 'center
,{
} :profileImagePlaceholder
,height: 100
,width: 100
,borderRadius: 15
,'backgroundColor: '#E1E2E6
,'justifyContent: 'center
,'alignItems: 'center
,{

};({  
  
;'import React, {useState, useEffect, useRef} from 'react
} import
,View
,StyleSheet
,Image
,ScrollView
,TouchableOpacity
,NativeSyntheticEvent
,NativeScrollEvent
,Alert
,RefreshControl
,Text
;'from 'react-native {
;'import {useNavigation} from '@react-navigation/native
;'import Header from '../components/Header
;'import colors from '../utils/colors
;'import CategoryList from '../components/Category
;'import CouponCard from '../components/CouponCard
;'import client from '../api/client
```

```

} interface Coupon
:id: string_
;name: string
;count: number
;couponCode: string
;CompanyId: string
;ValidDate: Date
;selectBrand: string
;imageUrl: string
{

} <= () = const HomePage: React.FunctionComponent
;("")<const [search, setSearch] = useState<string>
const [currentImageIndex, setCurrentImageIndex] =
;(0)<useState<number>
;const [isPaused, setIsPaused] = useState<boolean>(false)
;([])<[]const [coupons, setCoupons] = useState<Coupon[]>
;const [refreshing, setRefreshing] = useState(false)
;const scrollViewRef = useRef<ScrollView>(null)
;()const navigation = useNavigation

} <= const updateSearch = (search: string)
;setSearch(search)
;{

] = const imagesWithCompanyIds
source: require('../assets/NikeC.jpeg'), companyId: {}
,"Nike"
source: require('../assets/AdidasC.jpeg'), companyId: {}
,"Adidas"
;[

] = const images2WithCompanyIds
source: require('../assets/iHerbC.jpeg'), companyId: {}
,"iHerb"
source: require('../assets/ZaraC.jpeg'), companyId: {}
,"Zara"
;[

] = const images3WithCompanyIds
source: require('../assets/RolexC.jpeg'), companyId: {}
,"Rolex"
source: require('../assets/H&MC.jpeg'), companyId: {}
,"H&M"
;[

```

```

const handleScroll = (event:
} <= NativeSyntheticEvent<NativeScrollEvent>)
;const offsetX = event.nativeEvent.contentOffset.x
;const newIndex = Math.floor(offsetX / 330)
;setCurrentImageIndex(newIndex)
;{

} <= ()useEffect
;let interval: NodeJS.Timeout

} if (!isPaused)
} <= ()interval = setInterval
)setCurrentImageIndex
prevIndex => (prevIndex + 1) %
,imagesWithCompanyIds.length
;(
;(10000 ,{
{

;return () => clearInterval(interval)
;([isPaused] ,{

} <= ()useEffect
} if (scrollViewRef.current)
})scrollViewRef.current.scrollTo
,x: currentImageIndex * 330
,animated: true
;(
{
;([currentImageIndex] ,{

} <= () const fetchCoupons = async
} try
const {data} = await client.get<{foundCoupons:
)<>[]Coupon
,'coupons/GetCoupon/'
;(
;setCoupons(data.foundCoupons)
} (catch (error {
;console.error('Error fetching coupons:', error)
Alert.alert('Error', 'Failed to fetch coupons. Please try
;again later.')
} finally {
;setRefreshing(false)
{
;{

```

```
 } <= ()useEffect
;()fetchCoupons
;[],{  
  
 } <= ()const onRefresh = React.useCallback
;setRefreshing(true)
;()fetchCoupons
;[],{  
  
 } <= const handleImagePress = (companyId: string)
;navigation.navigate('CouponPage', {companyId})
;{  
  
 ) return
ScrollView>
contentContainerStyle={styles.scrollViewContent}
}=refreshControl
RefreshControl refreshing={refreshing} >
</ onRefresh={onRefresh}
<{
<View style={styles.container}>
Header search={search} >
</ updateSearch={updateSearch}>  
  
ScrollView>
ref={scrollViewRef}
horizontal
pagingEnabled
showsHorizontalScrollIndicator={false}
onScroll={handleScroll}
{16}=scrollEventThrottle  
  
contentContainerStyle={styles.horizontalScrollViewContent
<}
</ View style={styles.sidePadding}>
) <= (imagesWithCompanyIds.map((image, index}
TouchableOpacity>
key={index}
onPress={() =>
<handleImagePress(image.companyId)}  
Image>
source={image.source}
style={styles.image}
"resizeMode="contain
</
```

```

<TouchableOpacity>
{(
</ View style={styles.sidePadding}>
<ScrollView/>

</ View style={styles.spacer}>
</ CategoryList>
</ View style={styles.spacer}>

ScrollView>
horizontal
pagingEnabled
showsHorizontalScrollIndicator={false}
onScroll={handleScroll}
{16}=scrollEventThrottle

contentContainerStyle={styles.horizontalScrollViewContent
<}
</ View style={styles.sidePadding}>
) <= (images2WithCompanyIds.map((image, index)
TouchableOpacity>
key={index}
onPress={() =>
<handleImagePress(image.companyId)>
Image>
source={image.source}
style={styles.image}
"resizeMode="contain
</
<TouchableOpacity/>
{(
</ View style={styles.sidePadding}>
<ScrollView/>

</ View style={styles.spacer}>
<View>
<Text style={styles.title}> | Top Coupons</Text>
<View/>
ScrollView>
horizontal
pagingEnabled
showsHorizontalScrollIndicator={false}

contentContainerStyle={styles.horizontalScrollViewContent
<}
</ View style={styles.sidePadding}>

```

```

) <= coupons.slice(0, 5).map(coupon)
View key={coupon._id} >
<style={styles.couponContainer}
</ CouponCard coupon={coupon}>
<View/>
{(
  TouchableOpacity>
  style={styles.seeAllContainer}
  <onPress={() => navigation.navigate('CouponPage')}>
  <Text style={styles.seeAllText}>SEE ALL</Text>
  <TouchableOpacity/>
</ View style={styles.sidePadding}>
<ScrollView/>

</ View style={styles.spacer}>
<ScrollView>
  horizontal
  pagingEnabled
  showsHorizontalScrollIndicator={false}
  onScroll={handleScroll}
  {16}=scrollEventThrottle

  contentContainerStyle={styles.horizontalScrollViewContent
  <}>
</ View style={styles.sidePadding}>
) <= (images3WithCompanyIds.map((image, index)
  TouchableOpacity>
  key={index}
  onPress={() =>
    handleImagePress(image.companyId)}
  Image>
  source={image.source}
  style={styles.image}
  "resizeMode="contain
  </
  <TouchableOpacity/>
{(
  </ View style={styles.sidePadding}>
<ScrollView/>

</ View style={styles.spacer}>
<View>
<Text style={styles.title}> | More Coupons</Text>
<View/>
<ScrollView>
  horizontal

```

```
pagingEnabled
showsHorizontalScrollIndicator={false}

contentContainerStyle={styles.horizontalScrollViewContent
<}
</ View style={styles.sidePadding}>
) <= coupons.slice(10, 15).map(coupon)
View key={coupon._id} >
<style={styles.couponContainer}
</ CouponCard coupon={coupon}>
<View/>
{(
<TouchableOpacity>
style={styles.seeAllContainer}
<onPress={() => navigation.navigate('CouponPage')}>
<Text style={styles.seeAllText}>SEE ALL</Text>
<TouchableOpacity/>
</ View style={styles.sidePadding}>
<ScrollView/>
<View/>
<ScrollView/>
;(
;{

})const styles = StyleSheet.create
} :scrollViewContent
,paddingHorizontal: 5
,{
} :horizontalScrollViewContent
,'alignItems: 'center
,paddingHorizontal: 10
,{
} :container
,flex: 1
,backgroundColor: colors.PRIMARY
,paddingVertical: 1
,'justifyContent: 'space-between
,{
} :image
, width: 300
,height: 150
,marginRight: 15
,'alignItems: 'center
,borderRadius: 20
,{
} :sidePadding
```

```

, width: -15
, {
} :spacer
, height: 35
, {
} :couponContainer
, marginHorizontal: 5
, justifyContent: 'space-between'
, {
} :imageSize
, width: 300
, height: 290
, {
} :title
, fontSize: 24
, 'fontWeight': 'bold'
, paddingHorizontal: 10
, marginBottom: 10
, 'color': 'black'
, {
} :seeAllContainer
, width: 100
, height: 150
, justifyContent: 'center'
, alignItems: 'center'
, borderWidth: 1
, 'borderColor': '#D3D3D3'
, 'borderStyle': 'dashed'
, marginHorizontal: 5
, {
} :seeAllText
, fontSize: 20
, 'color': 'black'
, {
};(
;

;export default HomePage

import AuthInputField from
'../../components/form/AuthInputField'
;import Form from '../../components/form/Form'
;import SubmitBtn from '../../components/form/SubmitBtn'
;import {FC} from 'react'
;import { StyleSheet, View } from 'react-native'
;import * as yup from 'yup'

```

```

;'import AppLink from '../../ui/AppLink
import AuthFormContainer from
;"../../components/AuthFormContainer
import { NavigationProp, useNavigation } from '@react-
;navigation/native
;'import { AuthStackParamList } from '../../@types/navigation
;'import { FormikHelpers } from 'formik
;'import client from '../../api/client

})const lostPasswordSchema = yup.object
email: yup
()string.
trim('Email is missing!').
email('Invalid email').
,required('Email is required').
;({

{} interface Props

} interface InitialValue
;email: string
{

} = const initialValues
," :email
;{

) const handleSubmit = async
,values: initialValue
,<actions: FormikHelpers<initialValue
} <= (
;actions.setSubmitting(true)
} try
we want to send these information to our api //
} ,const {data} = await client.post('/auth/forget-password
,values...
;({

;console.log(data)
} (catch (error {
;console.log('Lost Password error: ', error)
{

;actions.setSubmitting(false)
;{

```

```

} <= const LostPassword: FC<Props> = props
const navigation =
;()<<useNavigation<NavigationProp<AuthStackParamList
) return
Form>
onSubmit={handleSubmit}
initialValues={initialValues}
<validationSchema={lostPasswordSchema}
AuthFormContainer>
"!heading="Forget Password
subHeading="Oops, did you forget your password?
<".Don't worry, we'll help you get back in
<View style={styles.formContainer}>
AuthInputField>
"name="email
"placeholder="john@email.com
"label="Email
"keyboardType="email-address
"autoCapitalize="none
containerStyle={styles.marginBottom}
</
</ "SubmitBtn title="Send link>

<View style={styles.linkContainer}>
AppLink>
"title="Sign in
} <= ()>onPress
;navigation.navigate('SignIn')
{{{
</
AppLink>
"title="Sign up
} <= ()>onPress
;navigation.navigate('SignUp')
{{{
</
<View/>
<View/>
<AuthFormContainer/>
<Form/>
;((
;{

})const styles = StyleSheet.create
} :formContainer

```

```

,'width: '100%
,{  

} :marginBottom  

,marginBottom: 20
,{  

} :linkContainer
,'flexDirection: 'row
,'alignItems: 'center
,'justifyContent: 'space-between
,marginTop: 20
,{  

;({  

;export default LostPassword  

;'import React, {useCallback, useState} from 'react
} import
,Text
,View
,StyleSheet
,Image
,FlatList
,TouchableOpacity
,Alert
,TextInput
,Dimensions
,RefreshControl
;'from 'react-native {
import AsyncStorage from '@react-native-async-
;'storage/async-storage
;'import {useAuth} from '../utils/AuthContext
;'import colors from '../utils/colors
;'import {useFocusEffect} from '@react-navigation/native
;'import Icon from 'react-native-vector-icons/MaterialIcons
;'import Clipboard from '@react-native-clipboard/clipboard
import imageMap from '../utils/imageMap'; // Adjust the
import path as needed
;'import EditBar from '../components/EditBar
;'import LogoutButton from '../components/LogoutButton  

} interface Coupon
:id: string_
;name: string
;count: number
;couponCode: string

```

```

;CompanyId: string
;ValidDate: Date
;selectBrand: string
;imageUrl: string
{

} <= () = const ProfilePage: React.FC
;()const {user} = useAuth
const [savedCoupons, setSavedCoupons] =
;([])<>[]useState<Coupon
const [copiedCouponId, setCopiedCouponId] =
;useState<string | null>(null)
;const [refreshing, setRefreshing] = useState(false)
)const [profileImageUri, setProfileImageUri] = useState
,'https://i.pravatar.cc/100'
;(

} <= () const fetchSavedCoupons = async
} try
const savedCoupons = await
;AsyncStorage.getItem('savedCoupons')
} if (savedCoupons)
;setSavedCoupons(JSON.parse(savedCoupons))
{
} (catch (error {
;console.error('Failed to load saved coupons', error)
{
;{

)useFocusEffect
} <= ()useCallback
;()fetchSavedCoupons
,[] ,{
;(

} <= ()const onRefresh = useCallback
;setRefreshing(true)
;fetchSavedCoupons().then(() => setRefreshing(false))
;[],{


} <= const removeCoupon = async (couponId: string)
} try
)const filteredCoupons = savedCoupons.filter
,coupon => coupon._id !== couponId
;(
)await AsyncStorage.setItem

```

```
'savedCoupons'
,JSON.stringify(filteredCoupons)
;(
;setSavedCoupons(filteredCoupons)
;Alert.alert('Removed', 'Coupon has been removed!')
} (catch (error {
;Alert.alert('Error', 'Failed to remove the coupon.')
{
;{

} <= const copyIt = (code: string, id: string)
;Clipboard.setString(code)
;setCopiedCouponId(id)
)Alert.alert
,'Copied to Clipboard'
,!Coupon code ${code} copied to clipboard'
;(
;{

} <= const renderCoupon = ({item}: {item: Coupon})
;const imageSource = imageMap[item.imageUrl]

) return
<View style={styles.cardWrapper}>
<View style={styles.card}>
<TouchableOpacity>
onPress={() => removeCoupon(item._id)}
<style={styles.saveBar}
Icon name="bookmark" size={28} >
</ color={colors.SECONDARY}
<TouchableOpacity>
<View style={styles.main}>
<View style={styles.colImg}>
Image source={imageSource} style={styles.image}>
</
<View/>
</ View style={styles.vertical}>
<View style={styles.content}>
<Text style={styles.h1}>
{item.count}% <Text
<style={styles.span}>{item.name}</Text
<Text/>
<Text style={styles.p}>
})()Valid till {new Date(item.ValidDate).toDateString
<Text/>
<View/>
```

```
<View/>
<View style={styles.copyButton}>
Text >
<style={styles.input}>{item.couponCode}</Text
TouchableOpacity>
onPress={() => copyIt(item.couponCode, item._id)}
<style={styles.button}>
<Text style={styles.buttonText}>
copiedCouponId === item._id ? 'COPIED' : }
{"COPY"
<Text/>
<TouchableOpacity/>
<View/>
<View/>
<View/>
;( 
;{
) return
<View style={styles.container}>
Image>
style={styles.backgroundImage}
source={require('../..../assets/background.jpg')}
</
{/* Edit Bar */}
</EditBar>
</LogoutButton>
<View style={styles.dpContainer}>
<View style={styles.dpOrangeRound}>
Image style={styles.profileImage} source={{uri: >
</profileImageUri}}
<View/>
<View/>
<View>
) ? user}
<View style={styles.profileTextContainer}>
Text style={[styles.name, {color: >
<'black'>]}>{user.name}</Text
<View/>
) : (
<Text style={styles.notSignedIn}></Text>
{(
<View/>

<View style={styles.messageContainer}>
</View style={styles.messageBar}>
```

```
Text style={styles.savedCouponsMessage}>Your >
<saved coupons</Text
<View/>


```

```
,borderWidth: 5
,borderColor: colors.SECONDARY
,'overflow: 'hidden
,{
} :profileImage
,'height: '100%
,'width: '100%
,'resizeMode: 'cover
,{
} :profileTextContainer
,'alignItems: 'center
,marginTop: 1
,{
} :name
,'alignSelf: 'center
,marginTop: 60
,'fontWeight: 'bold
,fontSize: 30
,'color: 'black
,{
} :notSignedIn
,fontSize: 16
,'color: 'gray
,'textAlign: 'center
,marginTop: 32
,{
} :messageContainer
,'flexDirection: 'row
,'alignItems: 'center
,marginLeft: 20
,marginTop: 10
,{
} :messageBar
, width: 5
, height: 20
,backgroundColor: colors.SECONDARY
,marginRight: 10
,borderRadius: 5
,{
} :savedCouponsMessage
,fontSize: 20
,'fontWeight: 'bold
,color: colors.SECONDARY
,{
} :cardWrapper
,marginBottom: 20
```

```
,{  
} :card  
,width: 345  
,height: 200  
,borderRadius: 5  
,backgroundColor: colors.PRIMARY  
,padding: 10  
,position: 'relative'  
,shadowColor: colors.SECONDARY  
,shadowOffset: {width: 0, height: 4}  
,shadowOpacity: 0.2  
,shadowRadius: 6  
,elevation: 8  
,  
} :saveBar  
,position: 'absolute'  
,top: 5  
,right: 15  
,padding: 5  
,  
} :main  
,flexDirection: 'row'  
,justifyContent: 'space-between'  
,alignItems: 'center'  
,  
} :colmg  
,marginRight: 10  
,  
} :image  
,width: 70  
,height: 60  
,left: 20  
,  
} :vertical  
,borderLeftWidth: 5  
,borderLeftColor: 'black'  
,borderStyle: 'dotted'  
,height: 100  
,  
} :content  
,marginLeft: -20  
,  
} :h1  
,fontSize: 35  
,color: '#565656  
,
```

```
 } :span
,fontSize: 18
,{
} :p
,fontSize: 16
,'color: '#696969
,{
} :copyButton
,'flexDirection: 'row
,'justifyContent: 'space-between
,'alignItems: 'center
,marginTop: 12
,height: 45
,borderRadius: 4
,borderWidth: 1
,borderColor: colors.CONTRAST
,paddingHorizontal: 5
,{
} :input
,flex: 1
,'height: '50%
,fontSize: 15
,color: colors.CONTRAST
,{
} :button
,padding: 5
,backgroundColor: colors.SECONDARY
,borderRadius: 4
,'justifyContent: 'center
,'alignItems: 'center
,{
} :buttonText
,'color: '#fff
,{
} :listContainer
,paddingHorizontal: 10
,{
;({  

  
;export default ProfilePage  

  
;'import React, {FC, useState} from 'react
;'import {StyleSheet, View} from 'react-native
;'import * as yup from 'yup
;'import {FormikHelpers} from 'formik
```

```

import {useNavigation, NavigationProp} from '@react-navigation/native'
;import {useDispatch} from 'react-redux'
import AuthInputField from
;"../../components/form/AuthInputField"
;import Form from "../../components/form/Form"
;import SubmitBtn from "../../components/form/SubmitBtn"
import PasswordVisibilityIcon from
;"../../ui/PasswordVisibilityIcon"
;import AppLink from "../../ui/AppLink"
import AuthFormContainer from
;"../../components/AuthFormContainer"
;import client from "../../api/client"
import {updateLoggedInState, updateProfile} from
;"../../store/auth"
;import {AuthStackParamList} from "../../@types/navigation"
;import {useAuth} from "../../utils/AuthContext"
import AsyncStorage from '@react-native-async-storage/async-storage'

})const signInSchema = yup.object
email: yup
()string.
trim('Email is missing!').
email('Invalid email!').
,required('Email is required!').
password: yup
()string.
trim('Password is missing!').
min(8, 'Password is too short!').
,required('Password is required!').
;({


{} interface Props

} interface SignInUserInfo
;email: string
;password: string
{



} = const initialValues
," :email
," :password
;{


} <= const SignIn: FC<Props> = props

```

```

;const [secureEntry, setSecureEntry] = useState(true)
const navigation =
;()<>useNavigation<NavigationProp<AuthStackParamList
;()const dispatch = useDispatch
;()const {login, user} = useAuth

} <= () = const togglePasswordView
;setSecureEntry(!secureEntry)
;{

) const handleSubmit = async
,values: SignInUserInfo
,<actions: FormikHelpers<SignInUserInfo
} <= (
;actions.setSubmitting(true)
) try
;await login(values)

*/
:id: string
:name: string
:email: string
:verified: boolean
:avatar?: string
:points: number
:favorites: number
/*
," :dispatch(updateProfile({id //
," | name: user.name  //
," | email: user.email  //
;({{verified: user.verified  //
;dispatch(updateLoggedInState(true))

;navigation.navigate('ProfilePage')
} (catch (error {
;console.log('Sign in error:', error)
{
;actions.setSubmitting(false)
;{

) return
Form>
onSubmit={handleSubmit}
initialValues={initialValues}
<validationSchema={signInSchema}
<"!AuthFormContainer heading="Welcome Back>

```

```

<View style={styles.formContainer}>
  AuthInputField>
    "name="email
    "placeholder="john@email.com
    "label="Email
    "keyboardType="email-address
    "autoCapitalize="none
    containerStyle={styles.marginBottom}
  </
  AuthInputField>
    "name="password
    "*****"=placeholder
    "label="Password
    "autoCapitalize="none
    secureTextEntry={secureEntry}
    containerStyle={styles.marginBottom}
    rightIcon={<PasswordVisibilityIcon
    privateIcon={secureEntry} />}
    onRightIconPress={togglePasswordView}
  </
</ "SubmitBtn title="Sign in>

<View style={styles.linkContainer}>
  AppLink>
    "?title="Forgot Your Password
  } <= ()=onPress
  ;navigation.navigate('LostPassword')
  {{{
  </
  AppLink>
    "title="Sign up
  } <= ()=onPress
  ;navigation.navigate('SignUp')
  {{{
  </
  <View/>
  <View/>
  <AuthFormContainer/>
  <Form/>
  ;(
  ;{

})const styles = StyleSheet.create
} :formContainer
,'width: '100%
,{


```

```

} :marginBottom
,marginBottom: 20
,{
} :linkContainer
,'flexDirection: 'row
,'alignItems: 'center
,'justifyContent: 'space-between
,marginTop: 20
,{
} :logo
,width: 190
,height: 70
,{
;({


;export default SignIn


import AuthInputField from
;"../../../components/form/AuthInputField
;'import Form from '../../../components/form/Form
;'import SubmitBtn from '../../../components/form/SubmitBtn
;'import {FC, useState} from 'react
import {SafeAreaView, StyleSheet, View, Image, Text} from
;"react-native
;'import * as yup from 'yup
import PasswordVisibilityIcon from
;"../../../ui/PasswordVisibilityIcon
;'import AppLink from '../../../ui/AppLink
;'import CircleUi from '../../../ui/CircleUi
import AuthFormContainer from
;"../../../components/AuthFormContainer
import {NavigationProp, useNavigation} from '@react-
;navigation/native
;'import {AuthStackParamList} from '../../../@types/navigation
;'import {useHandler} from 'react-native-reanimated
;'import {FormikHelpers} from 'formik
;'import axios from 'axios
;'import client from '../../../api/client


})const signupSchema = yup.object
name: yup
()string.
trim('Name is missing!').
min(3, 'Invalid name!').
,required('Name is required!').

```

```

email: yup
()string.
trim('Email is missing!').
email('Invalid email!').
,required('Email is required!').
password: yup
()string.
trim('Password is missing!').
min(8, 'Password is too short!').
)matches.
(?=.*[a-zA-Z])(?=.*\d)(?=.*[!@#$%^&*])[a-zA-Z/
,$+Z\d!@#$%^&*]
,'!Password is too simple'
(
,required('Password is required!').
;({


{} interface Props

} interface NewUser
;name: string
;email: string
;password: string
{

}= const initialValues
," :name
," :email
," :password
;{



}<= const SignUp: FC<Props> = props
;const [secureEntry, setSecureEntry] = useState(true)

const navigation =
;()<<useNavigation<NavigationProp<AuthStackParamList

}<= () = const togglePasswordView
;setSecureEntry(!secureEntry)
;{



}) const handleSubmit = async
,values: NewUser
,<actions: FormikHelpers<NewUser
} <=
;actions.setSubmitting(true)

```

```
    } try
    } ,const {data} = await client.post('/auth/create
,values...
;{

;navigation.navigate('Verification', {userInfo: data.user})
} (catch (error {
;console.log('Sign up error:', error)
{
;actions.setSubmitting(false)
;{

) return
Form>
onSubmit={handleSubmit}
initialValues={initialValues}
<validationSchema={signupSchema}
AuthFormContainer>
"!heading="Welcome
subHeading="Let's get started by creating your
<" .account
<View style={styles.formContainer}>
AuthInputField>
"name="name
"placeholder="John Doe
"label="Name
containerStyle={styles.marginBottom}
</
AuthInputField>
"name="email
"placeholder="john@email.com
"label="Email
"keyboardType="email-address
"autoCapitalize="none
containerStyle={styles.marginBottom}
</
AuthInputField>
"name="password
"*****"=placeholder
"label="Password
"autoCapitalize="none
secureTextEntry={secureEntry}
containerStyle={styles.marginBottom}
rightIcon={<PasswordVisibilityIcon
privateIcon={secureEntry} />}
onRightIconPress={togglePasswordView}
```

```
</
</ "SubmitBtn title="Sign up>

<View style={styles.linkContainer}>
AppLink>
"?title="Forgot Your Password
} <= ()=onPress
;navigation.navigate('LostPassword')
{{{
</
AppLink>
"title="Sign in
} <= ()=onPress
;navigation.navigate('SignIn')
{{{
</
<View/>
<View/>
<AuthFormContainer/>
<Form/>
;((
;{

})const styles = StyleSheet.create
} :formContainer
,'width: '100%
,{
} :marginBottom
,marginBottom: 20
,{
} :linkContainer
,'flexDirection: 'row
,'alignItems: 'center
,'justifyContent: 'space-between
,marginTop: 20
,{
} :logo
,width: 190
,height: 70
,{
;({(
;export default SignUp

;import {FC, useEffect, useRef, useState} from 'react
```

```

import {Keyboard, StyleSheet, Text, TextInput, View} from
;"react-native
;import AppLink from '../ui/AppLink
import AuthFormContainer from
;"../../../components/AuthFormContainer
;import OTPField from '../ui/OTPField
;import AppButton from '../ui/AppButton
import {NativeStackScreenProps} from '@react-
;navigation/native-stack
;import {AuthStackParamList} from '../@types/navigation
;import client from '../api/client
import {NavigationProp, useNavigation} from '@react-
;navigation/native
;import colors from '../utils/colors

type Props =
NativeStackScreenProps<AuthStackParamList,
;<"Verification

;(")const otpFields = new Array(6).fill

} <= const Verification: FC<Props> = ({route})
const navigation =
;()<<useNavigation<NavigationProp<AuthStackParamList
;const [otp, setOtp] = useState([...otpFields])
;(0)const [activeOtpIndex, setActiveOtpIndex] = useState
;const [submitting, setSubmitting] = useState(false)
;(60)const [coundDown, setCoundDown] = useState
const [canSendNewOtpRequest,
;setCanSendNewOtpRequest] = useState(false)

;const {userInfo} = route.params

;const inputRef = useRef<TextInput>(null)

} <= const handleChange = (value: string, index: number)
;const newOtp = [...otp]

} if (value === 'Backspace')
moves to the previous only if the field is empty //
;if (!newOtp[index]) setActiveOtpIndex(index - 1)
;" = newOtp[index]
} else {
update otp and move to the next //
;setActiveOtpIndex(index + 1)
;newOtp[index] = value

```

```

{
;setOtp([...newOtp])
;{

} <= const handlePaste = (value: string)
} if (value.length === 6)
;()Keyboard.dismiss
;(")const newOtp = value.split
;setOtp([...newOtp])
{
;{

} <= const isValidOtp = otp.every(value
;()return value.trim
;({


} <= () const handleSubmit = async
;if (!isValidOtp) return
;setSubmitting(true)
} try
} ,const {data} = await client.post('/auth/verify-email
,userId: userInfo.id
,(")token: otp.join
;({


navigate back to sign in //
;navigation.navigate('SignIn')
} (catch (error {
;console.log('Error inside Verification: ', error)
{
;setSubmitting(false)
;{



} <= () const requestForOTP = async
;(60)setCountDown
;setCanSendNewOtpRequest(false)
} try
} ,await client.post('/auth/re-verify-email
,userId: userInfo.id
;({


} (catch (error {
;console.log('Requesting for new otp: ', error)
{
;{



} <= ()useEffect

```

```

;()inputRef.current?.focus
;([activeOtpIndex] ,{

} <= ()useEffect
;if (canSendNewOtpRequest) return

} <= ()const intervalId = setInterval
} <= setCountDown(oldCountDown
} if (oldCountDown <= 0)
;setCanSendNewOtpRequest(true)
;clearInterval(intervalId)

;return 0
{
;return oldCountDown - 1
;(
;(1000 ,{

} <= () return
;clearInterval(intervalId)
;{
;([canSendNewOtpRequest] ,{

) return
AuthFormContainer heading="Please look at your >
<".email
<View style={styles.inputContainer}>
} <= (otpFields.map(_ , index)
) return
OTPField>
ref={activeOtpIndex === index ? inputRef : null}
key={index}
"*"=placeholder
} <= ({onKeyPress={({nativeEvent
;handleChange(nativeEvent.key , index)
{
onChangeText={handlePaste}
"keyboardType="numeric
value={otp[index] || ""}
</
;(
{{{
<View/>

AppButton busy={submitting} title="Submit" >
</ onPress={handleSubmit}

```

```
<View style={styles.linkContainer}>
) ? countDown > 0}
Text style={styles.countDown}>{countDown} >
<sec</Text
{null : (
AppLink>
active={canSendNewOtpRequest}
"title="Re-send OTP
onPress={requestForOTP}
</
<View/>
<AuthFormContainer/>
;(
;{
})const styles = StyleSheet.create
}:inputContainer
,'width: '100%
,'flexDirection: 'row
,'justifyContent: 'space-between
,'alignItems: 'center
,marginBottom: 20
,{
}:linkContainer
,marginTop: 20
,'width: '100%
,'justifyContent: 'flex-end
,'flexDirection: 'row
,{
}:countDown
,color: colors.SECONDARY
,marginRight: 7
,{
;({{
;export default Verification
```

צד שרת:

```

;import { RequestHandler } from 'express'
;import Company from '../models/company'

export const createCompany: RequestHandler = async
} <= (req, res)
;const { companyName, ID, image } = req.body

const newCompany = new Company({
;companyName, ID, image })

;()const savedCompany = await newCompany.save

if (savedCompany)
res.json({ message: 'Company Created
Successfully!' })
else
;res.json({ message: 'Could not create Company!' })
;{

export const getCompanies: RequestHandler = async
} <= (req, res)
;()const foundCompanies = await Company.find

if (foundCompanies)
res.status(200).json({ companies: foundCompanies
;})
else
res.status(404).json({ error: 'No companies found in
the DB' })
;{

export const getCompanyById: RequestHandler =
} <= async (req, res)
;const { companyId } = req.params
const foundCompany = await
;Company.findById(companyId)

```

```
if (foundCompany)
;res.status(200).json({ company: foundCompany })
else
res.status(404).json({ error: 'Company not found in
;the DB' })
;{

export const updateCompanyById: RequestHandler =
} <= async (req, res)
;const { companyId } = req.params
} try
const updatedCompany = await
Company.findByIdAndUpdate(companyId, req.body, {
;new: true })
if (updatedCompany)
res.status(200).json({ message: 'Company
;updated successfully!', updatedCompany })
else
res.status(404).json({ error: 'Company not found in
;the DB' })
} (catch (error {
res.status(500).json({ error: 'Error updating
;company in the DB' })
{
;
;{

export const deleteCompanyById: RequestHandler =
} <= async (req, res)
;const { companyId } = req.params

const deletedCompany = await
;Company.findByIdAndDelete(companyId)

if (deletedCompany)
res.json({ message: 'Company deleted
;successfully!' })
else
;res.json({ message: 'Could not delete Company!' })
;{
```

User Controller

```

;"import { RequestHandler } from "express"
;"import jwt from "jsonwebtoken"

import { CreateUser, VerifyEmailRequest } from
;"../../../types/user
;"import User from "../../../models/user"
import { formatProfile, generateToken } from
;"../../../utils/helper
} import
,sendForgotPasswordLink
,sendPassResetSuccessEmail
,sendVerificationMail
;"from "../../../utils/mail {
import EmailVerificationToken from
;"../../../models/emailVerificationToken
import PasswordResetToken from
;"../../../models/passwordResetToken
;"import { isValidObjectId } from "mongoose"
;"import crypto from "crypto"
import { JWT_SECRET, PASSWORD_RESET_LINK } {
;"from "../../../utils/variables"
import { RequestWithFiles } from
;"../../../middleware/fileParser"
;"import cloudinary from "../../../cloud"
;"import formidable from "formidable"
;"import { error } from "console"

export const create: RequestHandler = async (req:
} <= CreateUser, res)
;const { email, password, name } = req.body

;const oldUser = await User.findOne({ email })
if (oldUser)
return res.status(403).json({ error: "Email is already
;in use!" })

```

```
const user = await User.create({ name, email,
;password })

;()const token = generateToken
})await EmailVerificationToken.create
,owner: user._id
,token
;({

sendVerificationMail(token, { name, email, userId:
;}) ()user._id.toString

res.status(201).json({ user: { id: user._id, name, email
;} })
;{

) export const verifyEmail: RequestHandler = async
,req: VerifyEmailRequest
res
} <= (
;const { token, userId } = req.body

const verificationToken = await
})EmailVerificationToken.findOne
,owner: userId
;({

if (!verificationToken)
;return res.status(403).json({ error: "Invalid token!" })

const matched = await
;verificationToken.compareToken(token)
if (!matched) return res.status(403).json({ error:
;"Invalid token!" })

} ,await User.findByIdAndUpdate(userId
,verified: true
;({
```

```
await
EmailVerificationToken.findByIdAndDelete(verificationT
oken._id)

;res.json({ message: "Your email is verified." })
;{

export const sendReVerificationToken:
} <= RequestHandler = async (req, res)
;const { userId } = req.body

if (!isValidObjectId(userId))
return res.status(403).json({ error: "Invalid request!" })
;{}

;const user = await User.findById(userId)
if (!user) return res.status(403).json({ error: "Invalid
:request!" })

if (user.verified)
return res.status(422).json({ error: "Your account is
;already verified" })

})await EmailVerificationToken.findOneAndDelete
,owner: userId
;({

;()const token = generateToken

})await EmailVerificationToken.create
,owner: userId
,token
;({

},sendVerificationMail(token
,name: user?.name
,email: user?.email
,()userId: user?._id.toString
;({
```

```
;res.json({ message: "Please check you mail." })
;{

export const generateForgetPasswordLink:
} <= RequestHandler = async (req, res)
;const { email } = req.body

;const user = await User.findOne({ email })
if (!user) return res.status(404).json({ error: "Account
;not found!" })

generate the link //
https://yourapp.com/reset- //
passwod?token=hfkshf4322hfjkds&userId=67jhfdsahf4
3

})await PasswordResetToken.findOneAndDelete
,owner: user._id
;({

;const token = crypto.randomBytes(36).toString("hex")

})await PasswordResetToken.create
,owner: user._id
,token
;({

const resetLink =
`${PASSWORD_RESET_LINK}?token=${token}&userl
;`d=${user._id}

sendForgetPasswordLink({ email: user.email, link:
;resetLink })

;res.json({ message: "Check you registered mail." })
;{

export const grantValid: RequestHandler = async (req,
} <= res)
;res.json({ valid: true })
```

```
;{

export const updatePassword: RequestHandler =
} <= async (req, res)
;const { password, userId } = req.body

;const user = await User.findById(userId)
if (!user) return res.status(403).json({ error:
;"Unauthorized access!" })

const matched = await
;user.comparePassword(password)
if (matched)
return res
(422)status.
json({ error: "The new password must be .
;different!" })

;user.password = password
;()await user.save

await PasswordResetToken.findOneAndDelete({
;owner: user._id })
send the success email //

sendPassResetSuccessEmail(user.name,
;user.email)
res.json({ message: "Password resets successfully."
;})
;{

export const signIn: RequestHandler = async (req, res)
} <=
;const { password, email } = req.body

})const user = await User.findOne
,email
;(
if (!user) return res.status(403).json({ error:
;"Email/Password mismatch!" })
```

```

compare the password //
const matched = await
;user.comparePassword(password)
if (!matched)
return res.status(403).json({ error: "Email/Password
;mismatch!" })

.generate the token for later use //
const token = jwt.sign({ userId: user._id },
;JWT_SECRET)
;user.tokens.push(token)

;()await user.save

})res.json
} :profile
,id: user._id
,name: user.name
,email: user.email
,verified: user.verified
,avatar: user.avatar?.url
,points: user.points.length
,favorites: user.favorites.length
,{
,token
;(
;{

export const updateProfile: RequestHandler = async
} <= (req, res)
let { name, email } = req.body; // Keep as let to
reassign if needed
;const { avatar } = req.files

Handle different types for name //
} if (typeof name !== "string")
} if (Array.isArray(name))
;(" ")name = name.join

```

```

(else if (typeof name === "object" && name !== null {
}
;name = JSON.stringify(name)
} else {
return res.status(422).json({ error: "Invalid name!
;Name must be a string." })
{
{
Handle different types for email //
} if (typeof email !== "string")
} if (Array.isArray(email))
;(" ")email = email.join
(else if (typeof email === "object" && email !== null {
}
;email = JSON.stringify(email)
} else {
return res.status(422).json({ error: "Invalid email!
;Email must be a string." })
{
{
} try
;const user = await User.findById(req.user.id)
} if (!user)
;console.error('User not found!')
return res.status(404).json({ error: 'User not found!'
;})
{
Validate email format //
} if (!isValidEmail(email))
return res.status(422).json({ error: 'Invalid email
;format!' })
{
Trim the inputs //
;()const newName = name.trim
;()const newEmail = email.trim

```

```

Update user object //
;user.name = newName
;user.email = newEmail

Handle avatar upload if provided //
} if (avatar)
const avatarPath = `uploads/avatars/${user.id}-
;`.jpg(${Date.now}
await avatar.mv(avatarPath); // Assuming you use
`a file upload library that supports `mv
user.avatar = { url: avatarPath, publicId: " " }; //
Update avatar field
{

Save updated user //
;()await user.save

Respond with updated profile //
;res.json({ profile: formatProfile(user) })
} (catch (error {
;console.error('Error updating profile:', error)
res.status(500).json({ error: 'Failed to update
;profile.', details: error })
{
;{

} function isValidEmail(email: string): boolean
Simple email format validation //
;return /^[^@]+@[^@]+\\.[^@]+$/\\.test(email)
{


export const sendProfile: RequestHandler = (req, res)
} <=
;res.json({ profile: req.user })
;{

export const logOut: RequestHandler = async (req, res)
} <=

```

```
;const { fromAll } = req.query

;const token = req.token
;const user = await User.findById(req.user.id)
if (!user) throw new Error("something went wrong,
;user not found!")

logout from all //
;[] = if (fromAll === "yes") user.tokens
else user.tokens = user.tokens.filter((t) => t !==
;token)

;()await user.save
;res.json({ success: true })
;{
```

DB

```
;"import mongoose, { Mongoose } from "mongoose"
;"import { MONGO_URI } from "../utils/variables
```

```
;mongoose.set("strictQuery", true)
mongoose
connect(MONGO_URI).
} <= ()then.
;console.log("db is connected")
({
} <= (catch((err.
;console.log("db connection failed:", err)
;({
```

```
mail
} interface Options
;title: string
;message: string
;link: string
;logo: string
;banner: string
;btnTitle: string
{
```

```
export const generateTemplate = (options: Options)
} <=
const { title, message, link, logo, banner, btnTitle } =
;options

` return
<!DOCTYPE html!>

html>
"lang="en
xmlns:o="urn:schemas-microsoft-
"com:office:office
"xmlns:v="urn:schemas-microsoft-com:vml
<
<head>
<title></title>
meta content="text/html; charset=utf-8" http->
</ "equiv="Content-Type
meta content="width=device-width, initial->
</ "scale=1.0" name="viewport
<style>
} *
;box-sizing: border-box
{

} body
;margin: 0
;padding: 0
{

} a[x-apple-data-detectors]
;color: inherit !important
;text-decoration: inherit !important
{

} MessageViewBody a#
;color: inherit
;text-decoration: none
{
```

```
 } p  
;line-height: inherit  
{  
  
,desktop_hide.  
} desktop_hide table.  
;mso-hide: all  
;display: none  
;max-height: 0px  
;overflow: hidden  
{  
  
} menu_block.desktop_hide .menu-links span.  
;mso-hide: all  
{  
  
} media (max-width: 700px)@  
,desktop_hide table.icons-inner.  
} social_block.desktop_hide .social-table.  
;display: inline-block !important  
{  
  
} icons-inner.  
;text-align: center  
{  
  
} icons-inner td.  
;margin: 0 auto  
{  
  
,fullMobileWidth.  
,image_block img.big.  
} row-content.  
;width: 100% !important  
{  
  
menu-checkbox[type="checkbox"] ~ .menu-  
} links  
;display: none !important
```

```
;padding: 5px 0
{
menu-checkbox[type="checkbox"]:checked ~ .
} .menu-trigger .menu-open
;display: none !important
{
menu-checkbox[type="checkbox"]:checked ~ .
,.menu-links
menu-checkbox[type="checkbox"] ~ .menu-
} trigger
;display: block !important
;max-width: none !important
;max-height: none !important
;font-size: inherit !important
{
menu-checkbox[type="checkbox"] ~ .menu-
,links > a
menu-checkbox[type="checkbox"] ~ .menu-
} links > span.label
;display: block !important
;text-align: center
{
menu-checkbox[type="checkbox"]:checked ~ .
} .menu-trigger .menu-close
;display: block !important
{
} mobile_hide.
;display: none
{
} stack .column.
;width: 100%
;display: block
{
```

```
 } mobile_hide.  
;min-height: 0  
;max-height: 0  
;max-width: 0  
;overflow: hidden  
;font-size: 0px  
{  
  
,desktop_hide.  
} desktop_hide table.  
;display: table !important  
;max-height: none !important  
{  
{  
  
} memu-r7c0m2:checked ~ .menu-links#  
;background-color: #000000 !important  
{  
  
,memu-r7c0m2:checked ~ .menu-links a#  
} memu-r7c0m2:checked ~ .menu-links span#  
;color: #ffffff !important  
{  
  
<style/>  
<head/>  
body>  
"=style  
;background-color: #fff0e3  
;margin: 0  
;padding: 0  
;webkit-text-size-adjust: none-  
;text-size-adjust: none  
"  
  
<  
table>  
"border="0  
"cellpadding="0  
"cellspacing="0  
"class="nl-container  
"role="presentation
```

```
"=style
;mso-table-lspace: Opt
;mso-table-rspace: Opt
;background-color: #ffff0e3
"
"width="100%
<
<tbody>
<tr>
<td>
126
```

```
<tr>
127
```

```
"cellspacing="0
"class="row row-2
"role="presentation
style="mso-table-lspace: Opt; mso-table-
"rspace: Opt
"width="100%
<
<tbody>
<tr>
<td>

```

```
"width="33.33333333333336%
<
div>
"class="spacer_block
"=style
;height: 10px
;line-height: 5px
;font-size: 1px
"
<

<div/>
<td/>
td>
"class="column column-2
"=style
;mso-table-lspace: Opt
;mso-table-rspace: Opt
;font-weight: 400
;text-align: left
;vertical-align: top
;border-top: 0px
;border-right: 0px
;border-bottom: 0px
;border-left: 0px
"
"width="33.33333333333336%
<
table>
"border="0
"cellpadding="0
"cellspacing="0
"class="image_block block-2
"role="presentation
"=style
;mso-table-lspace: Opt
;mso-table-rspace: Opt
"
"width="100%
<
```

```
<tr>

"
"title="Company Logo
"width="147
</
<div/>
<td/>
<tr/>
<table/>
<td/>

```

```
;text-align: left  
;vertical-align: top  
;border-top: 0px  
;border-right: 0px  
;border-bottom: 0px  
;border-left: 0px  
"  
"width="33.33333333333336%"  
<  
div>  
"class="spacer_block  
"=style  
;height: 10px  
;line-height: 5px  
;font-size: 1px  
"  
<  
  
<div/>  
<td/>  
<tr/>  
<tbody/>  
<table/>  
<td/>  
<tr/>  
<tbody/>  
<table/>  
table>  
"align="center  
"border="0  
"cellpadding="0  
"cellspacing="0  
"class="row row-3  
"role="presentation  
style="mso-table-lspace: 0pt; mso-table-  
"rspace: 0pt  
"width="100%"  
<  
<tbody>  
<tr>
```

```
<td>





```

```
;font-size: 1px
"
<

<div/>
<td/>
<tr/>
<tbody/>
<table/>
<td/>
<tr/>
<tbody/>
<table/>

table>
"align="center
"border="0
"cellpadding="0
"cellspacing="0
"class="row row-5
"role="presentation
style="mso-table-lspace: 0pt; mso-table-
"rspace: 0pt
"width="100%
<
<tbody>
<tr>
<td>
table>
"align="center
"border="0
"cellpadding="0
"cellspacing="0
"class="row-content stack
"role="presentation
"=style
;mso-table-lspace: 0pt
;mso-table-rspace: 0pt
;background-color: #ffffff
;color: #000000
```

```
;width: 680px
"
"width="680
<
<tbody>
<tr>
134
```

```
"style="line-height: 10px
<
![Resetting Password](${banner})
"
"title="Resetting Password
"width="374
</
<div/>
<td/>
<tr/>
<table/>





```

```
<


# "=style ;margin: 0 ;color: #101010 ;direction: ltr ;font-family: Arial, Helvetica ,Neue ;Helvetica, sans-serif ;font-size: 27px ;font-weight: normal ;letter-spacing: normal ;line-height: 120% ;text-align: center ;margin-top: 0 ;margin-bottom: 0 " < <strong>${title}</strong> <h1/> <td/> <tr/> <table/> <td/> <tr/> <tbody/> <table/> <td/> <tr/> <tbody/> <table/>


```

```
"width="100%
<
<tbody>
<tr>
<td>
137
```

```
"=style
;height: 10px
;line-height: 5px
;font-size: 1px
"
<

<div/>
<td/>
td>
"class="column column-2
"=style
;mso-table-lspace: Opt
;mso-table-rspace: Opt
;font-weight: 400
;text-align: left
;vertical-align: top
;border-top: 0px
;border-right: 0px
;border-bottom: 0px
;border-left: 0px
"
"width="66.6666666666667%
<
table>
"border="0
"cellpadding="0
"cellspacing="0
"class="text_block block-2
"role="presentation
"=style
;mso-table-lspace: Opt
;mso-table-rspace: Opt
;word-break: break-word
"
"width="100%
<
<tr>
td>
"class="pad
```

```
"=style
;padding-bottom: 10px
;padding-left: 20px
;padding-right: 10px
;padding-top: 15px
"
<
div style="font-family: sans->
<"serif
div>
""=class
"=style
;font-size: 12px
mso-line-height-alt:
;21.6px
;color: #848484
;line-height: 1.8
font-family: Arial,
,Helvetica Neue
;Helvetica, sans-serif
"
<
p>
"=style
;margin: 0
;font-size: 14px
;text-align: center
mso-line-height-alt:
;25.2px
"
<
span style="font-size: >
"14px
{message}</span$<
<
<p/>
<div/>
<div/>
<td/>
<tr/>
```

```
<table/>
table>
"border="0
"cellpadding="0
"cellspacing="0
"class="button_block block-4
"role="presentation
"=style
;mso-table-lspace: Opt
;mso-table-rspace: Opt
"
"width="100%
<
<tr>
td>
"class="pad
"=style
;padding-bottom: 35px
;padding-left: 10px
;padding-right: 10px
;padding-top: 20px
;text-align: center
"
<
div align="center" >
<"class="alignment
[if mso]><v:roundrect --!>
xmlns:v="urn:schemas-microsoft-com:vml"
xmlns:w="urn:schemas-microsoft-com:office:word"
href="www.example.com"
style="height:44px;width:160px;v-text-anchor:middle;"
arcsize="10%" strokeweight="0.75pt"
strokecolor="#101"
fillcolor="#101"><w:anchorlock/><v:textbox
inset="0px,0px,0px,0px"><center style="color:#ffffff;
!><"font-family:Arial, sans-serif; font-size:16px
[endif]--><a
"href="${link}
"=style
;text-decoration: none
```

```
;display: inline-block  
;color: #ffffff  
;background-color: #101  
;border-radius: 4px  
;width: auto  
border-top: 1px solid  
#101  
;font-weight: undefined  
border-right: 1px solid  
#101  
border-bottom: 1px solid  
#101  
border-left: 1px solid  
#101  
;padding-top: 5px  
;padding-bottom: 5px  
font-family: Arial,  
,Helvetica Neue  
;Helvetica, sans-serif  
;font-size: 16px  
;text-align: center  
;mso-border-alt: none  
;word-break: keep-all  
"  
"target=_blank  
span><  
"=style  
;padding-left: 20px  
;padding-right: 20px  
;font-size: 16px  
;display: inline-block  
;letter-spacing: normal  
"  
span><  
"=style  
word-break: break-  
;word  
;line-height: 32px  
"  
{btnTitle}</span$<
```

```
span/><
a/><
<
[if --!>
<!--mso]></center></v:textbox></v:roundrect><![endif]
<div/>
<td/>
<tr/>
<table/>
<td/>
td>
"class="column column-3
"=style
;mso-table-lspace: Opt
;mso-table-rspace: Opt
;font-weight: 400
;text-align: left
;vertical-align: top
;border-top: 0px
;border-right: 0px
;border-bottom: 0px
;border-left: 0px
"
"width="16.66666666666668%
<
div>
"class="spacer_block
"=style
;height: 10px
;line-height: 5px
;font-size: 1px
"
<

<div/>
<td/>
<tr/>
<tbody/>
<table/>
<td/>
```

```
<tr/>
<tbody/>
<table/>

table>
"align="center
"border="0
"cellpadding="0
"cellspacing="0
"class="row row-9
"role="presentation
style="mso-table-lspace: Opt; mso-table-
"rspace: Opt
"width="100%
<
<tbody>
<tr>
<td>
table>
"align="center
"border="0
"cellpadding="0
"cellspacing="0
"class="row-content stack
"role="presentation
"=style
;mso-table-lspace: Opt
;mso-table-rspace: Opt
;color: #000000
;width: 680px
"
"width="680
<
<tbody>
<tr>

```

```
;font-weight: 400
;text-align: left
;vertical-align: top
;border-top: 0px
;border-right: 0px
;border-bottom: 0px
;border-left: 0px
"
"width="16.66666666666668%
<
div>
"class="spacer_block
"=style
;height: 10px
;line-height: 5px
;font-size: 1px
"
<

<div/>
<td/>
td>
"class="column column-2
"=style
;mso-table-lspace: Opt
;mso-table-rspace: Opt
;font-weight: 400
;text-align: left
;vertical-align: top
;border-top: 0px
;border-right: 0px
;border-bottom: 0px
;border-left: 0px
"
"width="66.6666666666667%
<
div>
"class="spacer_block
"=style
;height: 45px
```

```
;line-height: 5px
;font-size: 1px
"
<

<div/>
<td/>
td>
"class="column column-3
"=style
;mso-table-lspace: Opt
;mso-table-rspace: Opt
;font-weight: 400
;text-align: left
;vertical-align: top
;border-top: 0px
;border-right: 0px
;border-bottom: 0px
;border-left: 0px
"
"width="16.66666666666668%
<
div>
"class="spacer_block
"=style
;height: 10px
;line-height: 5px
;font-size: 1px
"
<

<div/>
<td/>
<tr/>
<tbody/>
<table/>
<td/>
<tr/>
<tbody/>
<table/>
```

```
table>
"align="center
"border="0
"cellpadding="0
"cellspacing="0
"class="row row-10
"role="presentation
style="mso-table-lspace: Opt; mso-table-
"rspace: Opt
"width="100%
<
<tbody>
<tr>
<td>
table>
"align="center
"border="0
"cellpadding="0
"cellspacing="0
"class="row-content stack
"role="presentation
"=style
;mso-table-lspace: Opt
;mso-table-rspace: Opt
;color: #000000
;width: 680px
"
"width="680
<
<tbody>
<tr>

```

```
;padding-bottom: 5px
;border-top: 0px
;border-right: 0px
;border-bottom: 0px
;border-left: 0px
"
"width="100%
<
table>
"border="0
"cellpadding="0
"cellspacing="0
"class="icons_block block-1
"role="presentation
"=style
;mso-table-lspace: Opt
;mso-table-rspace: Opt
"
"width="100%
<
<tr>

```

```
"  
"width="100%  
<  
<tr>  
td>  
"class="alignment  
"=style  
;vertical-align: middle  
;text-align: center  
"  
<  
[if vml]><table --!>  
align="left" cellpadding="0" cellspacing="0"  
role="presentation" style="display:inline-block;padding-  
left:0px;padding-right:0px;mso-table-lspace: Opt;mso-  
<--table-rspace: Opt;"><![endif]  
<!--><![if !vml]--!>  
table>  
"cellpadding="0  
"cellspacing="0  
"class="icons-inner  
"role="presentation  
"=style  
;mso-table-lspace: Opt  
;mso-table-rspace: Opt  
;display: inline-block  
;margin-right: -4px  
;padding-left: 0px  
;padding-right: 0px  
"  
<  
<--[endif]!>--!>  
<table/>  
<td/>  
<tr/>  
<table/>  
<td/>  
<tr/>  
<table/>  
<td/>
```

```

<tr/>
<tbody/>
<table/>
<td/>
<tr/>
<tbody/>
<table/>
<td/>
<tr/>
<tbody/>
<table/>
<body/>
<html/>

```

```

`  
;{
```

MiddleWare

```

;"import { RequestHandler } from "express"
import PasswordResetToken from
;"../models/passwordResetToken
;"import { JwtPayload, verify } from "jsonwebtoken"
;"import { JWT_SECRET } from "../utils/variables"
;"import User from "../models/user"

export const isValidPassResetToken: RequestHandler
} <= = async (req, res, next)
;const { token, userId } = req.body

const resetToken = await
;PasswordResetToken.findOne({ owner: userId })
if (!resetToken)
return res
(403)status.
json({ error: "Unauthorized access, invalid token!" .
;})

const matched = await
;resetToken.compareToken(token)
if (!matched)

```

```
return res
(403)status.
json({ error: "Unauthorized access, invalid token!" .
;})

;()next
;{

export const mustAuth: RequestHandler = async (req,
} <= res, next)
;const { authorization } = req.headers
;console.log(req.headers)
;[1]const token = authorization?.split("Bearer ")
;console.log(token)
if (!token) return res.status(403).json({ error:
;"Unauthorized request!" })

const payload = verify(token, JWT_SECRET) as
;JwtPayload
;const id = payload.userId

const user = await User.findOne({ _id: id, tokens:
;token })
if (!user) return res.status(403).json({ error:
;"Unauthorized request!" })

} interface UserRequest
:id: any
;name: string
;email: string
;verified: boolean
;avatar: any
;favorites: number
;points: number
{

} = (req as any).user
,id: user._id
,name: user.name
,email: user.email
```

```
,verified: user.verified
,avatar: user.avatar?.url
,favorites: user.favorites.length
,points: user.points.length
;as UserRequest {

;req.token = token

;()next
;{

;"import { ErrorRequestHandler } from "express

export const errorHandler: ErrorRequestHandler = (err,
} <= req, res, next)

res.status(500).json({error:err.message})
;{
;"import { Request, RequestHandler } from "express
;"import formidable, { Files } from "formidable

} export interface RequestWithFiles extends Request
;files?: Files
{

const fileParser: RequestHandler = (req:
} <= RequestWithFiles, res, next)
if (!req.headers["content-
type"]?.startsWith("multipart/form-data;"))
return res.status(422).json({ error: "Only accepts
;form-data!" })

;const form = formidable({ multiples: false })

} <= (form.parse(req, (err, fields, files
;if (err) return next(err)

;req.body = fields
;req.files = files
```

```
;()next
;(
;{
;{

;export default fileParser

;"import { RequestHandler } from "express
import PasswordResetToken from
;"../../../models/passwordResetToken
;"import User from "../../../models/user

export const isValidPassResetToken: RequestHandler
} <= = async (req, res, next)
} try
;const { token, userId } = req.body

const resetToken = await
;PasswordResetToken.findOne({ owner: userId })
} if (!resetToken)
return res.status(403).json({ error: "Unauthorized
;access, invalid token!" })
{

const matched = await
;resetToken.compareToken(token)
} if (!matched)
return res.status(403).json({ error: "Unauthorized
;access, invalid token!" })
{

;()next
} (catch (error {
console.error("Error in isValidPassResetToken:",
;error)
;res.status(500).json({ error: "Internal server error." })
{
;{

;"import { RequestHandler } from "express
;"import * as yup from "yup
```

```

export const validate = (schema: any): RequestHandler
} <=
} <= return async (req, res, next)
if (!req.body)
return res.status(422).json({ error: "Empty body is
;not expected!" })
})const schemaToValidate = yup.object
,body: schema
;({

} try
)await schemaToValidate.validate
}
,body: req.body
,{
}
,abortEarly: true
{
;(
;()next
} (catch (error {
} if (error instanceof yup.ValidationError)
;res.status(422).json({ error: error.message })
{
{
;{
;{
;
```

Models

```

import mongoose, { Schema, Document } from
;""mongoose

} export interface Coupons extends Document
;name: String
;count: Number
;couponCode: String
;CompanyId: String
;ValidDate: Date
;
```

```

;selectBrand: string
;imageUrl: String
{
}

})const CouponSchema = new Schema
,name: { type: String, required: true }
,count: { type: Number, required: true }
,couponCode: { type: String, required: true, unique:
,true }
,CompanyId: { type: String }
,ValidDate: { type: Date }
,selectBrand: { type: String, required: true }
,imageUrl: { type: String, required: true }
;({
```

export default mongoose.model<Coupons>("Coupons",
;CouponSchema)

```

import { Model, model, ObjectId, Schema } from
;""mongoose
;"import { hash, compare } from "bcrypt

interface (typescript) //
} interface EmailVerificationTokenDocument
;owner: ObjectId
;token: string
;createdAt: Date
{

} interface Methods
;<compareToken(token: string): Promise<boolean
{
```

expire them after 1 hrs //

```

>const emailVerificationTokenSchema = new Schema
,EmailVerificationTokenDocument
,{}
```

```

Methods
})<
} :owner
,type: Schema.Types.ObjectId
,required: true
,"ref: "User
,{
} :token
,type: String
,required: true
,{
} :createdAt
,type: Date
expires: 3600, // 60 min * 60 sec = 3600s
,()default: Date.now
,{
;({}

emailVerificationTokenSchema.pre("save", async
} (function (next
hash the token //
} if (this.isModified("token"))
;this.token = await hash(this.token, 10)
{

;()next
;({

emailVerificationTokenSchema.methods.compareToke
} n = async function (token)
;const result = await compare(token, this.token)
;return result
;{

)export default model
,"EmailVerificationToken"
emailVerificationTokenSchema
as Model<EmailVerificationTokenDocument, {}, (
;<Methods

```

```
import { Model, model, ObjectId, Schema } from
;""mongoose
;"import { hash, compare } from "bcrypt

interface (typescript) //
} interface PasswordResetTokenDocument
;owner: ObjectId
;token: string
;createdAt: Date
{

} interface Methods
;<compareToken(token: string): Promise<boolean
{

expire them after 1 hrs //

>const passwordResetTokenSchema = new Schema
,PasswordResetTokenDocument
,{}
Methods
})<
} :owner
,type: Schema.Types.ObjectId
,required: true
,"ref: "User
,{
} :token
,type: String
,required: true
,{
} :createdAt
,type: Date
expires: 3600, // 60 min * 60 sec = 3600s
,()default: Date.now
,{
;({
```

```
passwordResetTokenSchema.pre("save", async
} (function (next
hash the token //
} if (this.isModified("token"))
;this.token = await hash(this.token, 10)
{

;()next
;({


passwordResetTokenSchema.methods.compareToken
} = async function (token)
;const result = await compare(token, this.token)
;return result
;{



export default model("PasswordResetToken",
>passwordResetTokenSchema) as Model
>PasswordResetTokenDocument
,{}
Methods
;<



;"import { compare, hash } from "bcrypt"
import { Model, model, ObjectId, Schema } from
;""mongoose

} export interface UserDocument
:id: any_
;name: string
;email: string
;password: string
;verified: boolean
;avatar?: { url: string; publicId: string }
;[]tokens: string
;[]favorites: ObjectId
;[]points: ObjectId
{
```

```

} interface Methods
comparePassword(password: string):
;<Promise<boolean
{
const userSchema = new Schema<UserDocument, {}, <Methods
}
, name: { type: String, required: true, trim: true }
email: { type: String, required: true, trim: true, unique: true }
,password: { type: String, required: true }
,verified: { type: Boolean, default: false }
favorites: [{ type: Schema.Types.ObjectId, ref: "Company" }]
,points: [{ type: Schema.Types.ObjectId, ref: "User" }]
tokens: [String]
;({ timestamps: true }, {
} (userSchema.pre("save", async function (next
hash the token //
} if (this.isModified("password"))
;this.password = await hash(this.password, 10)
{
;()next
;({ userSchema.methods.comparePassword = async
} function (password)
const result = await compare(password,
;this.password)
;return result
;{
export default model("User", userSchema) as
;<Model<UserDocument, {}, Methods

```

```
Routers
} import
,create
,generateForgotPasswordLink
,grantValid
,logOut
,sendProfile
,sendReVerificationToken
,signIn
,updatePassword
,updateProfile
,verifyEmail
;"from "../controllers/user {
import { isValidPassResetToken, mustAuth } from
;"../middleware/auth
;"import { validate } from "../middleware/validator
} import
,CreateUserSchema
,SignInValidationSchema
,TokenAndIDValidation
,UpdatePasswordSchema
;"from "../utils/validationSchema {
;"import { Router } from "express
;"import fileParser from "../middleware/fileParser

;()const router = Router

router.post("/create", validate(CreateUserSchema),
;create)
router.post("/verify-email",
;validate(TokenAndIDValidation), verifyEmail)
router.post("/re-verify-email",
;sendReVerificationToken)
router.post("/forget-password",
;generateForgotPasswordLink)
)router.post
,"verify-pass-reset-token/"
,validate(TokenAndIDValidation)
,isValidPassResetToken
```

```

grantValid
;(
)router.post
,"update-password/"
,validate(UpdatePasswordSchema)
,isValidPassResetToken
updatePassword
;(
router.post("/sign-in",
;validate(SignInValidationSchema), signIn)
;router.get("/is-auth", mustAuth, sendProfile)

router.post("/update-profile", mustAuth, fileParser,
;updateProfile)
;router.post("/log-out", mustAuth, logOut)

;export default router

```

```

;'import express from 'express
} import
,createCoupons
,getCouponsCode
,getCouponsCodeById
,updateCouponsCode
	deleteCouponsCode
;'from '../controllers/coupons {

;()const router = express.Router

;router.post('/create', createCoupons)
;router.get('/GetCoupon', getCouponsCode)
router.get('/:CouponsId', getCouponsCodeById)
;router.patch('/:CouponsId', updateCouponsCode)
;router.delete('/:CouponsId', deleteCouponsCode)

;export default router

```

utils

```
; "import { UserDocument } from "../models/user

} <= export const generateToken = (length = 6)
declare a variable //
;"" = let otp

} for (let i = 0; i < length; i++)
;const digit = Math.floor(Math.random() * 10)
;otp += digit
{

;return otp
;{

} <= export const formatProfile = (user: UserDocument)
} return
,id: user._id
,name: user.name
,email: user.email
,verified: user.verified
,avatar: user.avatar?.url
,favorites: user.favorites.length
,points: user.points.length

;{
;{

;"import nodemailer from "nodemailer"
;"import path from "path

;"import User from "../models/user
import EmailVerificationToken from
;""../models/emailVerificationToken
} import
,MAILTRAP_PASS
,MAILTRAP_USER
,SIGN_IN_URL
,VERIFICATION_EMAIL
;"from "../utils/variables {
```

```

;"import { generateToken } from "../utils/helper"
;"import { generateTemplate } from "../mail/template

} <= () = const generateMailTransporter
})const transport = nodemailer.createTransport
,"host: "sandbox.smtp.mailtrap.io
,port: 2525
} :auth
,user: MAILTRAP_USER
,pass: MAILTRAP_PASS
,{
;({

;return transport
;{

} interface Profile
;name: string
;email: string
;userId: string
{

export const sendVerificationMail = async (token:
} <= string, profile: Profile)
;()const transport = generateMailTransporter

;const { name, email, userId } = profile

const welcomeMessage = `Hi ${name}, welcome to
Coupon Club! There are so much thing that we do for
`.verified users. Use the given OTP to verify your email

})transport.sendMail
,to: email
,from: VERIFICATION_EMAIL
,"subject: "Welcome message
})html: generateTemplate
,"title: "Welcome to Coupon Club
,message: welcomeMessage
,"logo: "cid:logo

```

```

,"banner: "cid:welcome
,"#" :link
,btnTitle: token
,({
] :attachments
}
,"filename: "logo.png
,path: path.join(__dirname, "../mail/logo.png")
,"cid: "logo
,{
}
,"filename: "welcome.png
,path: path.join(__dirname, "../mail/welcome.png")
,"cid: "welcome
,{
,[
;(
;{
;

} interface Options
;email: string
;link: string
{

export const sendForgetPasswordLink = async
} <= (options: Options)
;()const transport = generateMailTransporter

;const { email, link } = options

= const message
We just received a request that you forgot your "
password. No problem you can use the link below and
;".create brand new password

})transport.sendMail
,to: email
,from: VERIFICATION_EMAIL
,"subject: "Reset Password Link
})html: generateTemplate

```

```
, "title: "Forget Password
,message
,"logo: "cid:logo
,"banner: "cid:forget_password
,link
,"btnTitle: "Reset Password
,({
] :attachments
}
,"filename: "logo.png
, path: path.join(__dirname, "../mail/logo.png")
,"cid: "logo
,{
}
,"filename: "forget_password.png
path: path.join(__dirname,
,"../mail/forget_password.png")
,"cid: "forget_password
,{
,[
;(
;{

) export const sendPassResetSuccessEmail = async
,name: string
,email: string
} <= (
;()const transport = generateMailTransporter

const message = `Dear ${name} we just updated
your new password. You can now sign in with your new
;.password

})transport.sendMail
,to: email
,from: VERIFICATION_EMAIL
,"subject: "Password Reset Successfully
})html: generateTemplate
,"title: "Password Reset Successfully
,message
```

```

,"logo: "cid:logo
,"banner: "cid:forget_password
,link: SIGN_IN_URL
,"btnTitle: "Log in
,{
] :attachments
}
,"filename: "logo.png
,path: path.join(__dirname, "../mail/logo.png")
,"cid: "logo
,{
}
,"filename: "forget_password.png
,path: path.join(__dirname,
,"../mail/forget_password.png")
,"cid: "forget_password
,{
,[
;(
;{
;
;"import * as yup from "yup
;"import { isValidObjectId } from "mongoose

})export const CreateUserSchema = yup.object().shape
name: yup
()string.
()trim.
required("Name is missing!").
min(3, "Name is too short!").
,max(20, "Name is too long!").
email: yup.string().required("Email is
,missing!").email("Invalid email id!")
password: yup
()string.
()trim.
required("Password is missing!").
min(8, "Password is too short!").
)matches.

```

```
(?=.*[a-zA-Z])(?=.*\d)(?=.*[@#$%^&*])[a-zA-^/
,$+Z\d!@#$%^&*]
"!Password is too simple"
,((
;({
```

```
export const TokenAndIDValidation =
})yup.object().shape
,token: yup.string().trim().required("Invalid token!")
userId: yup
()string.
} (transform(function (value.
} if (this.isType(value) && isValidObjectId(value))
;return value
{
;"" return
({
,required("Invalid userId!").
;({
```

```
export const UpdatePasswordSchema =
})yup.object().shape
,token: yup.string().trim().required("Invalid token!")
userId: yup
()string.
} (transform(function (value.
} if (this.isType(value) && isValidObjectId(value))
;return value
{
;"" return
({
,required("Invalid userId!").
password: yup
()string.
(trim.
required("Password is missing!").
min(8, "Password is too short!").
)matches.
(?=.*[a-zA-Z])(?=.*\d)(?=.*[@#$%^&*])[a-zA-^/
,$+Z\d!@#$%^&*]
```

```
"!Password is too simple"
,((
;({  
  
export const SignInValidationSchema =  
})yup.object().shape  
email: yup.string().required("Email is  
,missing!").email("Invalid email id!")  
password: yup.string().trim().required("Password is  
,missing!")  
;({  
  
const { env } = process as { env: { [key: string]: string } }  
;  
  
} export const  
,MONGO_URI  
,MAILTRAP_USER  
,MAILTRAP_PASS  
,VERIFICATION_EMAIL  
,PASSWORD_RESET_LINK  
,SIGN_IN_URL  
,JWT_SECRET  
,CLOUD_NAME  
,CLOUD_KEY  
,CLOUD_SECRET  
;env = {  
  
;"import express from "express"  
;"import cors from "cors"  
;"import path from "path"  
;"import "dotenv/config"  
;"import "express-async-errors"  
;"import "./db"  
  
;'import companyRoutes from './routers/company'  
;'import couponsCodeRoute from './routers/coupon'  
;"import authRouter from './routers/auth'
```

```

;"import { errorHandler } from "./middleware/errors

;()const app = express

Add CORS middleware //
;app.use(cors())

Middleware to parse JSON request bodies //
;app.use(express.json())
;app.use(express.urlencoded({ extended: false }))

Serve static files (if any) //
app.use('/assets', express.static(path.join(__dirname,
;('../assets'))))

Routes //
;app.use("/auth", authRouter)
;app.use('/companies', companyRoutes)
;app.use('/coupons', couponsCodeRoute)

Error handling middleware //
;app.use(errorHandler)

;const PORT = process.env.PORT || 2717

} <= () ,app.listen(PORT
;console.log("Server is listening on port " + PORT)
;({


utils.ts //
} <= export const generateRandomLogoUrl = (): string
const possibleLogos = ['/assets/logo1.png',
;'/assets/logo2.png', '/assets/logo3.png']
const randomIndex = Math.floor(Math.random() *
;possibleLogos.length)
;return possibleLogos[randomIndex]

;{

```

Env:

MONGO_URI=mongodb://127.0.0.1:27017/couponClu
b
MAILTRAP_USER=fac5c22d21c032
MAILTRAP_PASS=f4f32a344e001e
VERIFICATION_EMAIL=couponClup@gmail.com
PASWORD_RESET_LINK=http://localhost:2717/reset-
password.html
SIGN_IN_URL=http://yourapp.com/sign-in
JWT_SECRET=a63a5e764877cc8a194a9d73ea57908
4ff60202af6cd5252a9809f14432d1dc33483d579
CLOUD_NAME=dm9n8pzds
CLOUD_KEY=356815518915479
CLOUD_SECRET=X8NfLiSmx2Jbxj7AcFex0Jkvbos