

## NEWSPAPER

### סיווג תגיות אוטומטי באמצעות AI

#### מה עושה התכונה?

המערכת מנתחת אוטומטית כל כתבה חדשה ומזהה לה תגיות רלוונטיות באמצעות בינה מלאכותית של OpenAI, לאחר מכן היא שולחת התרעות מותאמות אישית למשתמשים שמעוניינים בנושאים האלה.

#### איך זה עובד?

מציאת כתבות חדשות

```
var articles = _db.GetUntaggedArticles();
```

המערכת מחפשת כתבות שעדיין לא תויגו.

שליחה ל-AI-

```
var tagNames = await _tagService.DetectTagsAsync(article.
```

קטגוריות שהוגדרו: פוליטיקה, טכנולוגיה, ספורט, כלכלה, בריאות, תרבות, מדע, חינוך, צבא, עולם, חוק, פשע, סביבה, בידור.

עיבוד התשובה המערכת מנקה את התוצאות - מסירה רווחים, הופכת לאותיות קטנות, ומשאירה רק תגיות מהרשימה המאושרת.

שמירה במסד נתונים

```
int tagId = _db.GetOrAddTagId(name);
```

```
_db.InsertArticleTagIfNotExists(article.Id, tagId);
```

יוצרת קשר בין הכתבה לתגיות במסד הנתונים בטבלת קשר NEWS\_ARTICLETAGS שליחת התרעות

```
List<User> users = _db.GetUsersInterestedInTags(tagIds)
```

```
.Where(u => u.ReceiveNotifications)
```

```
.ToList();
```

מוצאת משתמשים שעוקבים אחרי התגיות האלה ושולחת להם מייל.

## מערכת התרעות במייל

### מה עושה התכונה?

המערכת שולחת התרעות במייל למשתמשים באופן אוטומטי באמצעות שרת SMTP של Gmail. ההתרעות נשלחות כהודעות HTML מעוצבות וכוללות קישורים לתוכן החדש באתר.

### איך זה עובד?

הגדרת חיבור SMTP

```
using var smtp = new SmtplibClient("smtp.gmail.com")
{
    Port = 587,
    Credentials = new NetworkCredential(senderEmail, senderPassword),
    EnableSsl = true,
    DeliveryMethod = SmtplibDeliveryMethod.Network,
    Timeout = 15000};
```

המערכת מתחברת לשרת Gmail.

יצירת הודעת מייל מעוצבת

```
using var mail = new MailMessage(senderEmail, toEmail, subject,
bodyHtml)
{
    IsBodyHtml = true
};
```

המערכת יוצרת מייל בפורמט HTML עם עיצוב, צבעים וקישורים. שליחה מיידית

```
smtp.Send(mail);
```

איפה זה מופעל? המערכת פועלת בשני שלבים:

שלב 1: המשתמש מפעיל התרעות

```
function toggleNotifications() {
    const isEnabled =
document.getElementById("notificationToggle").checked;
    שליחה לשרת לעדכון העדפות המשתמש.
```

שלב 2: שליחת מיילים אוטומטית, הפורמט והשליחה הפועלים ב-TaggingRunner:

```
string subject = "📰 New article in your interest!";
string body = $"@Hello {user.Name},<br/><br/>
A new article has been published that matches your interest:<br/>
<b>{article.Title}</b><br/><br/>
<a href='https://your-site-url.com'>Click here to read it on our
website</a>";
_mailer.Send(user.Email, subject, body);
```

כאשר מתפרסמת כתבה חדשה עם תגיות שמעניינות משתמש שבחר לקבל התרעות, המערכת שולחת לו מייל אוטומטית.

## מערכת לייקים בזמן אמת עם Firebase

### מה עושה התכונה?

המערכת מאפשרת למשתמשים לתת לייקים לכתבות, עם עדכונים בזמן אמת לכל המשתמשים הרואים את אותה כתבה באמצעות Firebase Realtime Database כל לייק או ביטול לייק מתעדכן מיידית אצל כל המשתמשים ללא צורך לרענן את הדף.

### איך זה עובד?

המשתמש לוחץ על כפתור הלייק ושולח בקשה לשרת עם מזהה המשתמש ומזהה הכתבה. עדכון במסד הנתונים

```
bool liked = _db.ToggleArticleLike(req.UserId, req.ArticleId);  
int newCount = _db.GetLikesCount(req.ArticleId);
```

המערכת מבצעת - Toggle אם המשתמש כבר נתן לייק, היא מבטלת אותו.  
אם לא, היא מוסיפה לייק. לאחר מכן היא סופרת את המספר החדש של הלייקים.  
עדכון בזמן אמת בFirebase

```
await _firebase.UpdateLikeCount(req.ArticleId, newCount);  
:Firebase Realtime Database  
public async Task UpdateLikeCount(int articleId, int count)  
string url = $"{{_baseUrl}}/likes/article_{{articleId}}.json";  
var content = new StringContent(JsonSerializer.Serialize(count),  
Encoding.UTF8, "application/json");  
await _http.PutAsync(url, content);
```

עדכון אוטומטי בדפדפנים כל המשתמשים שרואים את הכתבה באותו רגע.

## עדכון התרעות אינבוקס בזמן אמת

### מה עושה התכונה?

המערכת מעדכנת בזמן אמת את מספר ההתרעות החדשות באינבוקס של כל משתמש באמצעות Firebase Realtime Database, כאשר משתמש מקבל שיתוף ממשמש אחר, מספר ההתרעות מתעדכן מיידית אצלו בממשק ללא צורך לרענן את הדף.

### איך זה עובד?

קבלת שיתוף חדש כאשר משתמש משתף כתבה עם משתמש אחר, המערכת מזהה את האירוע ומתחילה בעדכון ההתרעות. עדכון במסד הנתונים המערכת מעדכנת את מספר ההתרעות במסד הנתונים ומחשבת את המספר החדש של ההתרעות הלא נקראות. עדכון בזמן אמת ב-Firebase-

```
await _firebase.UpdateInboxCount(targetUserId, newCount);  
        המערכת שולחת את המספר החדש ל-Firebase Realtime Database:  
public async Task UpdateInboxCount(int targetUserId, int count)  
{  
    string url = $"{_baseUrl}/userInboxCount/{targetUserId}.json";  
    var content = new StringContent(JsonSerializer.Serialize(count),  
    Encoding.UTF8, "application/json");  
    await _http.PutAsync(url, content);  
}
```

מתבצע עדכון אוטומטי בדפדפנים, כל המשתמשים שמחוברים מקבלים עדכון מידי על מספר ההתרעות החדש באינבוקס. כאשר המשתמש נכנס לקטגוריית INBOX מספר ההתרעות מתאפס.

מה עושה התכונה?

המערכת יוצרת אוטומטית פרסומות מלאות - טקסט פרסומת ותמונה מותאמת - לכל קטגוריה שהמתכנת בוחר בJS ושולח עם קטגוריה לקונטרולר

```
function loadAdBanner()  
fetch("/api/Ads/Generate?category=breaking news")  
    למשל, כאן בחרנו בקטגוריית חדשות חמות.  
    השימוש הוא בשני מודלים מתקדמים של OpenAI הראשון GPT 5 ליצירת טקסט  
    ו DALL-E 3 ליצירת התמונה.
```

איך זה עובד?

יצירת טקסט פרסומת

```
var adText = await ChatAsync(  
    "Write a short, direct, and engaging advertisement about: {category}. "  
    +  
    "Address the reader directly, be persuasive, and avoid emojis. " +  
    "Limit the ad to a maximum of 3 short sentences."  
);
```

המערכת שולחת ל GPT-5 בקשה מדויקת ליצירת טקסט פרסומת קצר ואפקטיבי.

יצירת תמונה מותאמת

```
var imagePrompt = "Create a realistic, modern advertisement image for:  
{category}. " +  
    "No text in the image. Photography-style composition, clean commercial  
lighting, " +  
    "product-focused subject, minimal background. High-resolution  
marketing photo.";
```

```
var imageUrl = await ImageAsync(imagePrompt);
```

המערכת שולחת ל DALL-E 3 בקשה ליצירת תמונה מקצועית בגודל 1792x1024

פורמט באנר רחב.

החזרת תוצאה מלאה

```
return new AdResult { Text = adText, ImageUrl = imageUrl };
```

המשתמש מקבל אובייקט שמכיל גם את הטקסט וגם את קישור התמונה.

## יצירת תמונות אוטומטית לכתבות

### מה עושה התכונה?

המערכת מזהה כתבות שלא נוספה להן תמונה ויוצרת עבורן אוטומטית תמונה מתאימה באמצעות DALL-E 3 של OpenAI התמונות נוצרות בסגנון עיתונאי מקצועי ומותאמות לתוכן הכתבה.

### איך זה עובד?

זיהוי כתבות ללא תמונה המערכת סורקת כתבות שפורסמו ללא תמונה מלווה ומזהה אותן לעיבוד.

בניית פרומפט חכם

```
string prompt = BuildPrompt(title, description);
```

המערכת בונה תיאור מפורט ומותאם:

```
return @$"Create a modern, professional editorial-style image for a news article titled \"{safeTitle}\".
```

```
The article is about: {about}.
```

```
Use abstract news cues (world map overlays, digital grids, subtle glowing headlines).
```

```
Avoid text in the image and avoid explicit or graphic content.
```

```
Style: clean, magazine-worthy composition, balanced colors, elegant lighting, subtle depth.
```

```
Aspect: landscape, suitable for a NEWS website.";
```

יצירת תמונה מקצועית

```
var payload = new
{
    model = "dall-e-3",
    prompt,
    size = "1792x1024", // landscape format
    quality = "standard",
    n = 1
};
```

המערכת קוראת ל DALL-E 3 ומבקשת תמונה ברזולוציה גבוהה בפורמט רחב. הגנה מפני כישלונות

```
return string.IsNullOrEmpty(url) ? DefaultImageUrl : url;
```

אם היצירה נכשלת, המערכת משתמשת בתמונת ברירת מחדל איכותית.

## מערכת דמות אוטאר לכל משתמש

### מה עושה התכונה?

המערכת מממשת מנגנון מתקדם שבו כל משתמש מקבל רמת אוטאר (BRONZE, SILVER, GOLD) בהתאם למספר הלייקים שהוא צבר על התוכן שלו. הרמה מתעדכנת באופן דינמי ומוצגת בפרופיל המשתמש עם אייקון מתאים.

### איך זה עובד?

חישוב רמות אוטומטי

```
db.ExecuteStoredProcudure("NewsSP_UpdateUserAvatarLevels");
```

המערכת מריצה Stored Procedure שסופר את כל הלייקים של כל משתמש ומחשב את רמת האוטאר שלו בהתאם לקריטריונים קבועים.

עדכון פרטי המשתמש

```
AvatarLevel = reader["AvatarLevel"] as string ?? "BRONZE"
```

הרמה נשמרת בטבלת המשתמשים ומוחזרת כחלק מפרטי המשתמש בכל בקשה.

הצגה דינמית בממשק

```
function loadAvatarLevel(level) {
```

```
    avatarLabel.innerText = level;
```

```
    const avatarIcons = {
```

```
        "BRONZE": "../pictures/avatar_bronze.png",
```

```
        "SILVER": "../pictures/avatar_silver.png",
```

```
        "GOLD": "../pictures/avatar_gold.png"
```

```
    };
```

```
    avatarImage.src = avatarIcons[level] || "../pictures/avatar_bronze.png";
```

```
}
```

הממשק מציג את הרמה עם אייקון ותווית מתאימים.