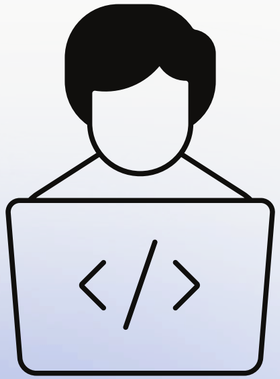
3 3
3

Learnsphere

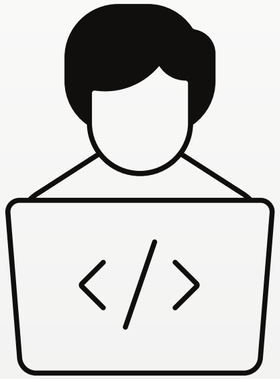
Empowering Tech Learning in Saudi Arabia

WELCOME

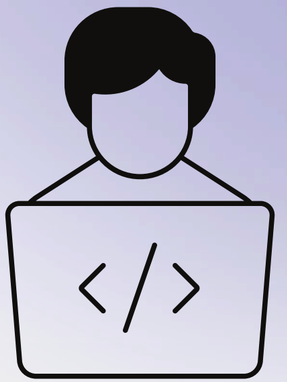
Team members :



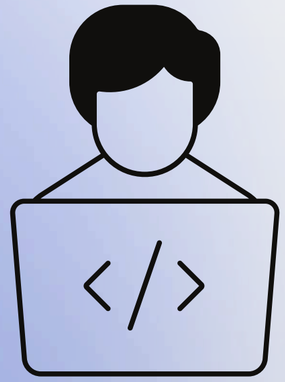
Atheer Al Otaibi
2210003305



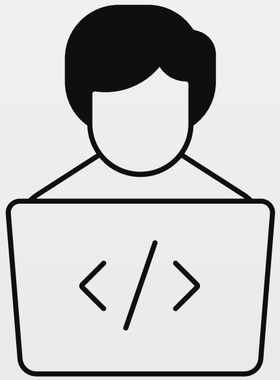
Jumana Khawaji
2200002705



Jana Albader
2210002846



Rama Alzahrani
2210002954



Ruba Alshehri
2210003377

NEXT →

Intrroduction

Under Saudi Arabia’s Vision 2030, there is a strong focus on enhancing digital skills and encouraging careers in technology through various computer science courses and training programs. While this development is promising, many beginners face challenges in choosing the right program, finding reliable resources, or staying motivated. To address these issues, our application offers certified content, personalized learning guidance, and flexible learning formats—both online and in-person. It also facilitates expert interaction, helping users gain both knowledge and confidence in navigating the tech industry.

NEXT →

Problem Statement

As Saudi Arabia embraces a tech-focused education system, many learners struggle to begin their computer science journey due to a lack of guidance in free learning materials. Our software addresses this issue by offering certified resources, structured foundational programs, and both online and in-person learning options. It also connects users with experts, helping them gain confidence and direction in their tech careers.

NEXT →

Objectives

- Design a system that is simple and intuitive with a user-friendly interface.
- Users can create, update, and delete an account.
- It provides the access to questions and answer where the users can ask questions to certain professionals.
- Admin can add, delete and update courses.
- Provide comprehensive learning resources for different fields.
- Provide online resources and in-person with their location and date.

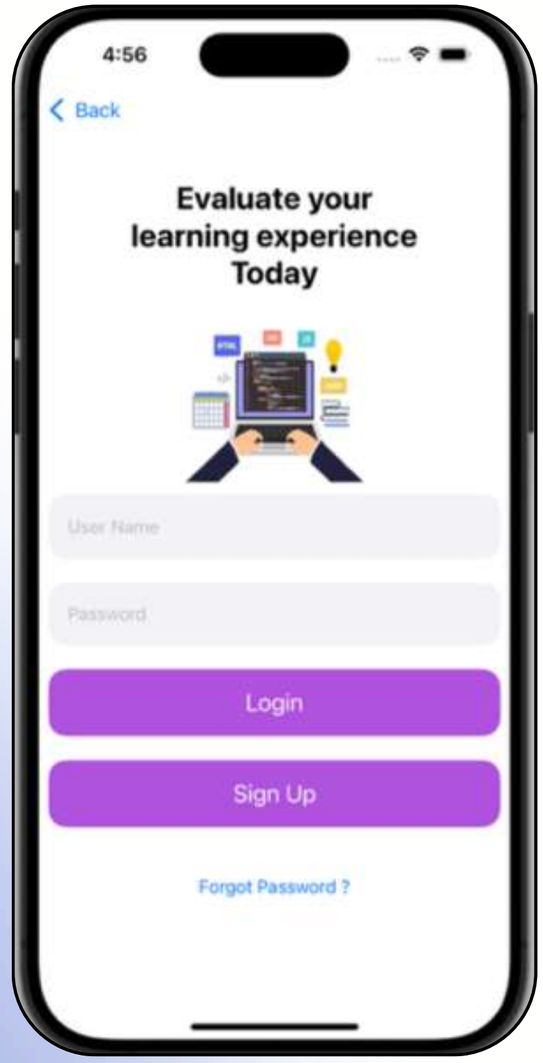
NEXT →

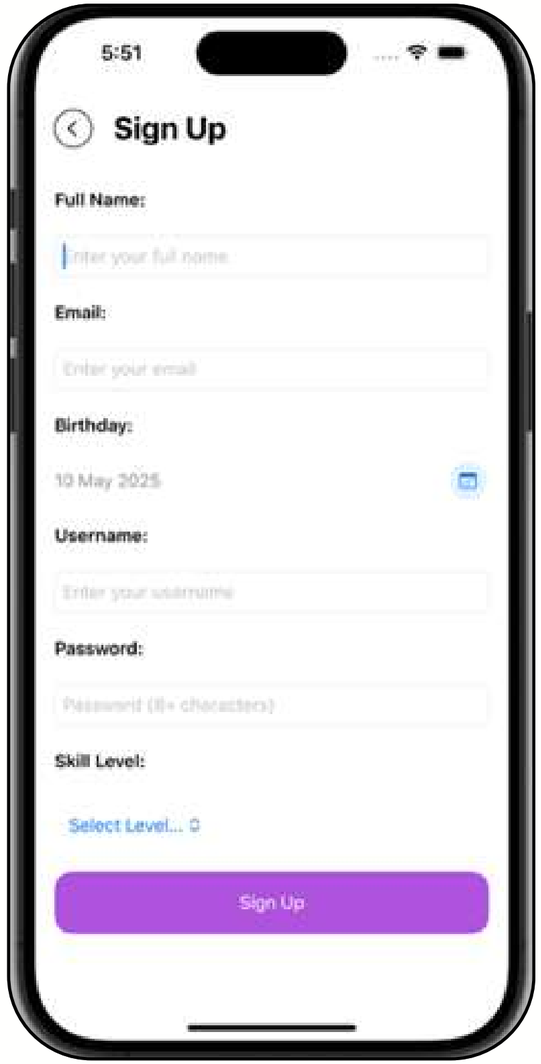
Learnsphere

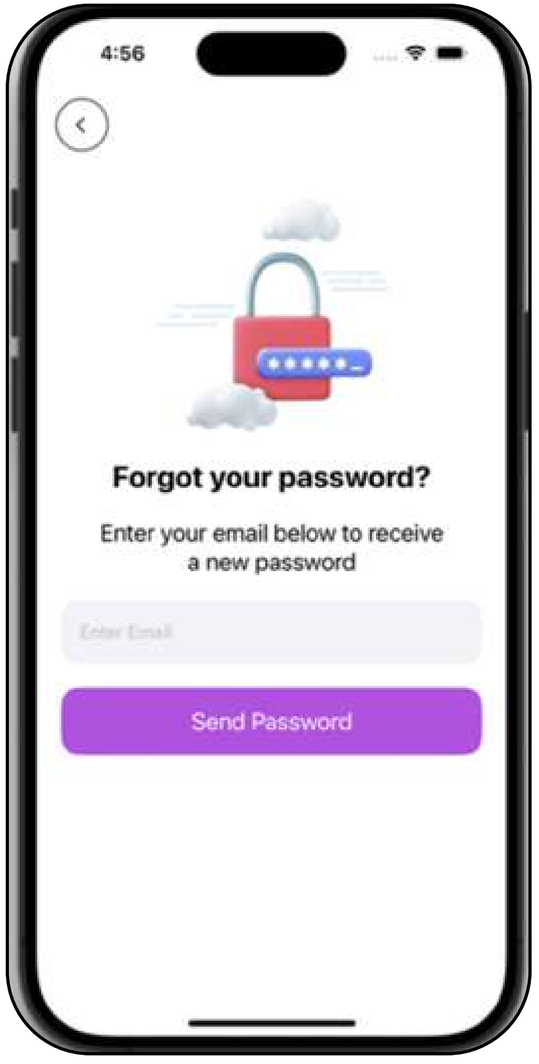
Features

User Authentication

- Login
- Sign Up
- Forgot Password







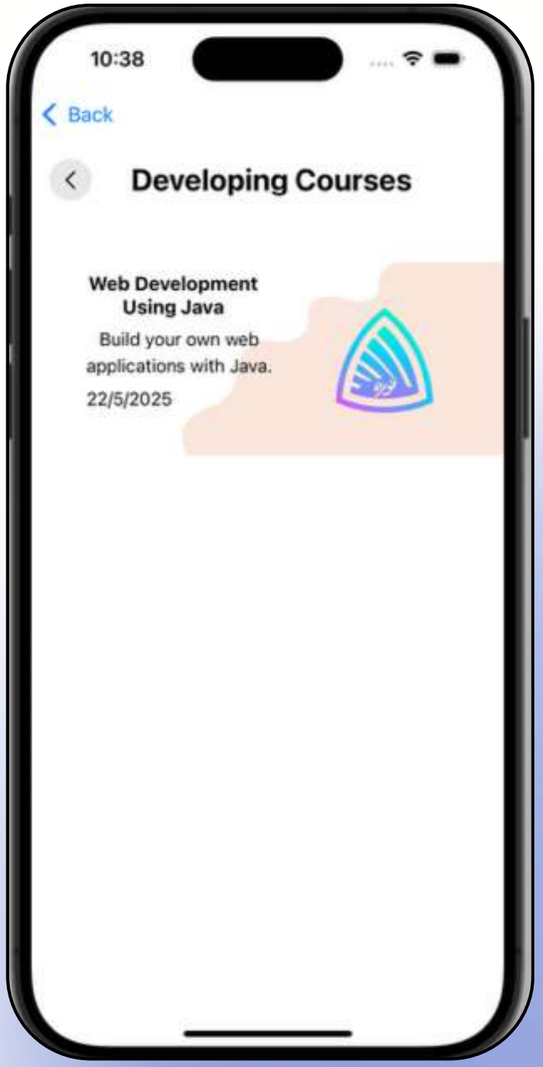
Rama Alzahrani
section: 3

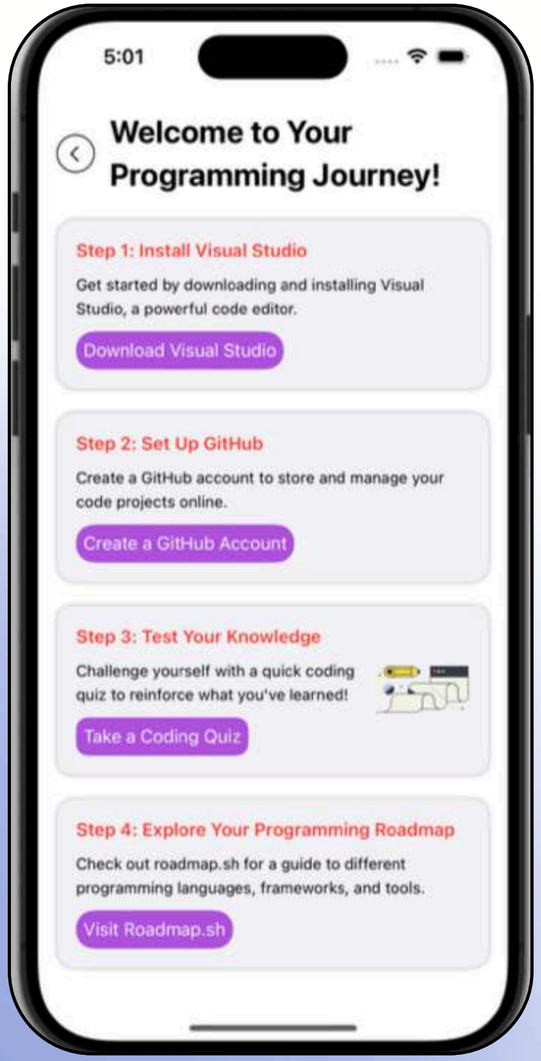
Learnsphere

Features

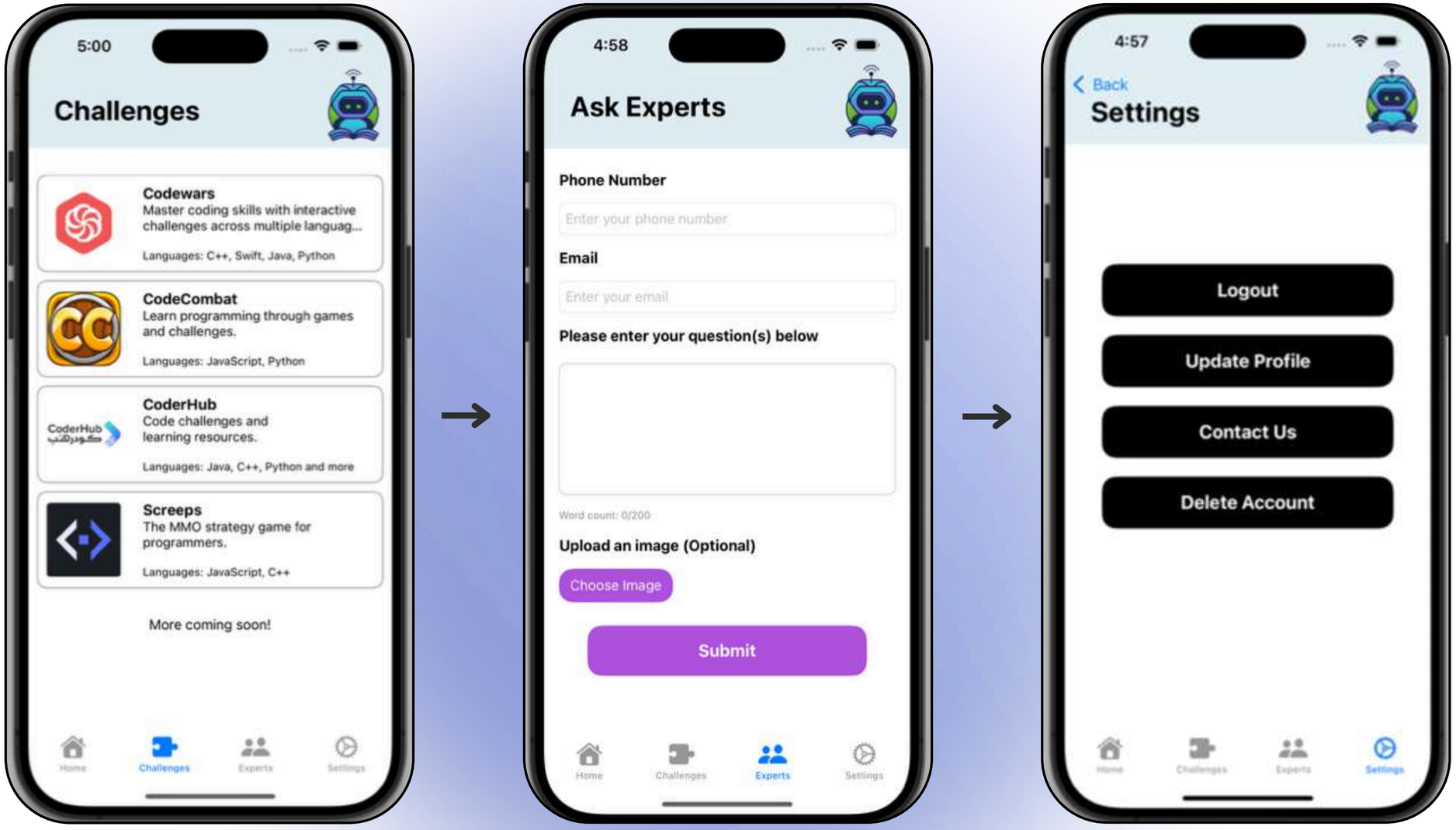
Main User Interfaces

- User Homepage
- Courses
- Start Up
- Challenges
- Ask Experts
- Settings



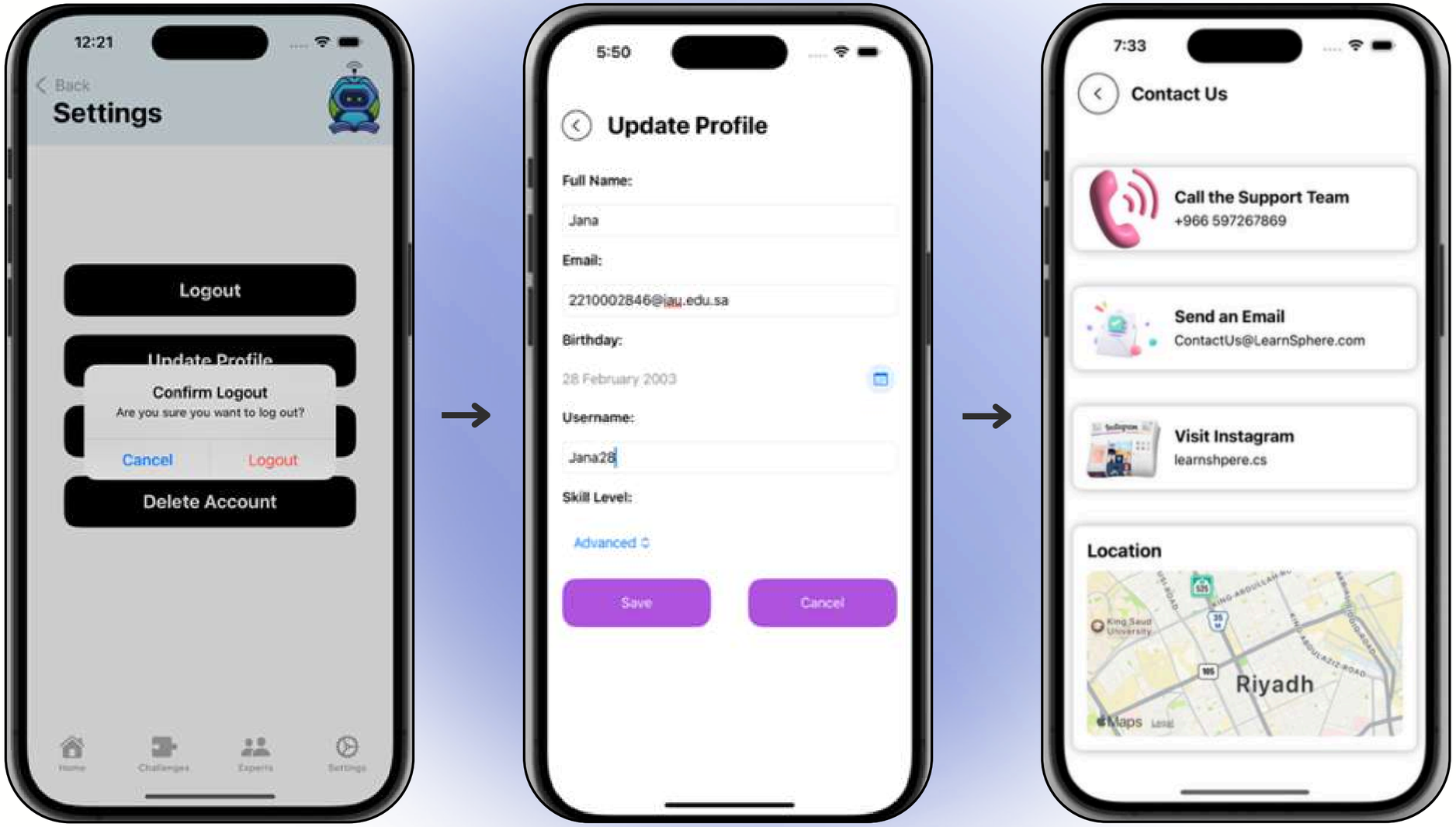


Rama Alzahrani
section: 3



User Flow

Rama Alzahrani
section: 3



User Flow

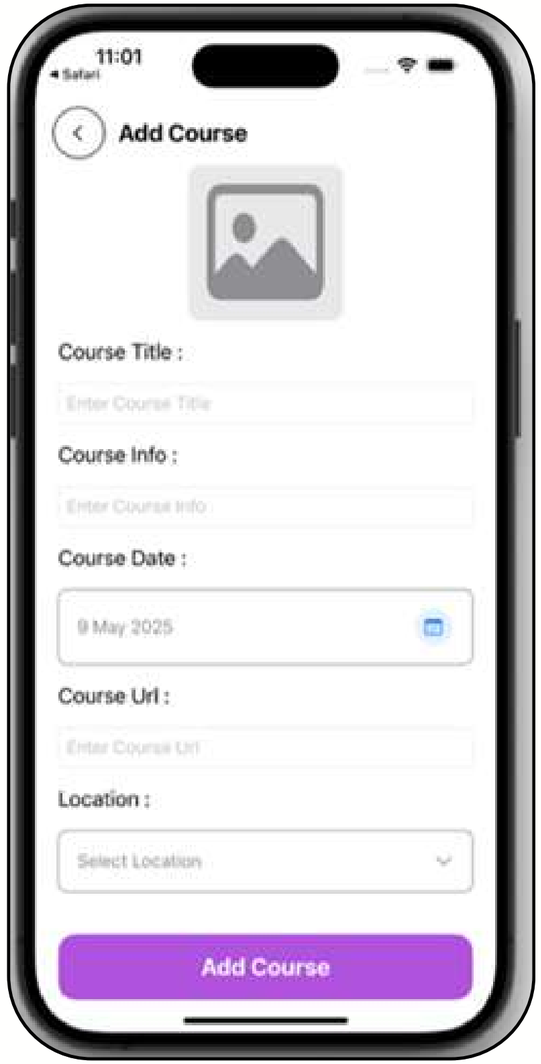
Rama Alzahrani
section: 3

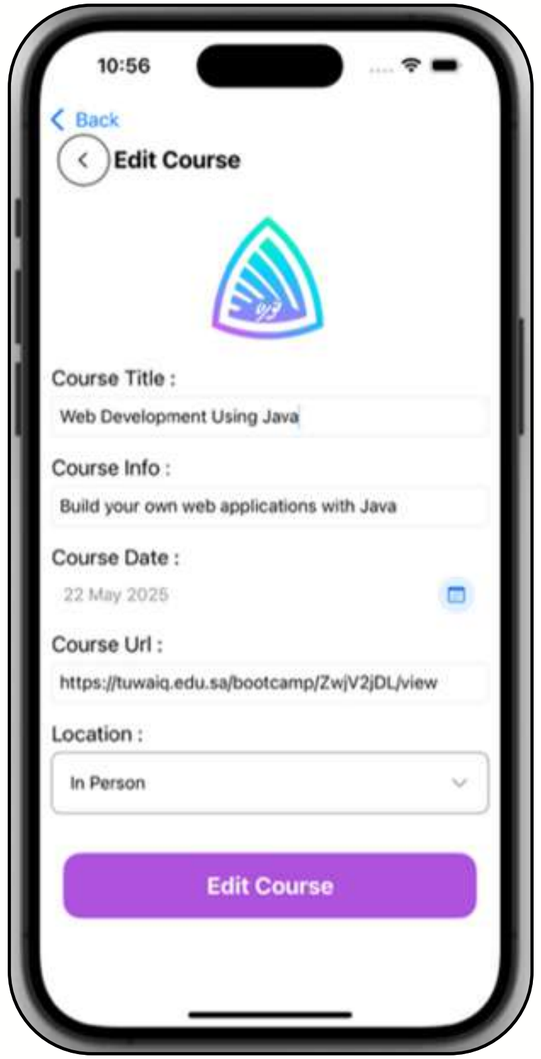
Learnsphere

Features

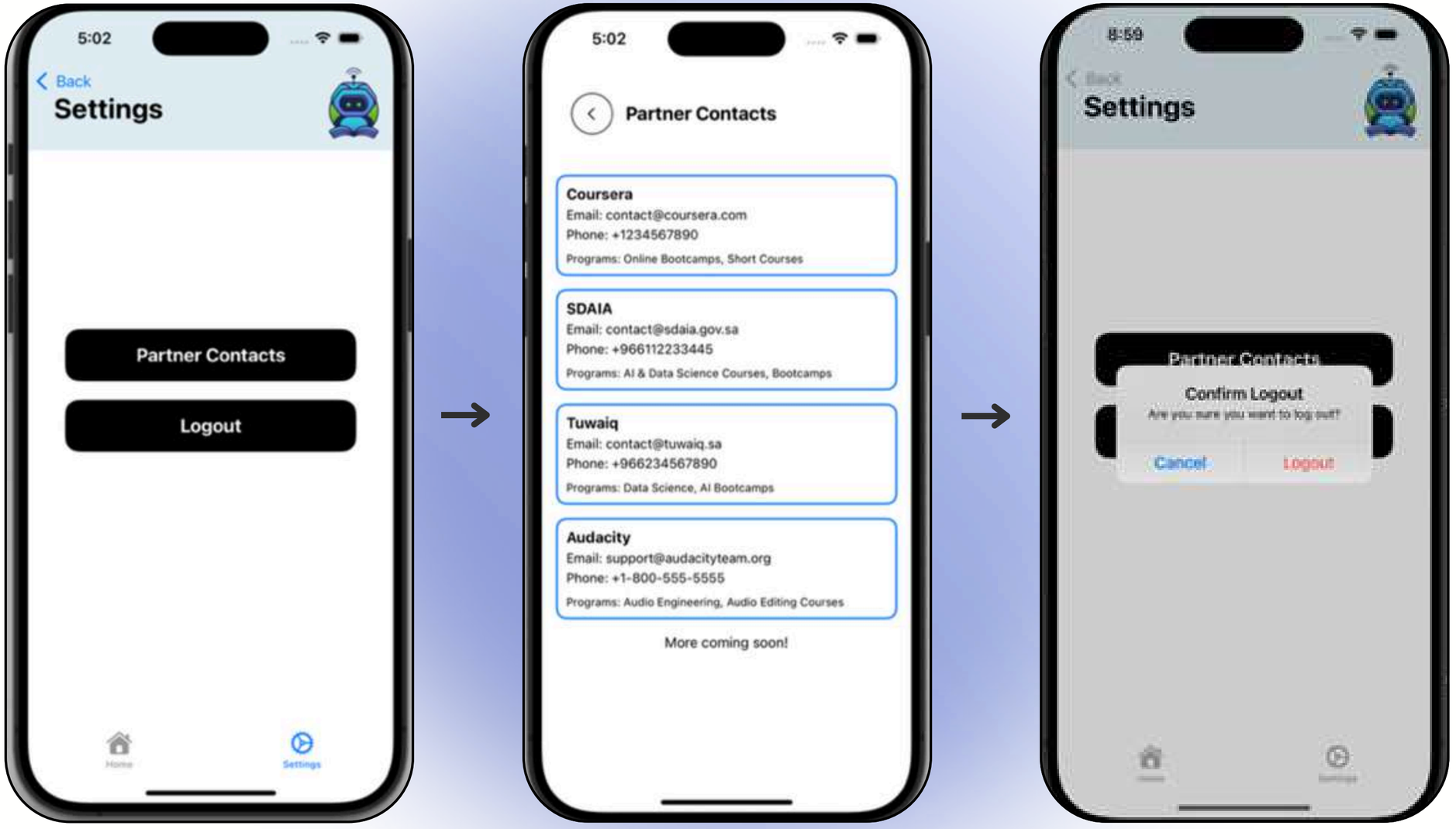
Main Admin Interfaces

- Admin Homepage
- Add course
- Update course
- Settings



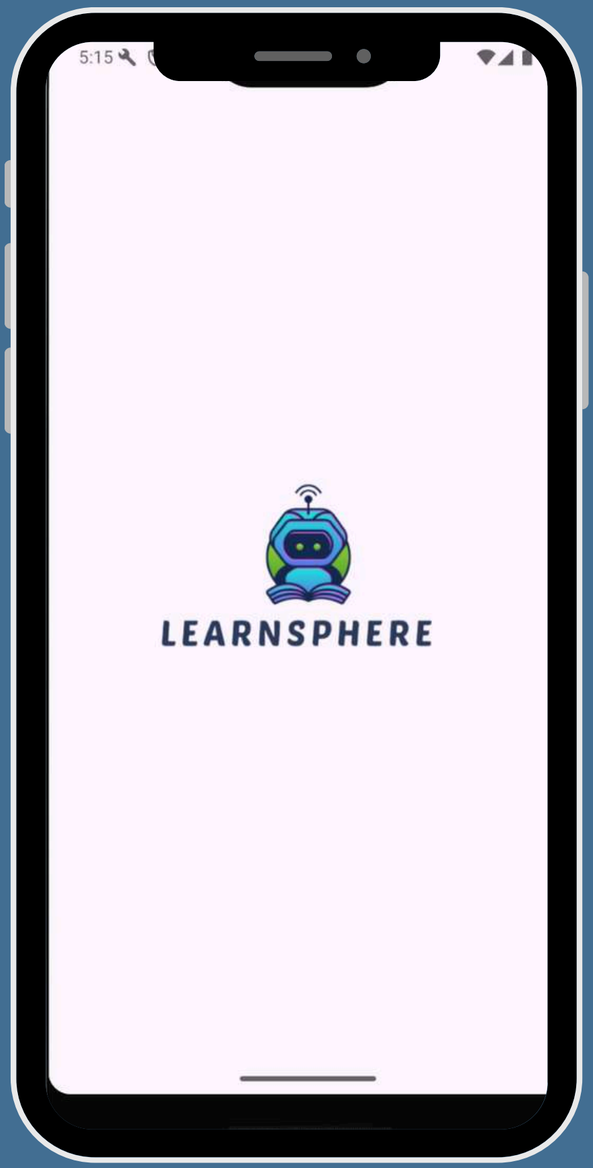


Rama Alzahrani
section: 3



User Flow

Jumana Khawaji
section: 3



LEARNSPHERE

Java Demo

Ruba Alshehri
section: 3

LEARNSPHERE

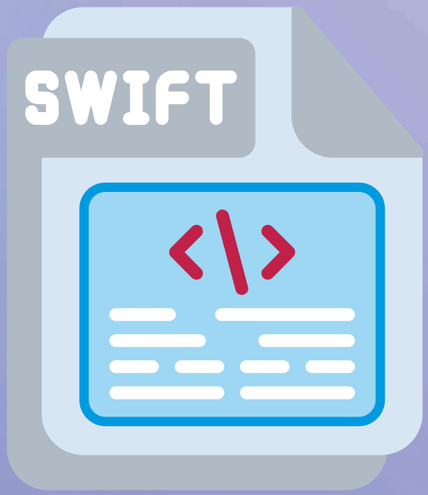
Swift Demo

Jana Albader
section: 3

JAVA

VS

SWIFT



Jana Albader
section: 3

Signup

Java

```java
public boolean insertUSER(User item, String password) {
    boolean didSucceed = false;
    try {
        ContentValues initialValues = new ContentValues();
        initialValues.put(Constants.COL_FullNAME, item.getFullName());
        initialValues.put(Constants.COL_Email, item.getEmail());
        initialValues.put(Constants.COL_Birthday, item.getBirthday());
        initialValues.put(Constants.COL_UserName, item.getUserName());
        initialValues.put(Constants.COL_PASSWORD, password);
        initialValues.put(Constants.COL_Skill, item.getSkill());
        didSucceed = database.insert(Constants.TABLE_USER, null, initialValues) > 0;

    } catch (Exception e) {
    }
    return didSucceed;
}
```

Swift

```swift
func signUp(username: String, password: String, fullName: String,
 email: String, birthday: Date, skillLevel: String) -> Bool {
    do {
        let insert = usersTable.insert(self.username <- username,
        self.password <- password, self.fullName <- fullName,
        self.email <- email, self.birthday <- birthday,
        self.skillLevel <- skillLevel)

        try db?.run(insert)
        return true
    } catch {
        print("Sign up error: \(error)")
        return false
    }
}
```

Login

Jana Albader
section: 3

Java

```java
public boolean loginUser(String username, String password) {
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.query(Constants.TABLE_USER,
    new String[]{Constants.COL_ID, Constants.COL_UserName, Constants.COL_PASSWORD},
    null, null, null, null, null, null);
    int userId = -1;
    if (cursor.moveToFirst()) {
        do {
            String _username = cursor.getString(1);
            String _password = cursor.getString(2);
            if (_username.equals(username) && _password.equals(password)) {
                userId = cursor.getInt(0);
                writeOnPreferenceId(userId);
                writeOnPreferenceUserName(username);
                cursor.close();
                break;
            }
        } while (cursor.moveToNext());
    }
    db.close();
    return userId > 0;
}
```

Swift

```swift
func login(username: String, password: String) -> Bool {
    do {
        let query = usersTable.filter(self.username == username && self.password == password)
        if let _ = try db?.pluck(query) {
            return true
        }
    } catch {
        print("Login error: \(error)")
    }
    return false
}
```

Jana Albader
section: 3

Add course

Java

```java
public boolean addCourse(Course item) {
    boolean addedSucceed = false;
    try {
        ContentValues initialValues = new ContentValues();
        initialValues.put(Course.COL_Image, item.getImgCourse());
        initialValues.put(Course.COL_TypeCourse, item.getTypeCourse());
        initialValues.put(Course.COL_CourseTitle, item.getCourseTitle());
        initialValues.put(Course.COL_CourseInfo, item.getCourseInfo());
        initialValues.put(Course.COL_CourseDate, item.getCourseDate());
        initialValues.put(Course.COL_CourseUrl, item.getCourseUrl());
        initialValues.put(Course.COL_Location, item.getLocation());
        addedSucceed = database.insert(Course.TABLE_NAME, null, initialValues) > 0;

    } catch (Exception e) {
    }
    return addedSucceed;
}
```

Swift

```swift
func insertCourse(title: String, info: String, url: String, location: String,
    date: Date, type: String, imagePath: String) {
    print("Attempting to insert course with title: \(title)")
    do {
        let formatter = DateFormatter()
        formatter.dateFormat = "yyyy-MM-dd"
        let dateString = formatter.string(from: date)

        try db?.run(coursesTable.insert(
            self.title <- title,
            self.info <- info,
            self.url <- url,
            self.location <- location,
            self.date <- dateString,
            self.type <- type,
            self.imagePath <- imagePath
        ))

        print("Course inserted successfully.")
    } catch {
        print("Insert failed: \(error)")
    }
}
```

Image picker

Jana Albader
section: 3

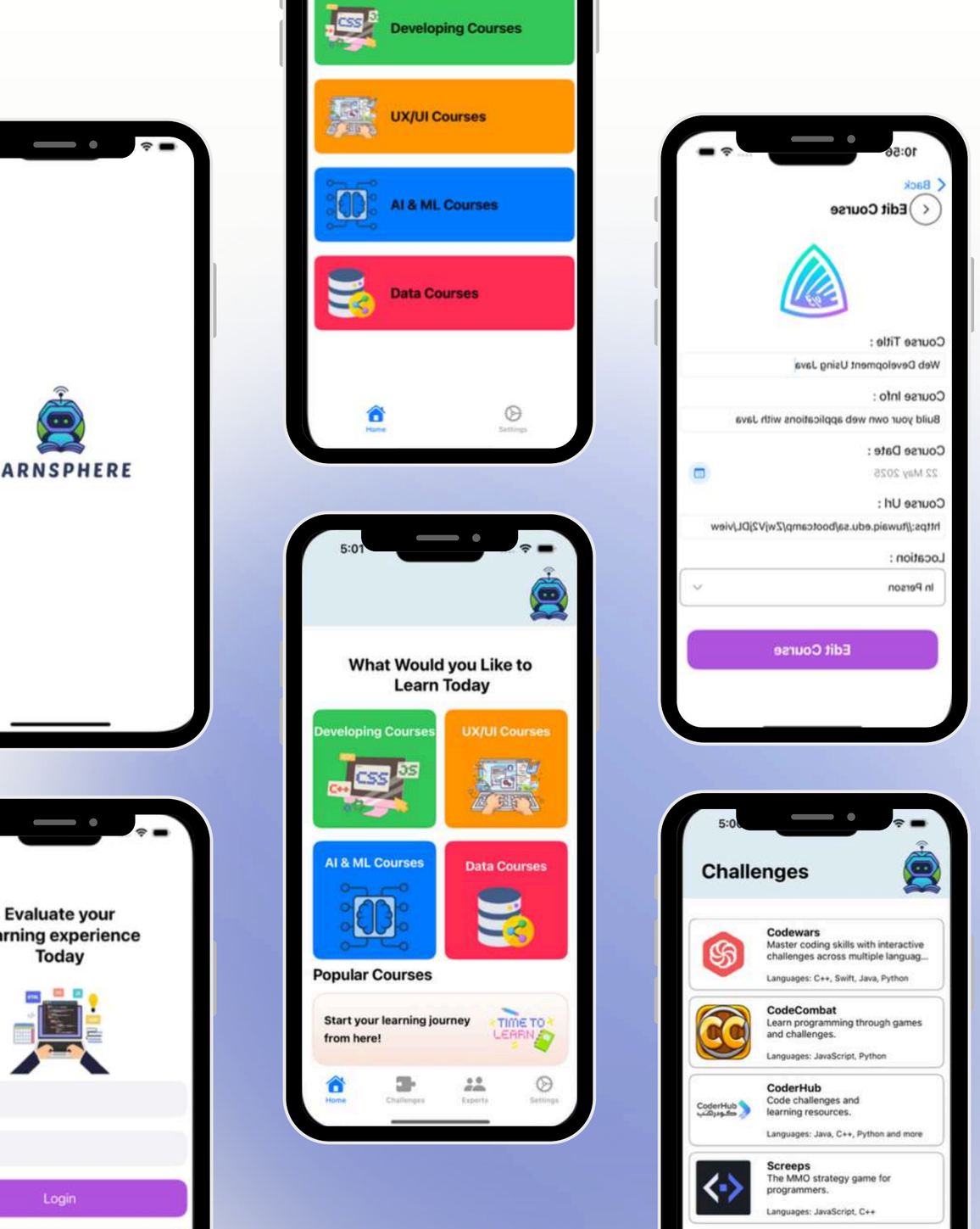
Java

```java
private ImageView uploadedImageView;

private ActivityResultLauncher<Intent> imagePickerLauncher = registerForActivityResult(
    new ActivityResultContracts.StartActivityForResult(),
    result -> {
        if (result.getResultCode() == Activity.RESULT_OK && result.getData() != null) {
            imageUri = result.getData().getData();
            uploadedImageView.setImageURI(imageUri);
            uploadedImageView.setVisibility(View.VISIBLE);}});

private void openImageChooser() {
    Intent intent = new Intent(Intent.ACTION_PICK);
    intent.setType("image/*");
    imagePickerLauncher.launch(intent);}
```

Swift

```swift
import SwiftUI
import PhotosUI

struct ImagePicker: UIViewControllerRepresentable {
    @Binding var image: UIImage?
    func makeUIViewController(context: Context) -> PHPickerViewController {
        var config = PHPickerConfiguration()
        config.filter = .images
        config.selectionLimit = 1
        let picker = PHPickerViewController(configuration: config)
        picker.delegate = context.coordinator
        return picker
    }
    func updateUIViewController(_ uiViewController: PHPickerViewController, context: Context) {
        // No updates needed
    }
    func makeCoordinator() -> Coordinator {
        Coordinator(self)
    }
    class Coordinator: NSObject, PHPickerViewControllerDelegate {
        let parent: ImagePicker
        init(_ parent: ImagePicker) {
            self.parent = parent}
        func picker(_ picker: PHPickerViewController, didFinishPicking results: [PHPickerResult]) {
            picker.dismiss(animated: true)
            guard let provider = results.first?.itemProvider,
                provider.canLoadObject(ofClass: UIImage.self) else {
                return}
            provider.loadObject(ofClass: UIImage.self) { image, error in
                if let uiImage = image as? UIImage {
                    DispatchQueue.main.async {
                        self.parent.image = uiImage
                    }}}}}
```

Jana Albader
section: 3



Conclusion