

TRƯỜNG ĐẠI HỌC THỦY LỢI



BÀI TẬP LỚN

PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG

ĐỀ TÀI: ỨNG DỤNG PHẦN MỀM TIN TỨC NEWS WATCH

Giáo viên hướng dẫn: ThS. Kiều Tuấn Dũng

Sinh viên thực hiện:

STT	Mã sinh viên	Họ và tên	Lớp
1	2251061847	Trần Phương Nam	64CNTT.NB
2	2251061714	Trần Tuấn Anh	64CNTT.NB
3	2251061837	Nguyễn Quang Minh	64CNTT.NB

Hà Nội, năm 2025

**BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG
TRƯỜNG ĐẠI HỌC THỦY LỢI**



BÀI TẬP LỚN

**PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG
ĐỀ TÀI: ỨNG DỤNG PHẦN MỀM ĐỌC TIN TỨC TRỰC TUYẾN
NEWS WATCH**

STT	Mã Sinh Viên	Họ và Tên	Ngày Sinh	Điểm	
				Bảng Số	Bảng Chữ
	2251061847	Trần Phương Nam	13/06/2002		
	2251061714	Trần Tuấn Anh	06/02/2004		
	2251061837	Nguyễn Quang Minh	26/07/2004		

CÁN BỘ CHẤM THI

Hà Nội, năm 2025

LỜI NÓI ĐẦU

Trong thời đại công nghệ số phát triển mạnh mẽ, việc cập nhật tin tức nhanh chóng và chính xác đã trở thành nhu cầu thiết yếu của mọi người. Nhờ có chiếc điện thoại thông minh, chúng ta có thể dễ dàng truy cập và đọc tin tức ở bất cứ đâu, bất cứ lúc nào chỉ với vài thao tác chạm.

Xuất phát từ mong muốn mang đến một trải nghiệm đọc tin hiện đại và tiện lợi, nhóm chúng em đã phát triển ứng dụng phần mềm đọc tin tức trực tuyến **NEWS WATCH** này. Với thiết kế giao diện trực quan, dễ sử dụng, cùng những tính năng thông minh như cập nhật tin tức theo thời gian thực, phân loại chủ đề đa dạng và cá nhân hóa nội dung theo sở thích, chúng em hy vọng ứng dụng sẽ trở thành người bạn đồng hành tin cậy, giúp người dùng tiếp cận thông tin một cách nhanh chóng và hiệu quả nhất.

Chúng em chân thành cảm ơn sự giúp đỡ của giảng viên ThS. Kiều Tuấn Dũng đã tận tình hướng dẫn trong suốt quá trình học tập, cung cấp những kiến thức quý báu và các tài liệu cần thiết để chúng em hoàn thành bài tập lớn này.

Trong quá trình tìm hiểu và thực hiện, do kiến thức còn hạn chế và thời gian nghiên cứu còn ít ỏi, nhóm chúng em không tránh khỏi những thiếu sót. Rất mong nhận được sự góp ý và hướng dẫn từ thầy để bài tập lớn của chúng em được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

MỤC LỤC

LỜI NÓI ĐẦU	1
MỤC LỤC	2
DANH MỤC HÌNH ẢNH	3
DANH MỤC BẢNG BIỂU	5
DANH MỤC CÁC TỪ VIẾT TẮT	6
Chương 1. TỔNG QUAN VỀ ĐỀ TÀI	7
1.1. Giới thiệu về đề tài	7
1.2. Mục tiêu của đề tài	7
1.3. Phạm vi của đề tài	7
1.4. Phân chia nhiệm vụ	8
Chương 2. KIẾN TRÚC VÀ CÔNG NGHỆ	11
2.1. Kiến trúc hệ thống.....	11
2.2. Giới thiệu về Công nghệ phát triển.....	11
Chương 3. XÂY DỰNG ỨNG DỤNG	11
3.1. Thiết kế Figma	12
3.1.1. Link Github	12
3.1.2. Ảnh kết xuất Figma các màn hình.....	12
3.2. Thiết kế cơ sở dữ liệu – UseCase.....	28
3.3. Giao diện ứng dụng.....	30
3.4. Code minh họa các chức năng cốt lõi	37
KẾT LUẬN	49
1. Kết quả đạt được.....	49
2. Nhược điểm	49
3. Hướng phát triển.....	49

TÀI LIỆU THAM KHẢO.....	50
--------------------------------	-----------

DANH MỤC HÌNH ẢNH

Hình 1 Ảnh kết xuất Figma App - Splash Screen	13
Hình 2 Ảnh kết xuất Figma App - Register.....	14

Hình 3 Ảnh kết xuất Figma App - Login	14
Hình 4 Ảnh kết xuất Figma App - Check OTP	15
Hình 5 Ảnh kết xuất Figma App - Forgot Password	15
Hình 6 Ảnh kết xuất Figma App - Set New Password	16
Hình 7 Ảnh kết xuất Figma App - Home	16
Hình 8 Ảnh kết xuất Figma App - Categories	17
Hình 9 Ảnh kết xuất Figma App - New for Category	17
Hình 10 Ảnh kết xuất Figma App - All News	18
Hình 11 Ảnh kết xuất Figma App - News Detail	18
Hình 12 Ảnh kết xuất Figma App - Search News	19
Hình 13 Ảnh kết xuất Figma App - Bookmark	19
Hình 14 Ảnh kết xuất Figma App - Comment	20
Hình 15 Ảnh kết xuất Figma App - Setting.....	20
Hình 16 Ảnh kết xuất Figma App - Change Infomation	21
Hình 17 Ảnh kết xuất Figma App - Help Center.....	21
Hình 18 Ảnh kết xuất Figma Admin - Login	22
Hình 19 Ảnh kết xuất Figma Admin - Dashboard	22
Hình 20 Ảnh kết xuất Figma Admin - Article Management.....	22
Hình 21 Ảnh kết xuất Figma Admin - Add Article.....	23
Hình 22 Ảnh kết xuất Figma Admin - Edit Article	23
Hình 23Ảnh kết xuất Figma Admin - Detail Article	24
Hình 24 Ảnh kết xuất Figma Admin - Account Management	24
Hình 25 Ảnh kết xuất Figma Admin - Add Employee.....	25
Hình 26 Ảnh kết xuất Figma Admin - Edit Employee	25
Hình 27 Ảnh kết xuất Figma Admin - Category Management	26
Hình 28 Ảnh kết xuất Figma Admin – Add Category.....	26
Hình 29 Ảnh kết xuất Figma Admin - Edit Category.....	27
Hình 30 Ảnh kết xuất Figma Admin - Change Infomation.....	27
Hình 31 Ảnh thiết kế CSDL	28
Hình 32 Giao diện ứng dụng đăng ký tài khoản.....	30
Hình 33 Giao diện ứng dụng dẫn nhập.....	31
Hình 34 Giao diện ứng dụng chức năng quên mật khẩu	31
Hình 35 Giao diện trang chính của ứng dụng.....	32
Hình 36 Giao diện ứng dụng tất cả tin tức	32
Hình 37 Giao diện ứng dụng tìm kiếm.....	33

Hình 38 Giao diện ứng dụng chi tiết tin tức	33
Hình 39 Giao diện ứng dụng danh mục tin tức	34
Hình 40 Giao diện ứng dụng danh sách tin tức được lưu.....	35
Hình 41 Giao diện ứng dụng bình luận	35
Hình 42 Giao diện ứng dụng cài đặt.....	35
Hình 43 Giao diện ứng dụng thông tin tài khoản	36
Hình 44 Giao diện ứng dụng trung tâm hỗ trợ	36

DANH MỤC BẢNG BIỂU

Bảng 1: Bảng Phân công nhiệm vụ	8
Bảng 2: Bảng User.....	28
Bảng 3: Bảng Article (Bài viết/Tin tức)	29
Bảng 4: Bảng Category (Danh mục bài viết)	29
Bảng 5: Bảng Comment (Bình luận)	29
Bảng 6: Bảng Bookmark (Lưu bài viết yêu thích)	29

DANH MỤC CÁC TỪ VIẾT TẮT

STT	TỪ VIẾT TẮT	VIẾT ĐẦY ĐỦ
1	OTP	One Time Password
2	API	Application Programming Interface
3	MVC	Model – View – Controller
4	UI	User Interface
5	HTTP	Hypertext Transfer Protocol

Chương 1. TỔNG QUAN VỀ ĐỀ TÀI

1.1. Giới thiệu về đề tài

Trong thời đại công nghệ số phát triển mạnh mẽ, nhu cầu cập nhật tin tức nhanh chóng và chính xác ngày càng trở nên quan trọng. Việc sử dụng điện thoại thông minh để truy cập và đọc tin tức đã trở thành một thói quen phổ biến, giúp người dùng nắm bắt thông tin mọi lúc, mọi nơi.

Nhằm đáp ứng nhu cầu này, nhóm chúng em đã phát triển một ứng dụng đọc tin tức trên thiết bị di động, giúp người dùng dễ dàng tiếp cận tin tức từ nhiều nguồn khác nhau, với giao diện thân thiện, tính năng thông minh và khả năng cá nhân hóa nội dung theo sở thích.

1.2. Mục tiêu của đề tài

- Xây dựng một ứng dụng đọc tin tức trên nền tảng Android, hỗ trợ cập nhật tin tức nhanh chóng, chính xác.
- Thiết kế giao diện thân thiện, dễ sử dụng giúp người dùng có trải nghiệm mượt mà.
- Cung cấp các tính năng đọc, cập nhật tin tức, phân loại tin tức theo chủ đề, giúp người dùng dễ dàng tìm kiếm và theo dõi những thông tin quan trọng.
- Hỗ trợ cá nhân hóa nội dung dựa trên sở thích của người dùng.
- Cho phép người dùng cập nhật thông tin cá nhân trực tiếp trên ứng dụng, đảm bảo dữ liệu luôn chính xác và cập nhật.

1.3. Phạm vi của đề tài

- Nền tảng phát triển: Ứng dụng được xây dựng trên hệ điều hành Android, sử dụng ngôn ngữ lập trình Java.
- Đối tượng sử dụng:
 - Người dùng: là những người có thể đọc hiểu tiếng anh và có nhu cầu thường xuyên cập nhật tin tức.
 - Người quản trị: là những người cập nhật dữ liệu mới cho ứng dụng
- Tính năng chính:
 - Người dùng
 - Đăng nhập, Đăng ký, Đổi mật khẩu, Đăng Xuất
 - Hiện thị danh sách tin tức, danh mục tin tức

- Tìm kiếm, lọc, chia sẻ, chi tiết tin tức
- Đọc bình luận, bình luận về bài viết
- Lưu tin tức vào danh sách riêng
- Thay đổi thông tin cá nhân
- Người quản lý
 - Đăng nhập
 - Quản lý tin tức
 - Quản lý danh mục
 - Quản lý tài khoản
 - Thay đổi thông tin tài khoản
- Công nghệ sử dụng
 - Frontend: Android studio (Java)
 - Backend: Spring Boot (API và Admin)
 - Cơ sở dữ liệu: MySQL
 - Giao tiếp: Retrofit (trong Android Studio để gọi API)
 - Bảo mật: Spring Security (Basic Authentication) để xác thực

1.4. Phân chia nhiệm vụ

Họ và tên	Nhiệm vụ
Trần Phương Nam	Thiết kế Figma Thiết kế database Giao diện Admin Giao diện User: <ul style="list-style-type: none"> - Giao diện trang Home - Giao diện trang News - Giao diện trang Category - Giao diện trang Bookmark - Giao diện trang Settings

	<p>Chức năng Admin</p> <ul style="list-style-type: none"> - Đăng nhập, Đăng xuất - Quản lý tài khoản - Quản lý tin tức - Quản lý danh mục - Thay đổi thông tin
Trần Tuấn Anh	<p>Thiết kế Figma</p> <p>Thiết kế database</p> <p>Chức năng User</p> <ul style="list-style-type: none"> - Đăng nhập, Đăng kí, Đăng xuất - Quên mật khẩu - Hiển thị danh sách tin tức - Hiển thị chi tiết tin tức - Hiển thị danh sách danh mục - Phân loại tin tức theo danh mục - Hiển thị danh sách tin tức đã lưu - Hiển thị danh sách bình luận - Bình luận vào bài viết - Tìm kiếm tin tức - Lưu, xóa tin tức trong danh sách riêng - Chia sẻ tin tức - Cập nhật thông tin người dùng
Nguyễn Quang Minh	<p>Thiết kế Figma</p> <p>Thiết kế database</p> <p>Giao diện User:</p> <ul style="list-style-type: none"> - Giao diện trang Login - Giao diện trang Register - Giao diện trang Change Information - Giao diện trang Comment - Giao diện trang Forgot Password

	<ul style="list-style-type: none"> - Giao diện trang Set New Password - Giao diện trang Check OTP - Giao diện trang News Category - Giao diện trang News Detail - Giao diện trang Help
--	---

Chương 2. KIẾN TRÚC VÀ CÔNG NGHỆ

2.1. Kiến trúc hệ thống

Ứng dụng **NewsWatch** được phát triển theo mô hình **Client-Server** với hai thành phần chính: **ứng dụng người dùng (Client)** và **máy chủ (Server)**. Dữ liệu được lưu trữ và quản lý bởi hệ quản trị cơ sở dữ liệu **MySQL**.

- Client:
 - Nền tảng: Ứng dụng được phát triển trên hệ điều hành Android, sử dụng ngôn ngữ lập trình Java
 - Chức năng: Đây là giao diện chính mà người dùng tương tác, với các chức năng do ứng dụng cung cấp như đăng nhập, đăng kí, đọc tin tức, tìm kiếm,...
 - Giao diện người dùng: Được thiết kế thân thiện với người dùng, dễ dàng sử dụng
 - Kết nối: Ứng dụng giao tiếp với máy chủ thông qua RESTful API, sử dụng các phương thức HTTP (GET, POST, PUT, DELETE)
- Server:
 - Nền tảng: Được xây dựng bằng Spring Boot, theo mô hình, sử dụng ngôn ngữ lập trình Java
 - Vai trò:
 - Xử lý logic nghiệp vụ: xác thực người dùng, xử lý truy vấn dữ liệu, điều phối các chức năng.
 - Cung cấp API cho ứng dụng client sử dụng.
 - Giao diện quản trị (Admin): Được phát triển riêng, cho phép quản trị viên đăng nhập, quản lý nội dung tin tức, kiểm soát người dùng và giám sát hoạt động hệ thống.

Hệ thống được thiết kế theo mô hình MVC nhằm tách biệt giao diện, xử lý logic và dữ liệu, giúp mã nguồn dễ bảo trì và mở rộng.

- **Model (M):** Nó chứa tất cả các hàm, các phương thức truy vấn trực tiếp với dữ liệu, Controller sẽ thông qua các hàm, phương thức đó để lấy dữ liệu rồi gửi qua View
- **View (V):** Là giao diện người dùng (UI), chứa các thành phần tương tác với người dùng, là nơi nhận dữ liệu từ Controller và hiển thị
- **Controller (C):** Là thành phần trung gian giữa Model và View, đảm nhận vai trò tiếp nhận yêu cầu từ người dùng, thông qua Model để lấy dữ liệu sau đó thông qua View để hiển thị cho người dùng

2.2. Giới thiệu về Công nghệ phát triển

- Android studio

- Ngôn ngữ: Java
- Mô hình kiến trúc: MVC
- Thư viện hỗ trợ
 - Retrofit: Gọi API để truy vấn tới cơ sở dữ liệu
 - Glide: Tải ảnh từ internet
 - Okhttp: Gửi và nhận dữ liệu từ server thông qua internet
- Spring Boot:
 - Ngôn ngữ: Java
 - Mô hình kiến trúc: MVC
 - Các module chính:
 - Spring Web: Xây dựng RESTful API
 - Spring Data JPA Kết nối, thao tác với cơ sở dữ liệu
 - Spring Security: Bảo mật (nếu có yêu cầu đăng nhập)
 - Thymeleaf: Xây dựng giao diện web, template
- MySQL: Lưu trữ toàn bộ dữ liệu
- Các công cụ khác:
 - Postman: Kiểm thử API
 - Github: Quản lý mã nguồn

Chương 3. XÂY DỰNG ỨNG DỤNG

3.1. Thiết kế Figma

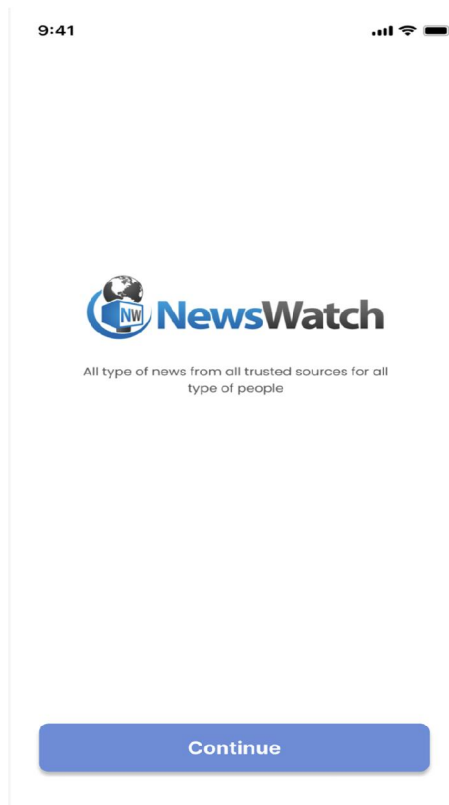
3.1.1. Link Github

Link Github: <https://github.com/Tanhhh1/Project-Android.git>

3.1.2. Ảnh kết xuất Figma các màn hình

❖ Mobile App

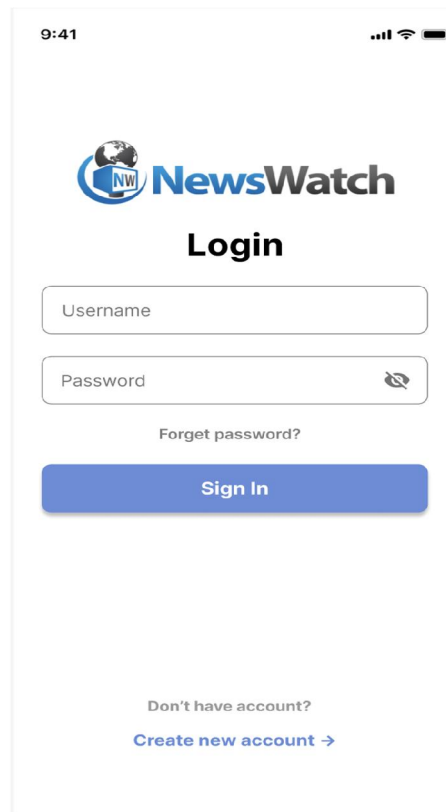
1) Splash Screen




2) Register

A mobile app sign-up screen for NewsWatch. At the top, the status bar shows the time 9:41, signal strength, Wi-Fi, and battery. The NewsWatch logo is at the top center. Below the logo, the text 'Sign Up' is centered. There are four input fields: 'Username', 'Email', 'Password', and 'Confirm Password'. The 'Password' and 'Confirm Password' fields have toggle icons on the right. Below the input fields is a blue rounded rectangular button with the text 'Sign Up'. At the bottom, the text 'Already have account?' is centered, followed by a blue link 'Sign In Here →'.

3) Login




9:41

 **NewsWatch**

Login

Username

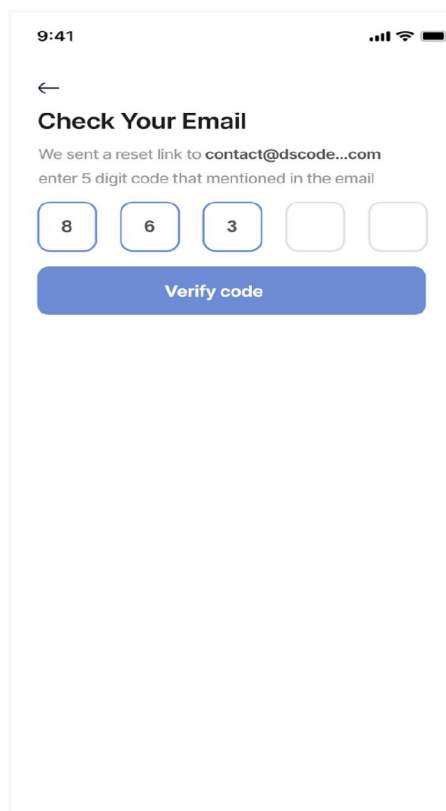
Password 

[Forget password?](#)

Sign In

Don't have account?
[Create new account →](#)

4) Check OTP



9:41

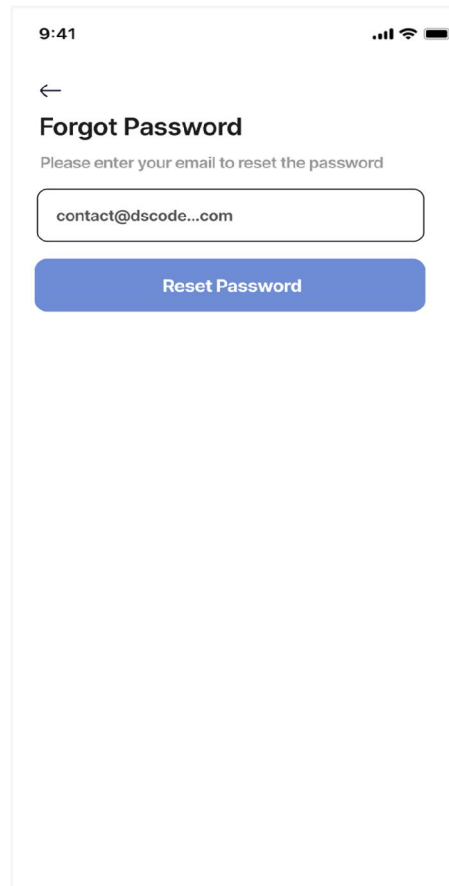
←

Check Your Email

We sent a reset link to [contact@dscodex.com](#)
enter 5 digit code that mentioned in the email

Verify code

5) Forgot Password



9:41

←

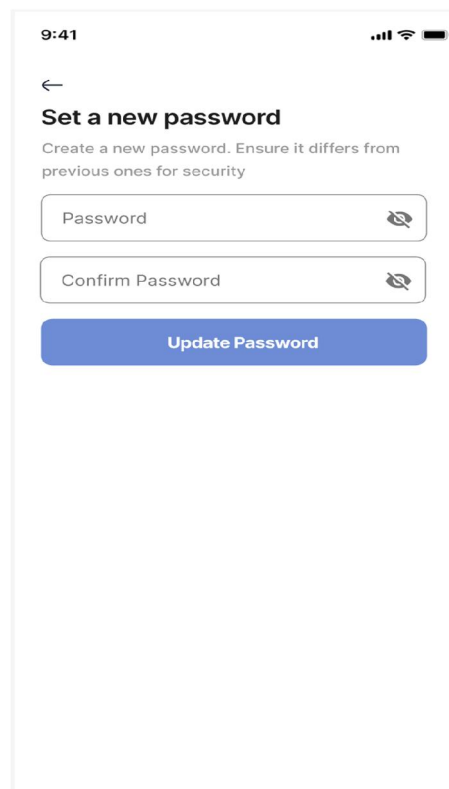
Forgot Password

Please enter your email to reset the password

Reset Password

This is a mobile app screen for the 'Forgot Password' feature. It has a white background and a light gray border. At the top, the status bar shows the time '9:41' and signal/battery icons. Below the status bar is a back arrow icon. The title 'Forgot Password' is in bold black text. Below the title is a subtitle 'Please enter your email to reset the password'. There is a text input field containing the email 'contact@dscode...com'. Below the input field is a blue button with the text 'Reset Password'.

6) Set New Password



9:41

←

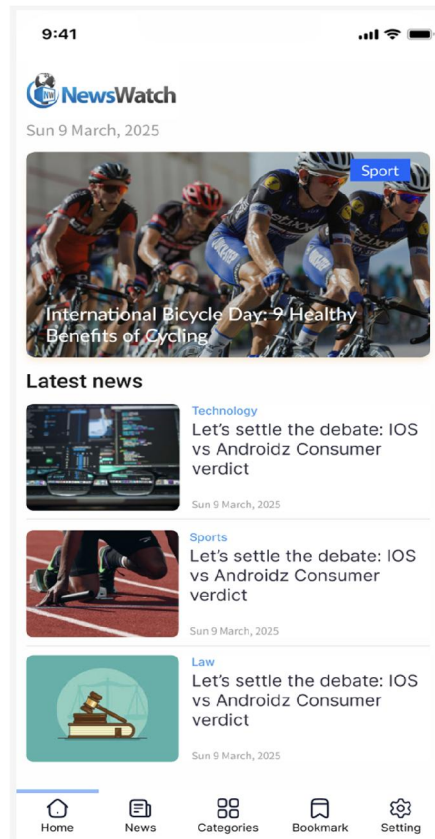
Set a new password

Create a new password. Ensure it differs from previous ones for security

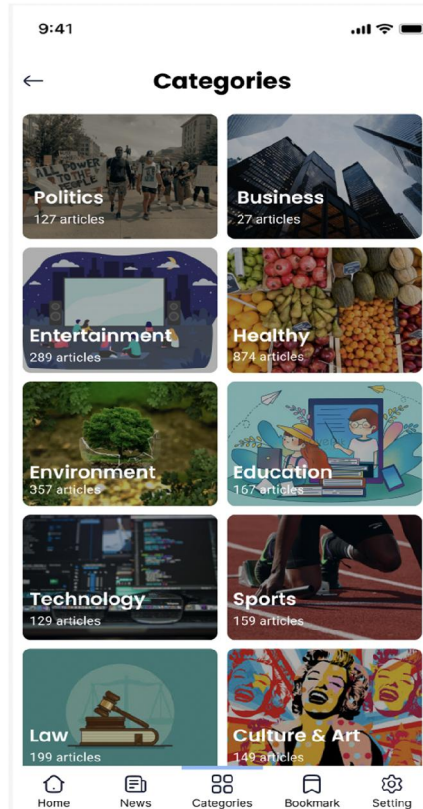
Update Password

This is a mobile app screen for the 'Set a new password' feature. It has a white background and a light gray border. At the top, the status bar shows the time '9:41' and signal/battery icons. Below the status bar is a back arrow icon. The title 'Set a new password' is in bold black text. Below the title is a subtitle 'Create a new password. Ensure it differs from previous ones for security'. There are two text input fields: the first is labeled 'Password' and the second is labeled 'Confirm Password'. Both input fields have a small eye icon to the right of the text. Below the input fields is a blue button with the text 'Update Password'.

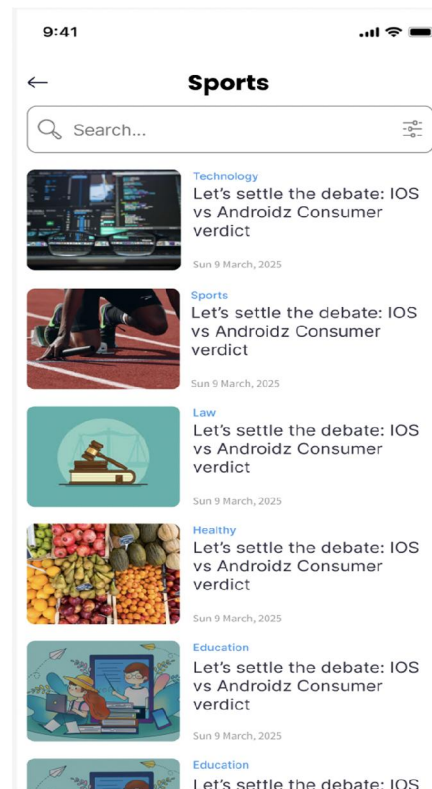
7) Home



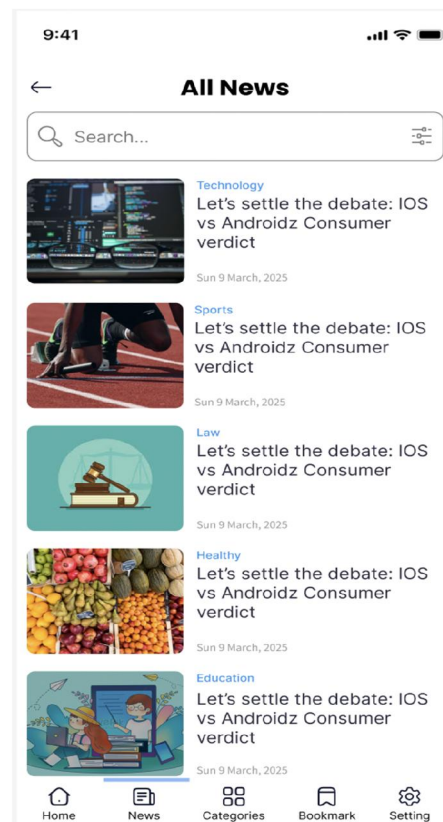
8) Categories



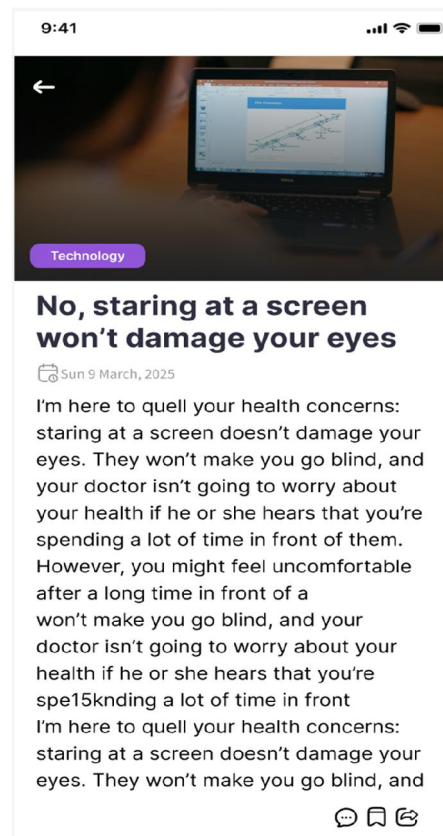
9) New for Category



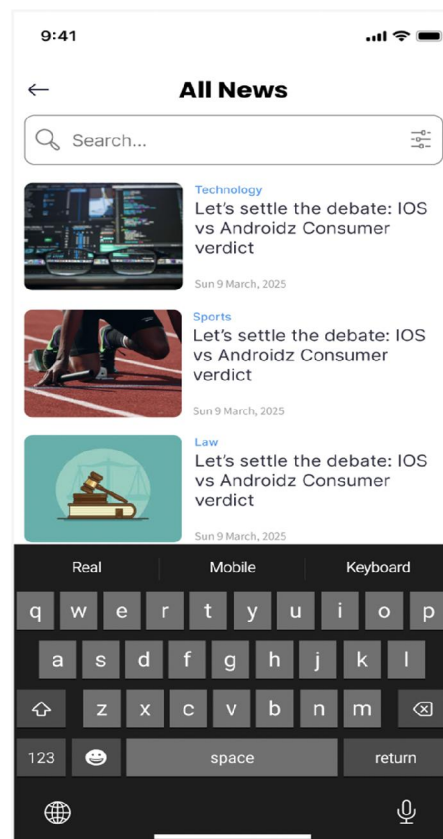
10) All News



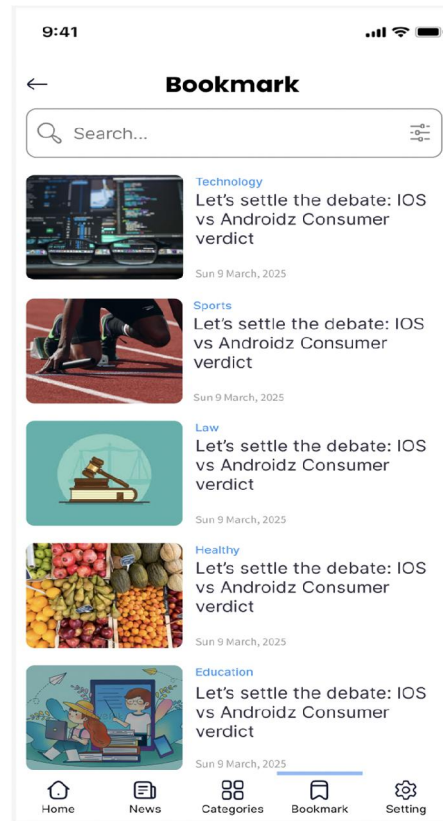
11) News Detail



12) Search News



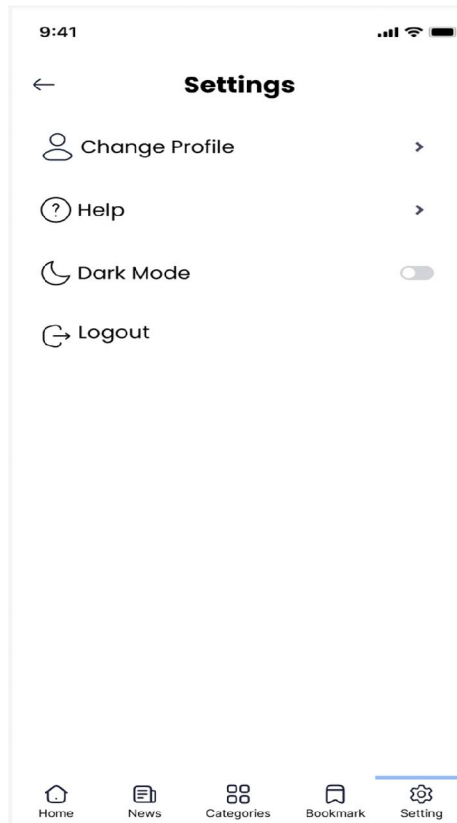
13) Bookmarks



14) Comment



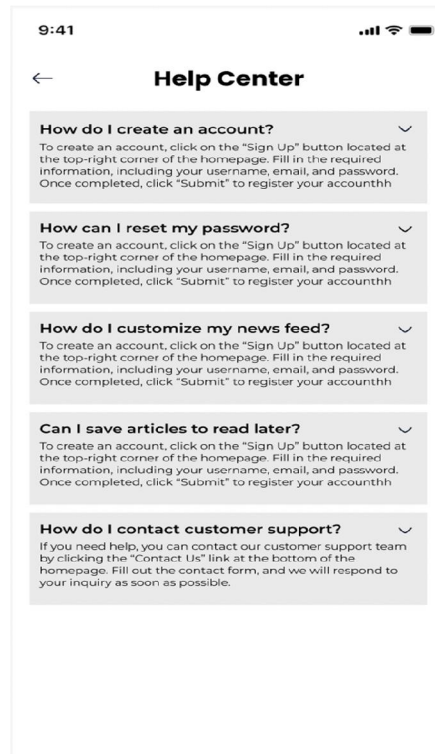
15) Setting



16) Change Information

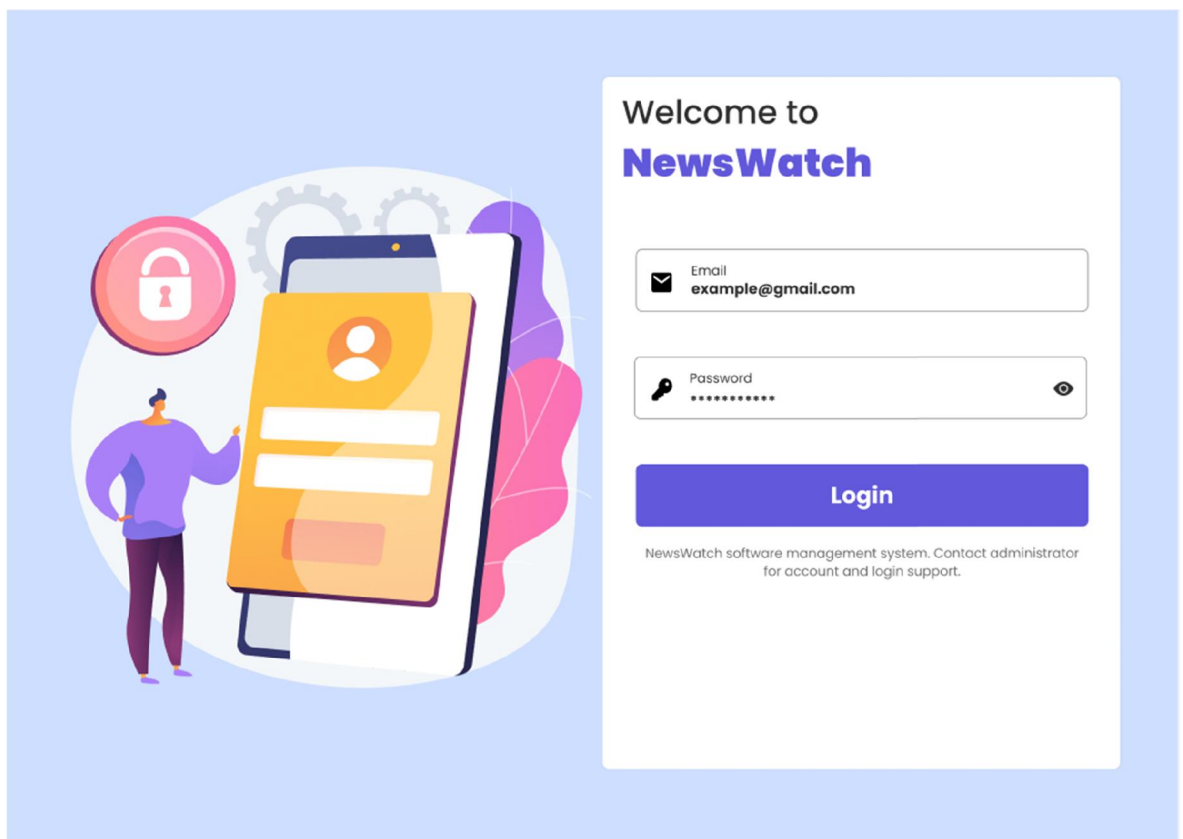
The screenshot shows the 'Change Profile' screen of a mobile application. At the top, the status bar displays the time '9:41' and signal/battery icons. Below the status bar is a back arrow and the title 'Change Profile'. The main content area features a large gray circular profile picture placeholder with a blue camera icon in the bottom right corner. Below the profile picture are four text input fields labeled 'Username', 'Full Name', 'Phone Number', and 'Email Address*'. At the bottom of the screen is a blue rounded rectangular button labeled 'Save'.

17) Help Center

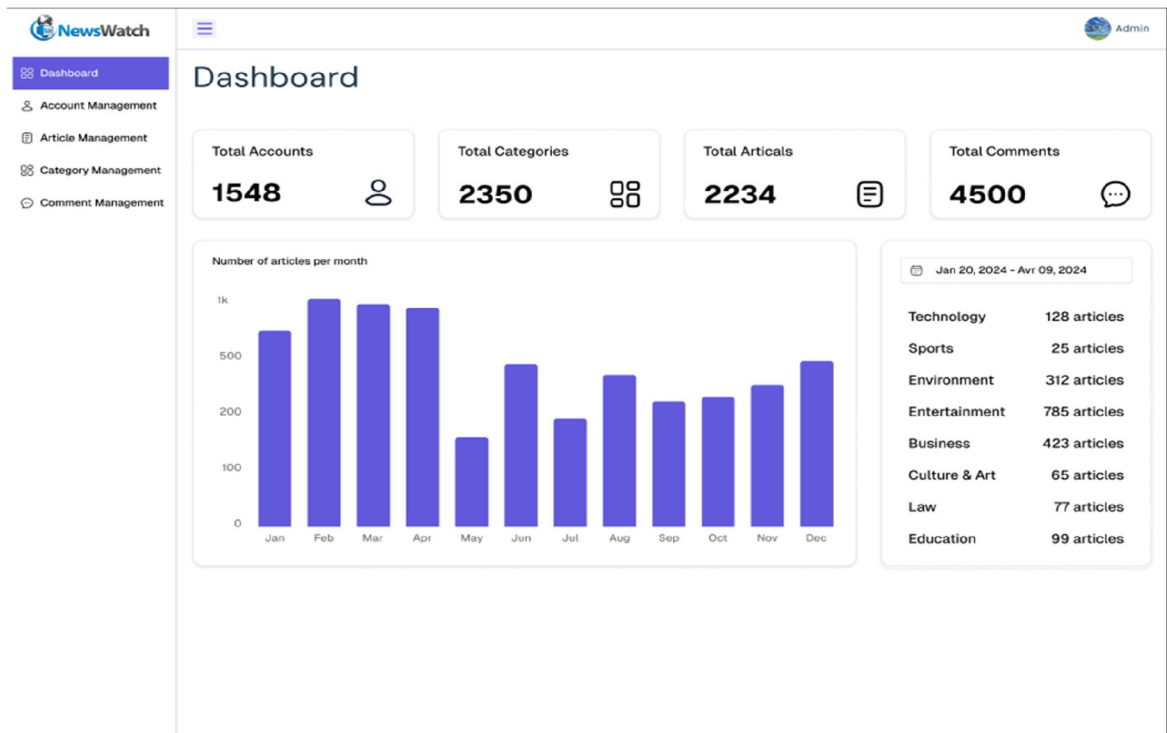


❖ Admin web

1) Login



2) Dashboard



3) Articles Management

- Trang hiển thị tất cả các bài viết, tin tức

The Article Management interface allows users to view and manage all articles. It includes a sidebar with navigation links for Dashboard, Account Management, Article Management, Category Management, and Comment Management. The main content area displays a table of articles with columns for Id, Image, Title, Category, Post Date, and Action. A pagination bar at the bottom shows the current page (1) and the total number of pages (8).

Id	Image	Title	Category	Post Date	Action
1		Let's settle the debate: IOS vs Androidz Consumer verdict	Technology	Sun 9 March, 2025	
1		Let's settle the debate: IOS vs Androidz Consumer verdict	Technology	Sun 9 March, 2025	
1		Let's settle the debate: IOS vs Androidz Consumer verdict	Technology	Sun 9 March, 2025	
1		Let's settle the debate: IOS vs Androidz Consumer verdict	Technology	Sun 9 March, 2025	
1		Let's settle the debate: IOS vs Androidz Consumer verdict	Technology	Sun 9 March, 2025	
1		Let's settle the debate: IOS vs Androidz Consumer verdict	Technology	Sun 9 March, 2025	
1		Let's settle the debate: IOS vs Androidz Consumer verdict	Technology	Sun 9 March, 2025	

- Thêm mới bài viết, tin tức

NewsWatch

Dashboard

Account Management

Article Management

Category Management

Comment Management

Admin

New Article

SaveCancel

Category

Title

Date Created:

Upload Image Banner For Article

Format: jpeg, png & Max file size: 25 MB

Browse Files

Description

- Sửa thông tin bài viết, tin tức

NewsWatch

Dashboard

Account Management

Article Management

Category Management

Comment Management

Admin

Edit Article

SaveCancel

Category

Title

Date Created:

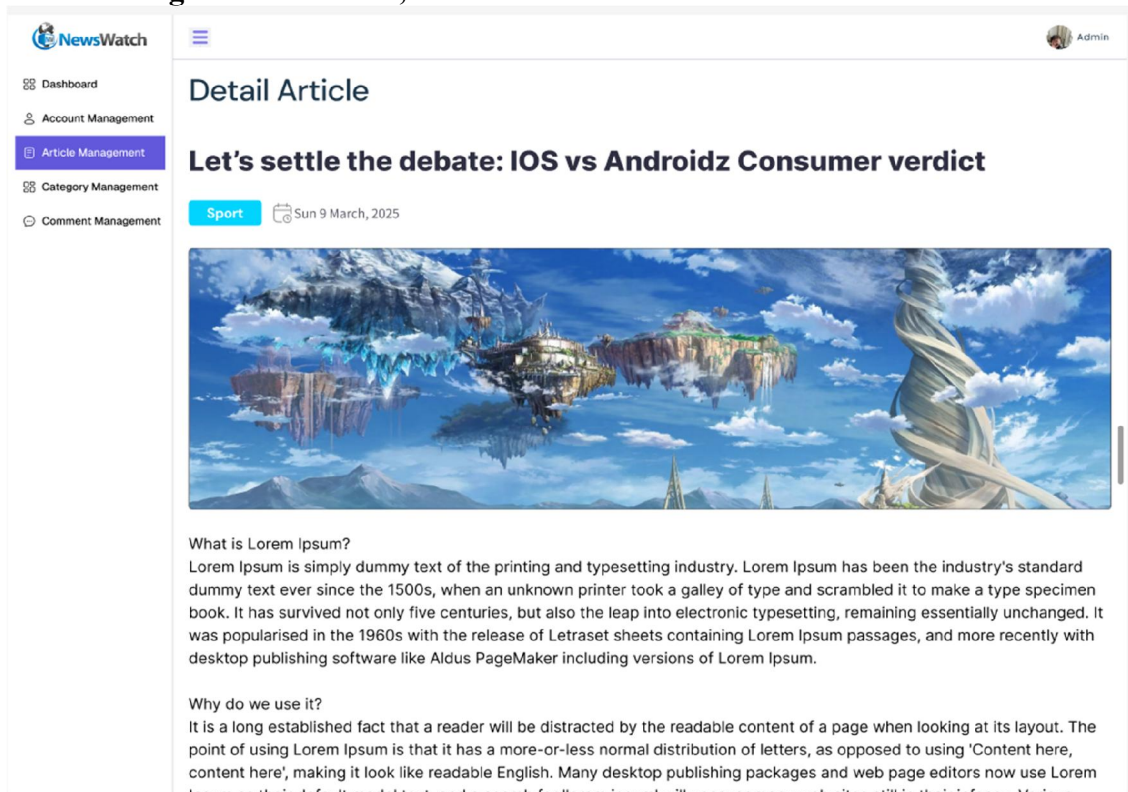
Upload Image Banner For Article

Format: jpeg, png & Max file size: 25 MB

Browse Files

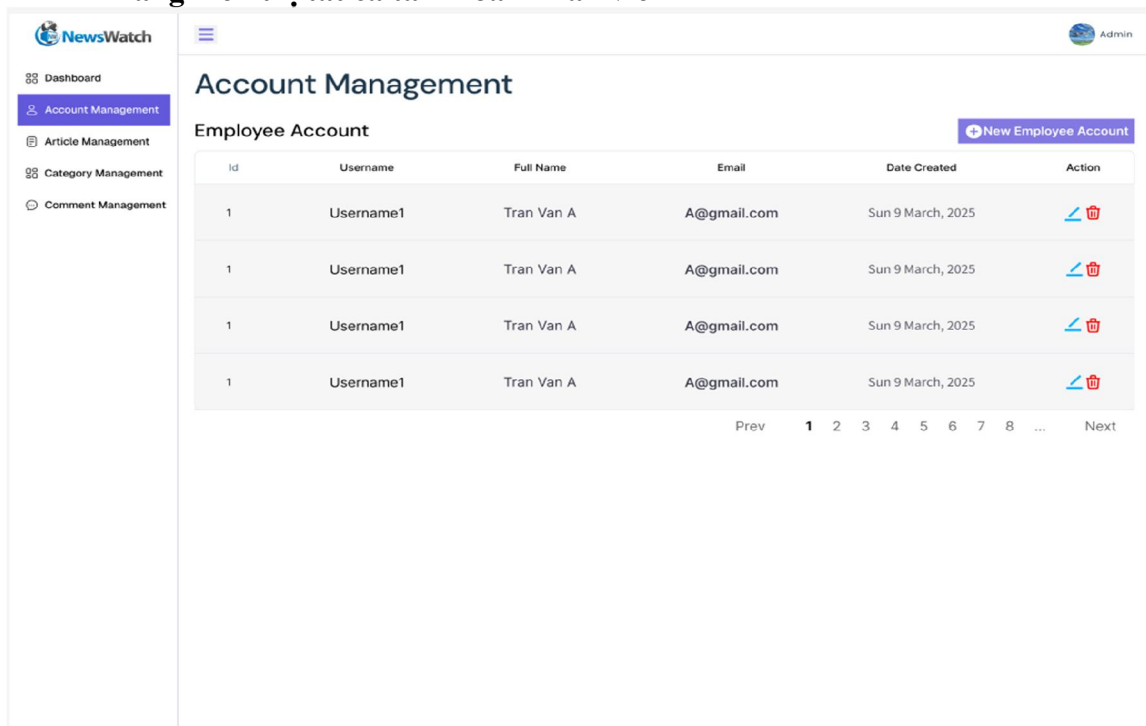
Description

- Trang chi tiết bài viết, tin tức




4) Account Management

- Trang hiển thị tất cả tài khoản nhân viên



- Trang thêm mới



Dashboard

Account Management

Article Management

Category Management

Comment Management

Admin

Add Employee Account

SaveCancel

Full Name

Date Created:


Username

Email

Password

Confirm Password

- Trang sửa thông tin



Dashboard

Account Management

Article Management

Category Management

Comment Management

Admin

Edit Employee Account

SaveCancel

Full Name

Date Created:

Username

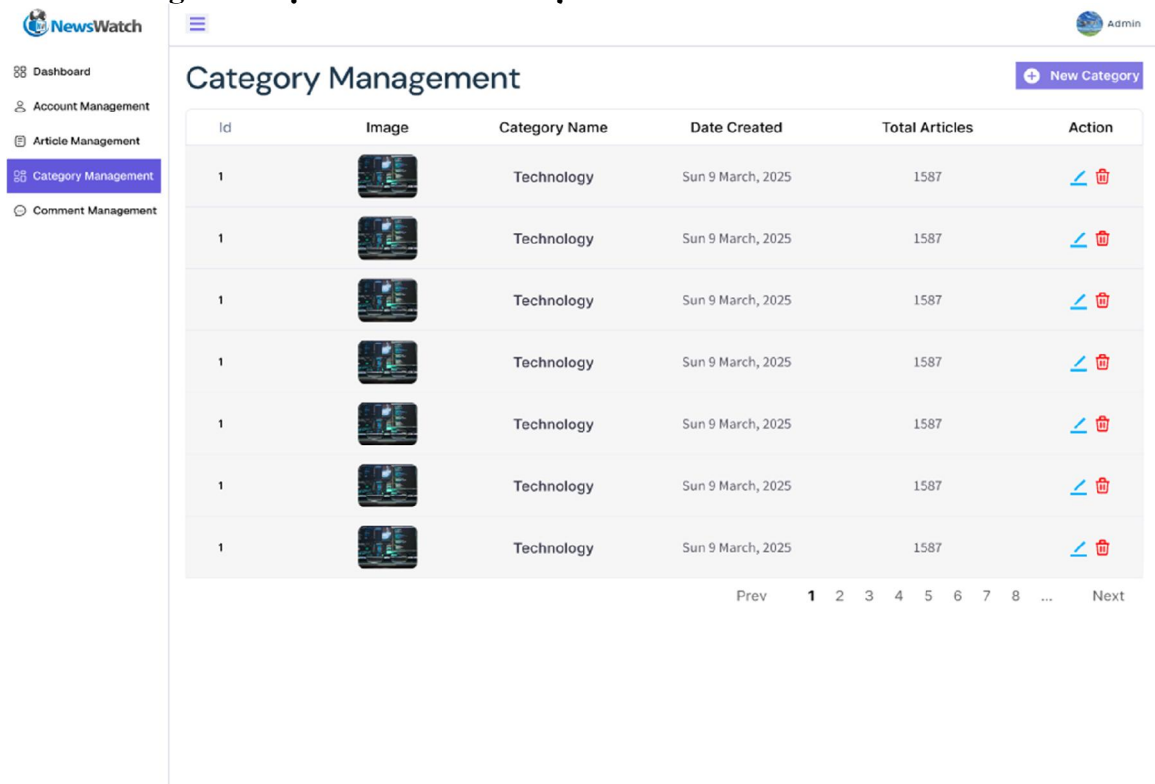
Email

Password

Confirm Password

5) Category Management

- Trang hiển thị tất cả các danh mục

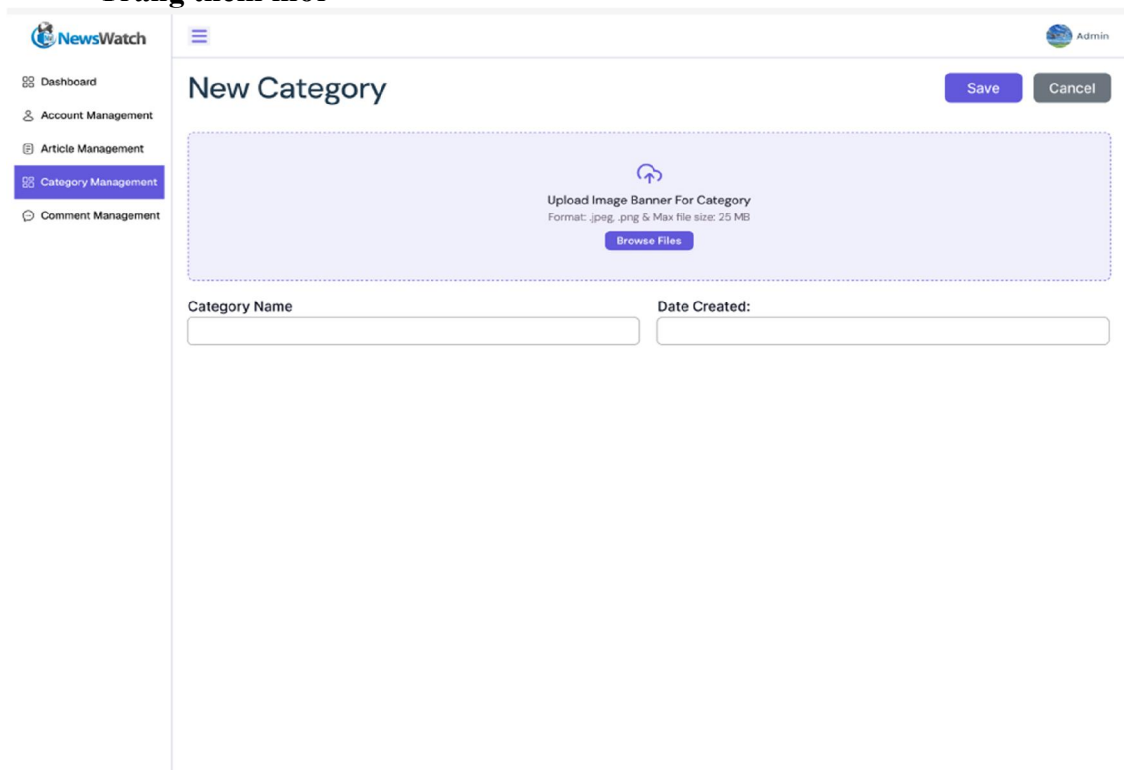


The screenshot displays the 'Category Management' page in the NewsWatch application. The left sidebar contains navigation links: Dashboard, Account Management, Article Management, Category Management (highlighted), and Comment Management. The top right shows the user 'Admin'. The main content area has a 'New Category' button and a table with the following data:

Id	Image	Category Name	Date Created	Total Articles	Action
1		Technology	Sun 9 March, 2025	1587	Edit Delete
1		Technology	Sun 9 March, 2025	1587	Edit Delete
1		Technology	Sun 9 March, 2025	1587	Edit Delete
1		Technology	Sun 9 March, 2025	1587	Edit Delete
1		Technology	Sun 9 March, 2025	1587	Edit Delete
1		Technology	Sun 9 March, 2025	1587	Edit Delete
1		Technology	Sun 9 March, 2025	1587	Edit Delete

At the bottom of the table, there is a pagination control showing 'Prev', '1' (selected), '2', '3', '4', '5', '6', '7', '8', '...', and 'Next'.

- Trang thêm mới



The screenshot shows the 'New Category' page in the NewsWatch application. The left sidebar is identical to the previous page. The main content area features a 'New Category' title, a 'Save' button, and a 'Cancel' button. Below the title is a large dashed box for uploading a category banner image, with the text 'Upload Image Banner For Category' and 'Format: jpeg, png & Max file size: 25 MB'. A 'Browse Files' button is located within this box. At the bottom, there are two input fields: 'Category Name' and 'Date Created:'.

- Trang sửa thông tin

- Dashboard
- Account Management
- Article Management
- Category Management**
- Comment Management

Admin

Edit Category

Save

Cancel

Upload Image Banner For Category

Format: .jpeg, .png & Max file size: 25 MB

Browse Files

Category Name

Date Created:

6) Change Information

- Dashboard
- Account Management
- Article Management
- Category Management**
- Comment Management

Admin

Change Information

Save

Full Name

Username

Birthday

Email

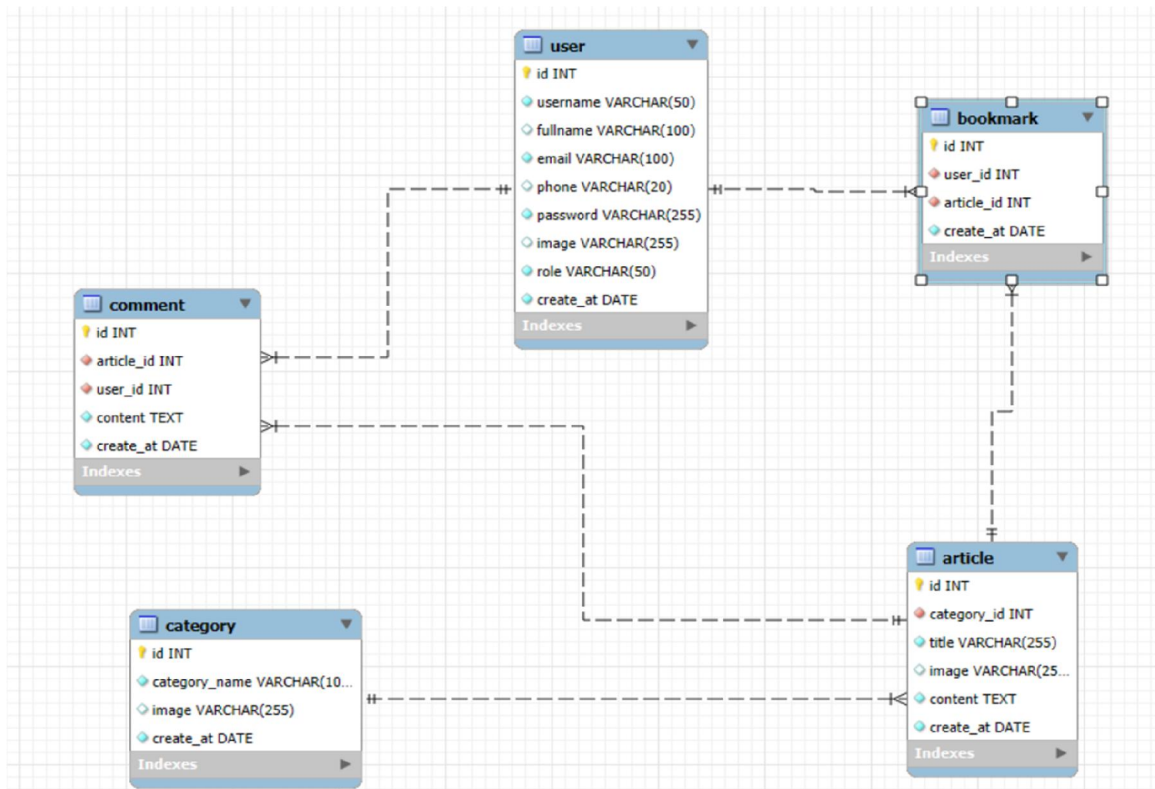
Phone

Password

Confirm Password

3.2. Thiết kế cơ sở dữ liệu

- Sơ đồ thiết kế cơ sở dữ liệu



1) **Bảng User (Người dùng):** Lưu trữ thông tin của người dùng trong hệ thống.

Tên trường	Kiểu	Mô tả	Ràng buộc
id	Integer	Định danh duy nhất cho mỗi người dùng	Khóa chính (PK)
username	Varchar(50)	Tên đăng nhập	
fullname	Varchar(100)	Họ và tên đầy đủ	
email	Varchar(100)	Địa chỉ email	
phone	Varchar(20)	Số điện thoại	
password	Varchar(255)	Mật khẩu (đã mã hóa)	
image	Varchar(255)	Ảnh đại diện người dùng	
role	Varchar(50)	Quyền của người dùng (admin, user,...)	
created_at	Date	Thời gian tạo tài khoản	

2) Bảng Article (Bài viết/Tin tức): Lưu trữ các bài báo hoặc tin tức.

Tên trường	Kiểu	Ý nghĩa	Ràng buộc
id	Integer	Định danh duy nhất cho bài viết	Khóa chính (Primary Key)
category_id	Integer	Liên kết với bảng Category để xác định danh mục bài viết	Khóa ngoại (Foreign Key)
title	Varchar(255)	Tiêu đề bài viết, tin tức	
image	Varchar(255)	Ảnh cho bài viết, tin tức	
content	Text	Nội dung bài viết, tin tức	
created_at	Date	Thời gian đăng bài viết, tin tức	

3) Bảng Category (Danh mục bài viết): Lưu danh mục của bài viết, giúp phân loại tin tức theo chủ đề.

Tên trường	Kiểu	Ý nghĩa	Ràng buộc
id	Integer	Định danh duy nhất cho danh mục	Khóa chính (Primary Key)
category_name	Varchar(100)	Tên danh mục (Ví dụ: Thể thao, Công nghệ, Giải trí,...).	
image	Varchar(255)	Ảnh mô tả cho danh mục	
created_at	Date	Thời gian tạo danh mục	

4) Bảng Comment (Bình luận): Lưu trữ bình luận của người dùng về các bài viết.

Tên trường	Kiểu	Ý nghĩa	Chú thích
id	Integer	Định danh duy nhất cho bình luận	Khóa chính (Primary Key)
article_id	Integer	Liên kết với bảng Article, xác định bình luận đó thuộc về bài viết nào.	Khóa ngoại (Foreign Key)
user_id	Integer	Liên kết với bảng User, xác định người viết bình luận	Khóa ngoại (Foreign Key)
content	Text	Nội dung bình luận.	
created_at	Date	Thời gian tạo bình luận	

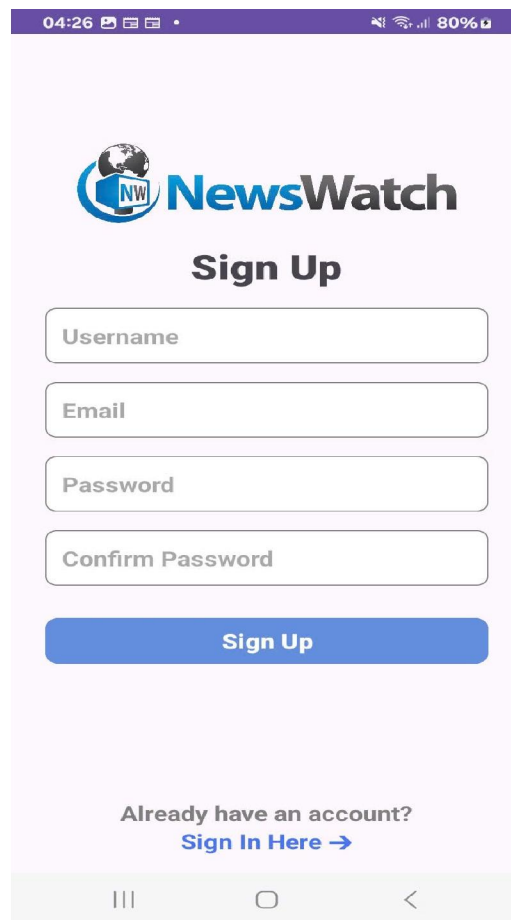
5) Bảng Bookmark (Lưu bài viết yêu thích): Lưu trữ các bài viết mà người dùng đã đánh dấu yêu thích.

Tên trường	Kiểu	Ý nghĩa	Chú thích
id	Integer	Định danh duy nhất cho mỗi bookmark	Khóa chính (Primary Key)
user_id	Integer	Liên kết với bảng User, xác định người lưu bài viết.	Khóa ngoại (Foreign Key)
article_id	Integer	Liên kết với bảng Article, xác định bài viết được người dùng lưu.	Khóa ngoại (Foreign Key)
created_at	Date	Thời gian lưu bài viết	

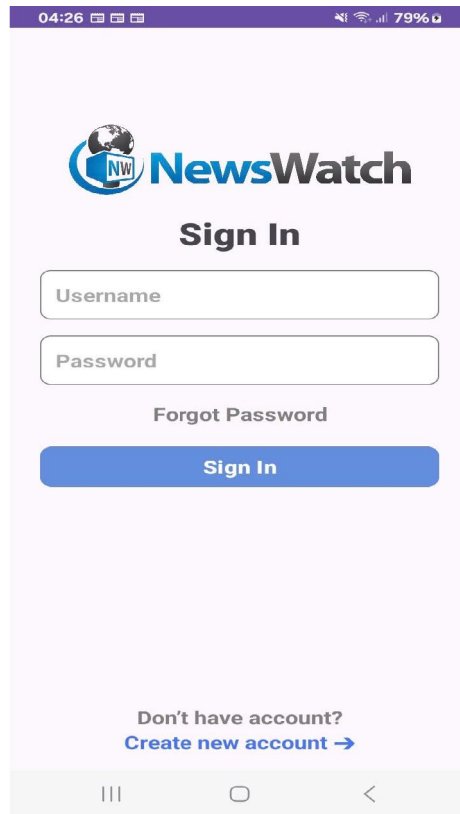
- Mối quan hệ giữa các bảng:
 1. **User - Comment(1-N)**: Một người dùng có thể bình luận nhiều lần, mỗi bình luận thuộc về một bài viết.
 2. **User - Bookmark(1-N)**: Một người dùng có thể lưu nhiều bài viết.
 3. **Article - Comment(1-N)**: Một bài viết có thể có nhiều bình luận.
 4. **Article - Category(N-1)**: Mỗi bài viết thuộc một danh mục cụ thể.

3.3. Giao diện ứng dụng

3.3.1. Màn hình đăng ký tài khoản

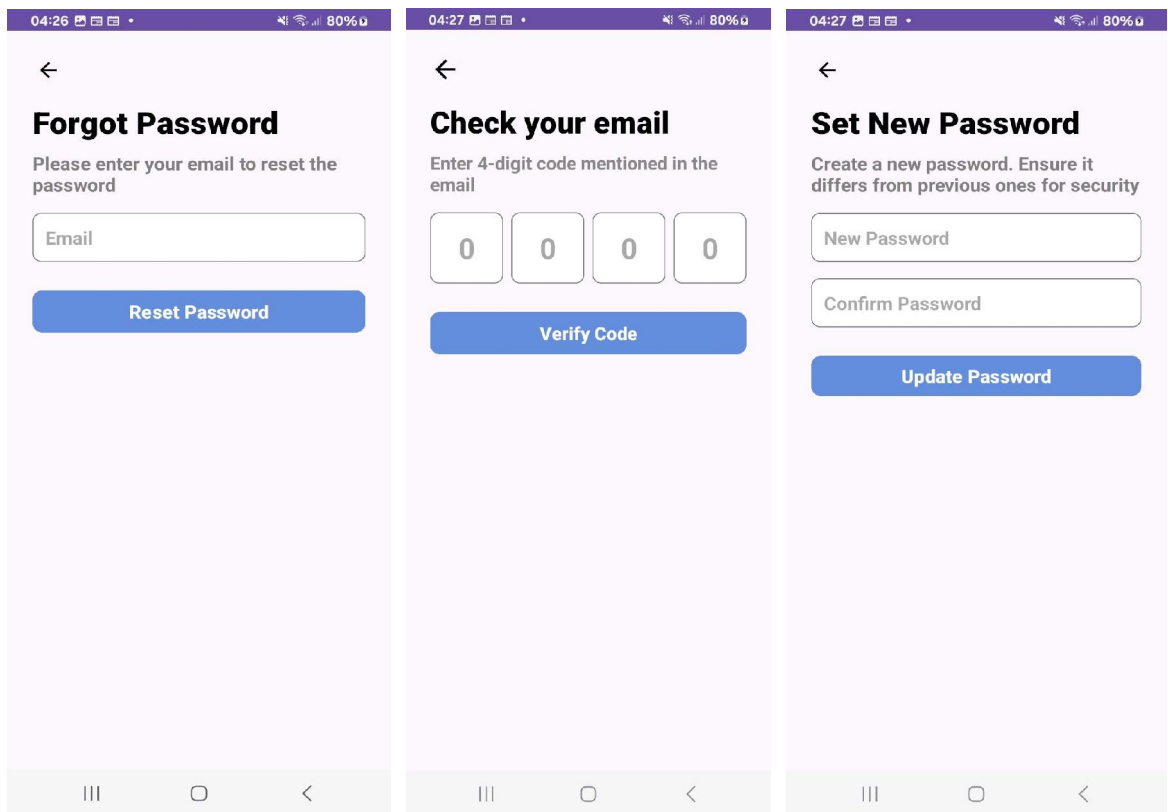


3.3.2. Màn hình đăng nhập ứng dụng



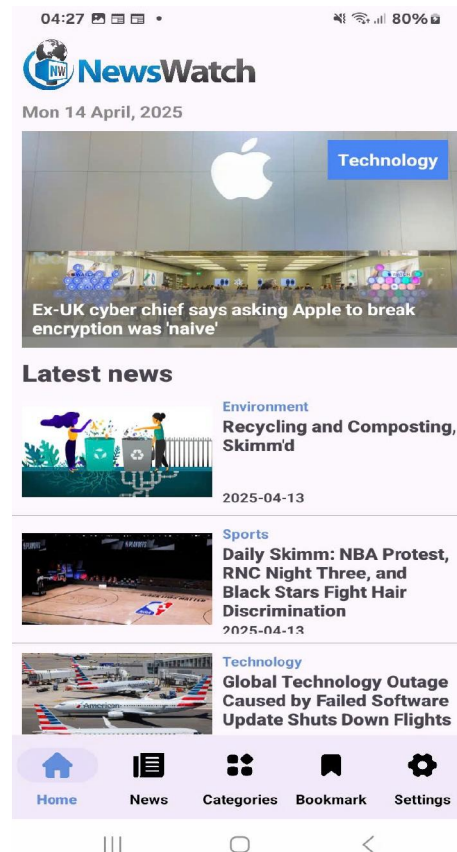
The screenshot shows the 'Sign In' screen of the NewsWatch application. At the top, there is a status bar with the time 04:26, signal strength, and 79% battery. Below the status bar is the NewsWatch logo, which consists of a blue circle with a white 'NW' inside, followed by the text 'NewsWatch' in a bold, sans-serif font. Under the logo, the text 'Sign In' is centered. Below this, there are two input fields: 'Username' and 'Password'. Below the 'Password' field, there is a link 'Forgot Password'. At the bottom of the form, there is a blue button labeled 'Sign In'. Below the button, there is a link 'Don't have account? Create new account →'. At the very bottom, there is a navigation bar with three icons: a list icon, a home icon, and a back icon.

3.3.3. Màn hình quên mật khẩu (Forgot Password, Check OTP, Set New PassWord)

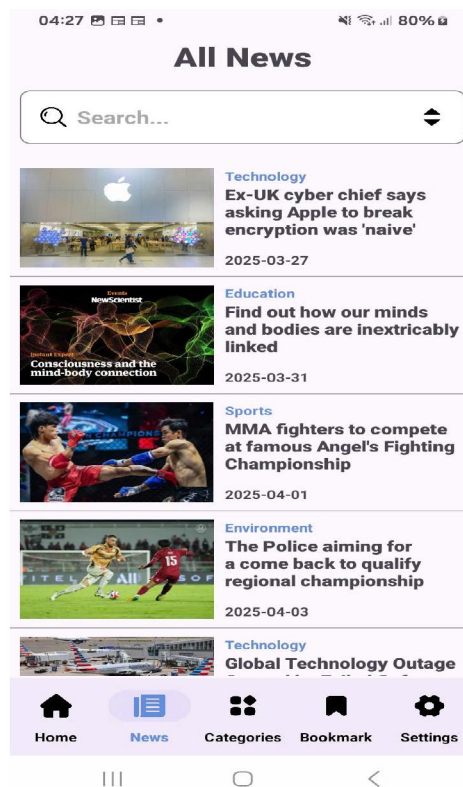


The three screenshots show the password reset process in the NewsWatch application. The first screenshot is titled 'Forgot Password' and asks the user to enter their email to reset the password. It has an 'Email' input field and a 'Reset Password' button. The second screenshot is titled 'Check your email' and asks the user to enter a 4-digit code mentioned in the email. It has four input fields for the code and a 'Verify Code' button. The third screenshot is titled 'Set New Password' and asks the user to create a new password, ensuring it differs from previous ones for security. It has 'New Password' and 'Confirm Password' input fields and an 'Update Password' button. All three screenshots have a back arrow at the top left and a navigation bar at the bottom with three icons: a list icon, a home icon, and a back icon.

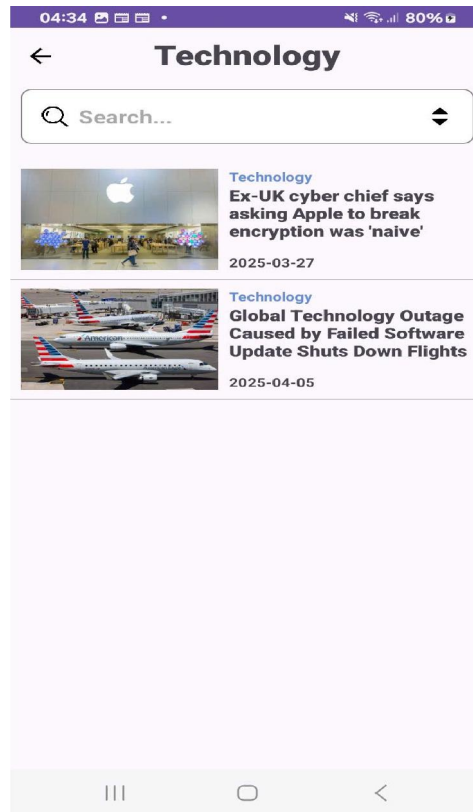
3.3.4. Màn hình trang chính của ứng dụng



3.3.5. Màn hình danh sách tất cả tin tức



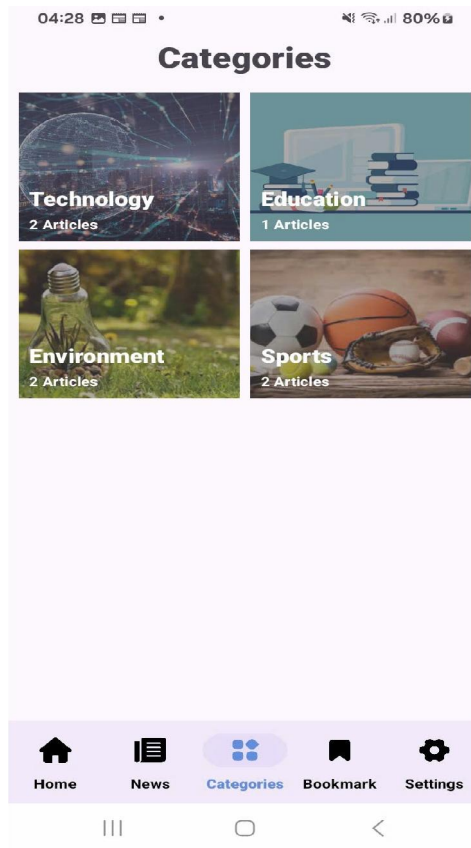
3.3.6. Màn hình tìm kiếm tin tức



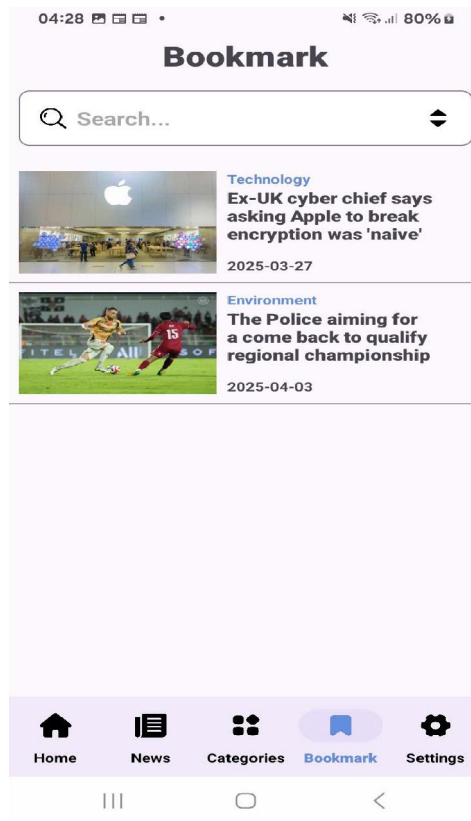
3.3.7. Màn hình chi tiết tin tức



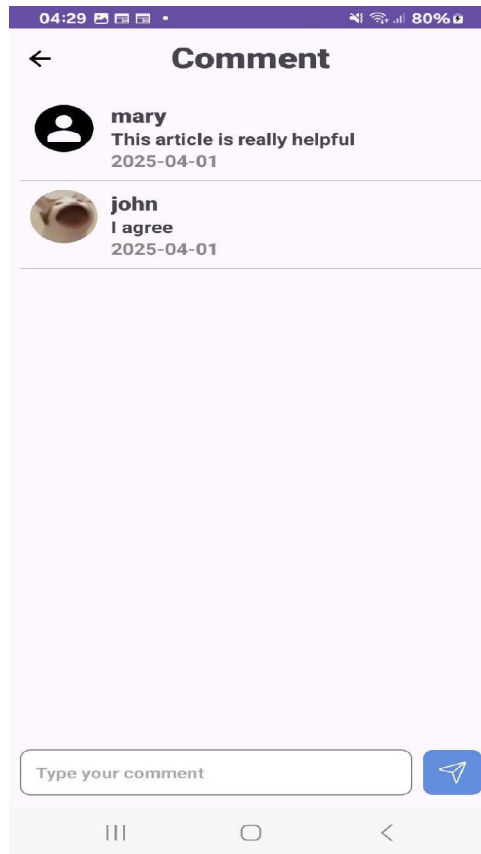
3.3.8. Màn hình danh mục tin tức



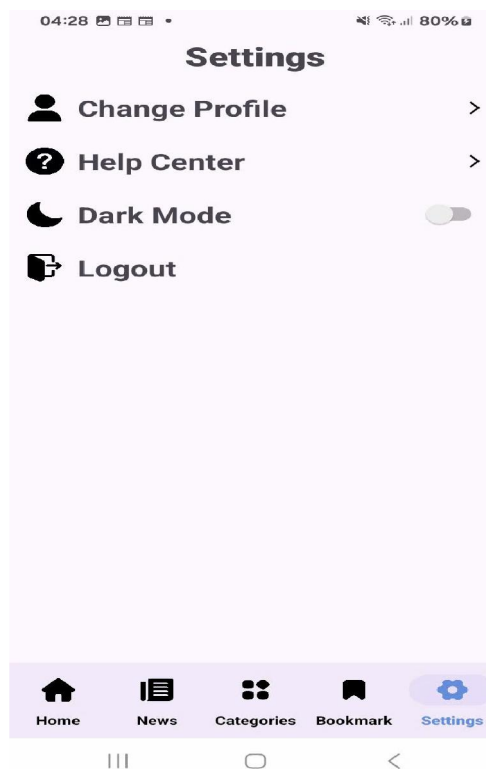
3.3.9. Màn hình danh sách tin tức được lưu



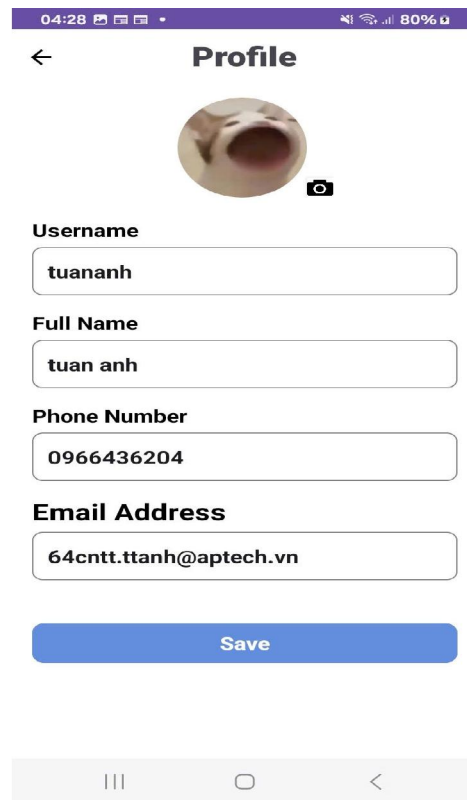
3.3.10.Màn hình bình luận



3.3.11.Màn hình cài đặt




3.3.12. Màn hình thông tin tài khoản



04:28 80%

← Profile



Username

tuannah

Full Name

tuàn anh

Phone Number

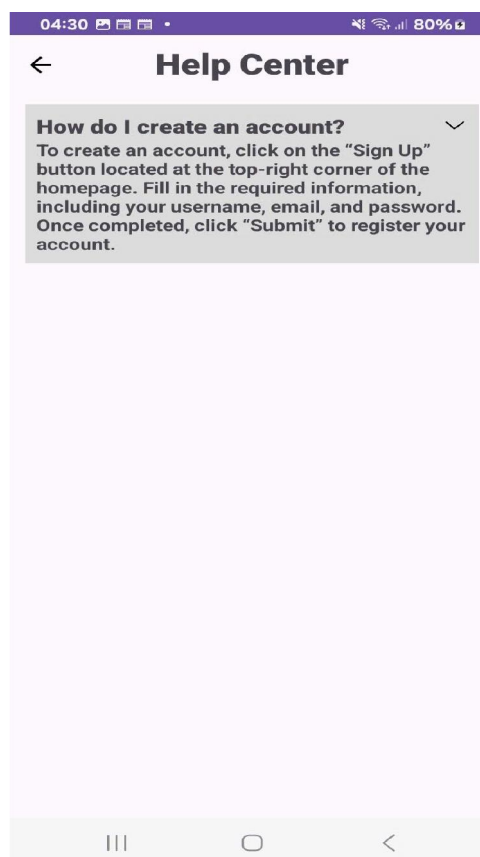
0966436204

Email Address

64cntt.tanh@aptech.vn

Save

3.3.13. Màn hình trung tâm trợ giúp



04:30 80%

← Help Center

How do I create an account? ✓

To create an account, click on the "Sign Up" button located at the top-right corner of the homepage. Fill in the required information, including your username, email, and password. Once completed, click "Submit" to register your account.

3.4. Code minh họa các chức năng cốt lõi

3.4.1. Đăng nhập

```
public class LoginActivity extends AppCompatActivity {
    private EditText etUsername, etPassword;
    private MaterialButton btnSignIn;
    private TextView txtSignUp, txtForgot;
    private SharedPreferences sharedPreferences;
    private static final String SHARED_PREF_NAME = "shared_user";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.login);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.signIn), (v, insets) ->
        {
            int systemBars = WindowInsetsCompat.Type.systemBars();
            v.setPadding(insets.getInsets(systemBars).left,
                insets.getInsets(systemBars).top,
                insets.getInsets(systemBars).right,
                insets.getInsets(systemBars).bottom);
            return insets;
        });
        txtSignUp = findViewById(R.id.tvSignUp);
        txtSignUp.setOnClickListener(v -> {
            Intent intent = new Intent(LoginActivity.this, RegisterActivity.class);
            startActivity(intent);
            finish();
        });
        txtForgot = findViewById(R.id.forgotPassword);
        txtForgot.setOnClickListener(v -> {
            Intent intent = new Intent(LoginActivity.this, ForgotPassword.class);
            startActivity(intent);
        });
        sharedPreferences = getSharedPreferences(SHARED_PREF_NAME, MODE_PRIVATE);
        etUsername = findViewById(R.id.etUsername);
        etPassword = findViewById(R.id.etPassword);
        btnSignIn = findViewById(R.id.btnSignIn);
        btnSignIn.setOnClickListener(v -> {
            String username = etUsername.getText().toString().trim();
            String password = etPassword.getText().toString().trim();

            if (username.isEmpty()) {
                etUsername.setError("Username is required");
                return;
            }
            if (password.isEmpty()) {
                etPassword.setError("Password is required");
                return;
            }
            loginUser(username, password);
        });
    }
}
```

```

private void loginUser(String username, String password){
    RetrofitService retrofitService = new RetrofitService(username, password);
    ApiService apiService = retrofitService.getRetrofit().create(ApiService.class);
    LoginRequest loginRequest = new LoginRequest(username, password);
    apiService.login(loginRequest).enqueue(new Callback<User>(){
        @Override
        public void onResponse(Call<User> call, Response<User> response){
            if (response.isSuccessful() && response.body() != null) {
                User user = response.body();

                SharedPreferences.Editor editor = sharedPreferences.edit();
                editor.putInt("id", user.getId());
                editor.putString("username", username);
                editor.putString("email", user.getEmail());
                editor.putString("phone", user.getPhone());
                editor.putString("fullname", user.getFullname());
                editor.putString("password", password);
                editor.putString("image", user.getImage());
                editor.putString("role", user.getRole());
                editor.apply();

                Intent intent = new Intent(LoginActivity.this, MainActivity.class);
                startActivity(intent);
                finish();
            } else {
                Toast.makeText(LoginActivity.this, "Login failed" + response.message(),
                    Toast.LENGTH_SHORT).show();
            }
        }
        @Override
        public void onFailure(Call<User> call, Throwable t) {
            Toast.makeText(LoginActivity.this, "Error" + t.getMessage(),
                Toast.LENGTH_SHORT).show();
        }
    });
}

```

3.4.2. Đăng ký

```

public class RegisterActivity extends AppCompatActivity {
    private EditText etUsername, etEmail, etPassword, etConfirmPassword;
    private MaterialButton btnSignUp;
    private RetrofitService retrofitService;
    private ApiService apiService;
    private String username, email, password, confirmPassword;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.register);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.signUp), (v, insets) -
        > {

```



```

        int systemBars = WindowInsetsCompat.Type.systemBars();
        v.setPadding(insets.getInsets(systemBars).left,
            insets.getInsets(systemBars).top,
            insets.getInsets(systemBars).right,
            insets.getInsets(systemBars).bottom);
        return insets;
    });
    TextView txtSignIn = findViewById(R.id.tvSignIn);
    txtSignIn.setOnClickListener(v -> {
        Intent intent = new Intent(RegisterActivity.this, LoginActivity.class);
        startActivity(intent);
        finish();
    });
    etUsername = findViewById(R.id.etUsername);
    etEmail = findViewById(R.id.etEmail);
    etPassword = findViewById(R.id.etPassword);
    etConfirmPassword = findViewById(R.id.etConfirmPassword);
    btnSignUp = findViewById(R.id.btnSignUp);
    apiService = new RetrofitService().getRetrofit().create(ApiService.class);
    btnSignUp.setOnClickListener(view -> {
        username = etUsername.getText().toString().trim();
        email = etEmail.getText().toString().trim();
        password = etPassword.getText().toString().trim();
        confirmPassword = etConfirmPassword.getText().toString().trim();
        if (isValidInput()) {
            registerUser();
        }
    });
}

private boolean isValidInput() {
    if (username.isEmpty()) {
        etUsername.setError("Username is required");
        return false;
    }
    if (email.isEmpty()) {
        etEmail.setError("Email is required");
        return false;
    }
    if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
        etEmail.setError("Invalid email format");
        return false;
    }
    if (password.isEmpty()) {
        etPassword.setError("Password is required");
        return false;
    }
    if (!password.equals(confirmPassword)) {
        etConfirmPassword.setError("Passwords do not match");
        return false;
    }
    return true;
}
}

```

```

private void registerUser() {
    User newUser = new User(
        username,
        email,
        null,
        password,
        null,
        null,
        "user",
        new Date()
    );
    newUser.setUsername(username);
    newUser.setEmail(email);
    newUser.setPassword(password);
    newUser.setRole("user");
    apiService.registerUser(newUser).enqueue(new Callback<ResponseBody>() {
        @Override
        public void onResponse(Call<ResponseBody> call, Response<ResponseBody>
response) {
            if (response.isSuccessful() && response.body() != null) {
                try {
                    String message = response.body().string();
                    Toast.makeText(RegisterActivity.this, "Registration success",
Toast.LENGTH_SHORT).show();
                    startActivity(new Intent(RegisterActivity.this, LoginActivity.class));
                    finish();
                } catch (Exception e) {
                    e.printStackTrace();
                    Toast.makeText(RegisterActivity.this, "Fail", Toast.LENGTH_SHORT).show();
                }
            } else {
                Toast.makeText(RegisterActivity.this, "Registration failed" + response.message(),
Toast.LENGTH_SHORT).show();
            }
        }
        @Override
        public void onFailure(Call<ResponseBody> call, Throwable t) {
            Toast.makeText(RegisterActivity.this, "Error" + t.getMessage(),
Toast.LENGTH_SHORT).show();
        }
    });
}
}

```

3.4.3. Hiển thị danh sách tin tức

```

public class NewsFragment extends Fragment {
    private ListView newsListView;
    private ArticleAdapter articleAdapter;
    private ApiService apiService;
    private EditText searchBox;
    private ImageView filterIcon;
}

```

```

private List<Article> originalArticleList;
private boolean isSortedAscending = false;

public NewsFragment() {

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup
container, @Nullable Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_news, container, false);
        newsListView = view.findViewById(R.id.newsListView);

        SharedPreferences sharedPreferences =
requireContext().getSharedPreferences("shared_user", getApplicationContext().MODE_PRIVATE);
        String username = sharedPreferences.getString("username", null);
        String password = sharedPreferences.getString("password", null);
        apiService = new RetrofitService(username,
password).getRetrofit().create(ApiService.class);

        fetchArticles();

        searchBox = view.findViewById(R.id.searchBox);
        searchBox.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int start, int count, int after) {}
            @Override
            public void onTextChanged(CharSequence s, int start, int before, int count) {
                filterArticles(s.toString());
            }
            @Override
            public void afterTextChanged(Editable s) {}
        });

        filterIcon = view.findViewById(R.id.filterIcon);
        filterIcon.setOnClickListener(v -> sortArticlesByDate());

        return view;
    }

    private void fetchArticles() {
        apiService.getAllArticles().enqueue(new Callback<List<Article>>() {
            @Override
            public void onResponse(Call<List<Article>> call, Response<List<Article>> response)
{
                if (response.isSuccessful() && response.body() != null) {
                    originalArticleList = response.body();
                    articleAdapter = new ArticleAdapter(getApplicationContext(), originalArticleList);
                    newsListView.setAdapter(articleAdapter);
                } else {
                    Toast.makeText(getApplicationContext(), "Fail", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }

```

```

newsListView.setOnItemClickListener((parent, view, position, id) -> {
    Article selectedArticle = (Article) parent.getItemAtPosition(position);
    Intent intent = new Intent(getContext(), NewsDetailActivity.class);
    intent.putExtra("article_id", selectedArticle.getId());
    startActivity(intent);
});
}

@Override
public void onFailure(Call<List<Article>> call, Throwable t) {
    Toast.makeText(getContext(), "Error", Toast.LENGTH_SHORT).show();
}
});
}

private void filterArticles(String keyword) {
    List<Article> filteredList = new ArrayList<>();
    for (Article article : originalArticleList) {
        if (article.getTitle().toLowerCase().contains(keyword.toLowerCase())) {
            filteredList.add(article);
        }
    }
    if (articleAdapter != null) {
        articleAdapter.updateData(filteredList);
    }

    RelativeLayout noResultsLayout = getView().findViewById(R.id.noResultsLayout);
    if (noResultsLayout != null) {
        noResultsLayout.setVisibility(filteredList.isEmpty() ? View.VISIBLE : View.GONE);
    }
}

private void sortArticlesByDate() {
    if (originalArticleList != null) {
        if (isSortedAscending) {
            Collections.sort(originalArticleList, (a1, a2) ->
a2.getCreatedAt().compareTo(a1.getCreatedAt()));
        } else {
            Collections.sort(originalArticleList, (a1, a2) ->
a1.getCreatedAt().compareTo(a2.getCreatedAt()));
        }
        isSortedAscending = !isSortedAscending;
        articleAdapter.updateData(originalArticleList);
    }
}
}
}

```

3.4.4. Chi tiết tin tức

```

public class NewsDetailActivity extends AppCompatActivity {
    private ImageView newsImage;
    private TextView tvCategory, tvTitle, tvDate, tvContent;
    private ImageButton backButton;
}

```

```

private ImageView btnBookmark, btnShare, btnComment;
private int userId, articleId;
private SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd",
Locale.getDefault());

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.news_detail);

    newsImage = findViewById(R.id.newsImage);
    tvCategory = findViewById(R.id.tvCategory);
    tvTitle = findViewById(R.id.tvTitle);
    tvDate = findViewById(R.id.tvDate);
    tvContent = findViewById(R.id.tvContent);
    backButton = findViewById(R.id.backButton);
    btnComment = findViewById(R.id.btnComment);
    backButton.setOnClickListener(v -> finish());

    SharedPreferences sharedPreferences = getSharedPreferences("shared_user",
MODE_PRIVATE);
    String username = sharedPreferences.getString("username", null);
    String password = sharedPreferences.getString("password", null);
    userId = sharedPreferences.getInt("id", -1);

    articleId = getIntent().getIntExtra("article_id", -1);
    if (username != null && password != null) {
        if (articleId != -1) {
            loadArticleDetail(articleId, username, password);
        } else {
            Toast.makeText(this, "Fail", Toast.LENGTH_SHORT).show();
        }
        btnComment.setOnClickListener(view -> {
            Intent intent = new Intent(NewsDetailActivity.this, CommentActivity.class);
            intent.putExtra("articleId", articleId);
            startActivity(intent);
        });
    }
    ApiService apiService = new RetrofitService(username,
password).getRetrofit().create(ApiService.class);
    apiService.getBookmarks(userId).enqueue(new Callback<List<Bookmark>>() {
        @Override
        public void onResponse(Call<List<Bookmark>> call, Response<List<Bookmark>>
response) {
            if (response.isSuccessful() && response.body() != null) {
                boolean isBookmarked = response.body().stream()
                    .anyMatch(b -> b.getArticle().getId() == articleId && b.getUser().getId() ==
userId);
                if (isBookmarked) {
                    btnBookmark.setColorFilter(Color.parseColor("#648DDB"));
                }
            }
        }
    });
}

```

```

    }
    @Override
    public void onFailure(Call<List<Bookmark>> call, Throwable t) {
        Toast.makeText(NewsDetailActivity.this, "Error checking bookmark: " +
t.getMessage(), Toast.LENGTH_SHORT).show();
    }
});

}

private void loadArticleDetail(int articleId, String username, String password) {
    ApiService apiService = new RetrofitService(username,
password).getRetrofit().create(ApiService.class);

    Call<Article> call = apiService.getArticleById(articleId);
    call.enqueue(new Callback<Article>() {
        @Override
        public void onResponse(Call<Article> call, Response<Article> response) {
            if (response.isSuccessful() && response.body() != null) {
                Article article = response.body();
                tvTitle.setText(article.getTitle());
                tvContent.setText(article.getContent());

                if (article.getCreateAt() != null) {
                    tvDate.setText(dateFormat.format(article.getCreateAt()));
                } else {
                    tvDate.setText("Unknown Date");
                }

                if (article.getCategoryName() != null) {
                    tvCategory.setText(article.getCategoryName());
                } else {
                    tvCategory.setText("Unknown Category");
                }
                Glide.with(NewsDetailActivity.this)
                    .load("http://192.168.2.13:8080/" + article.getImage())
                    .placeholder(R.drawable.placeholder)
                    .into(newsImage);
            } else {
                Toast.makeText(NewsDetailActivity.this, "Fail",
Toast.LENGTH_SHORT).show();
            }
        }
    });

    @Override
    public void onFailure(Call<Article> call, Throwable t) {
        Toast.makeText(NewsDetailActivity.this, "Error" + t.getMessage(),
Toast.LENGTH_LONG).show();
    }
});
}

```

3.4.5. Lưu bài viết

```
btnBookmark.setOnClickListener(v -> {
    if (userId != -1 && articleId != -1) {
        ApiService apiService = new RetrofitService(username,
password).getRetrofit().create(ApiService.class);

        apiService.getBookmarks(userId).enqueue(new Callback<List<Bookmark>>() {
            @Override
            public void onResponse(Call<List<Bookmark>> call, Response<List<Bookmark>>
response) {
                if (response.isSuccessful() && response.body() != null) {
                    List<Bookmark> bookmarks = response.body();
                    Optional<Bookmark> existingBookmark = bookmarks.stream()
                        .filter(b -> b.getArticle().getId() == articleId && b.getUser().getId() ==
userId)
                        .findFirst();

                    if (existingBookmark.isPresent()) {
                        int bookmarkId = existingBookmark.get().getId();
                        apiService.deleteBookmark(bookmarkId).enqueue(new Callback<Void>() {
                            @Override
                            public void onResponse(Call<Void> call, Response<Void> response) {
                                if (response.isSuccessful()) {
                                    Toast.makeText(NewsDetailActivity.this, "Bookmark removed",
Toast.LENGTH_SHORT).show();
                                    btnBookmark.setColorFilter(null);
                                    SharedPreferences prefs = getSharedPreferences("bookmark_pref",
MODE_PRIVATE);
                                    prefs.edit().putBoolean("bookmark_changed", true).apply();
                                } else {
                                    Toast.makeText(NewsDetailActivity.this, "Failed to remove bookmark",
Toast.LENGTH_SHORT).show();
                                }
                            }
                        });

                        @Override
                        public void onFailure(Call<Void> call, Throwable t) {
                            Toast.makeText(NewsDetailActivity.this, "Error: " + t.getMessage(),
Toast.LENGTH_SHORT).show();
                        }
                    });
                } else {
                    Bookmark bookmark = new Bookmark();
                    bookmark.setUserId(userId);
                    bookmark.setArticleId(articleId);
                    bookmark.setCreateAt(new Date());
                    apiService.sendBookmark(bookmark).enqueue(new Callback<Bookmark>() {
                        @Override
                        public void onResponse(Call<Bookmark> call, Response<Bookmark>
response) {
                            if (response.isSuccessful()) {
                                Toast.makeText(NewsDetailActivity.this, "Article saved", Toast.LENGTH_SHORT).show();
                            }
                        }
                    });
                }
            }
        });
    }
});
```



```

        backButton.setOnClickListener(v -> finish());

        listComment = findViewById(R.id.listComment);
        noResultsLayout = findViewById(R.id.noResultsLayout);

        SharedPreferences sharedPreferences = getSharedPreferences("shared_user",
        MODE_PRIVATE);
        String username = sharedPreferences.getString("username", null);
        String password = sharedPreferences.getString("password", null);
        apiService = new RetrofitService(username,
        password).getRetrofit().create(ApiService.class);

        articleId = getIntent().getIntExtra("articleId", 0);
        if (articleId > 0) {
            loadComments(articleId);
        }

        editComment = findViewById(R.id.addComment).findViewById(R.id.etComment);
        btnSend = findViewById(R.id.btnSend);

        btnSend.setOnClickListener(v -> {
            String content = editComment.getText().toString().trim();
            if (content.isEmpty()) {
                Toast.makeText(CommentActivity.this, "Please enter a comment",
                Toast.LENGTH_SHORT).show();
                return;
            }
            int userId = sharedPreferences.getInt("id", 0);
            if (userId == 0) {
                Toast.makeText(CommentActivity.this, "User not logged in",
                Toast.LENGTH_SHORT).show();
                return;
            }
            Comment newComment = new Comment();
            newComment.setArticleId(articleId);
            newComment.setUserId(userId);
            newComment.setContent(content);
            newComment.setCreatedAt(new Date());

            apiService.addComment(articleId, newComment).enqueue(new
            Callback<Comment>() {
                @Override
                public void onResponse(Call<Comment> call, Response<Comment> response) {
                    if (response.isSuccessful() && response.body() != null) {
                        Toast.makeText(CommentActivity.this, "Comment posted successfully",
                        Toast.LENGTH_SHORT).show();
                        editComment.setText("");
                        loadComments(articleId);
                    } else {
                        Toast.makeText(CommentActivity.this, "Post comment failed",
                        Toast.LENGTH_SHORT).show();
                    }
                }
            });

```

```

    }

    @Override
    public void onFailure(Call<Comment> call, Throwable t) {
        Toast.makeText(CommentActivity.this, "Error: " + t.getMessage(),
            Toast.LENGTH_SHORT).show();
    }
});
}

private void loadComments(int articleId) {
    ApiService.getComments(articleId).enqueue(new Callback<List<Comment>>() {
        @Override
        public void onResponse(Call<List<Comment>> call, Response<List<Comment>>
            response) {
            if (response.isSuccessful() && response.body() != null &&
                !response.body().isEmpty()) {
                commentAdapter = new CommentAdapter(CommentActivity.this,
                    response.body());
                listComment.setAdapter(commentAdapter);
                noResultsLayout.setVisibility(View.GONE);
                listComment.setVisibility(View.VISIBLE);
            } else {
                noResultsLayout.setVisibility(View.VISIBLE);
                listComment.setVisibility(View.GONE);
            }
        }
    });

    @Override
    public void onFailure(Call<List<Comment>> call, Throwable t) {
        noResultsLayout.setVisibility(View.VISIBLE);
        listComment.setVisibility(View.GONE);
    }
});
}
}

```

KẾT LUẬN

1. Kết quả đạt được

- Nhóm cơ bản đã hoàn thiện dự án ứng dụng phần mềm đọc tin tức **News Watch** trên nền tảng Android với đầy đủ các chức năng cơ bản của một ứng dụng tin tức
- Giao diện người dùng được thiết kế đơn giản, dễ sử dụng, tương thích với nhiều kích thước màn hình.
- Sử dụng API để có thể bảo trì dễ dàng, dễ kiểm thử, bảo mật tốt
- Thông tin của người dùng được bảo mật, tránh bị lộ thông tin

2. Nhược điểm

- Chưa hỗ trợ đa ngôn ngữ, giới hạn ngôn ngữ chỉ ở tiếng Anh.
- Hệ thống chưa có các chức năng tùy chỉnh giao diện theo ý của người dùng
- Hệ thống vẫn có độ trễ nhất định khi sử dụng

3. Hướng phát triển

- Hỗ trợ đa ngôn ngữ để mở rộng khả năng sử dụng ứng dụng cho người dùng toàn cầu.
- Cải thiện các chức năng có thể thay đổi giao diện theo ý người dùng
- Giải quyết vấn đề về độ trễ, cải thiện để có thể sử dụng ứng dụng mà không cần Internet
- Có thể tích hợp AI vào 1 số các chức năng, nhằm tăng trải nghiệm cho người dùng

TÀI LIỆU THAM KHẢO

- [1] https://www.youtube.com/watch?v=ev3-y9G8N70&list=PLhs1urmduZ2_jNSEfOMTDojkvxMjgWzmd
- [2] <https://www.youtube.com/watch?v=d1pw2FUnxgs>
- [3] <https://www.youtube.com/watch?v=99oZGQIMb0w&t=752s>