SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

MOBILE APPLICATION DEVELOPMENT

By
R.Parvathi
Assistant Professor/CSD

UNIT 3 MOBILE WEBSITES WITH ANDROID

mobile websites with android:choosing mobile wen option-adaptive mobile websites-dedicated mobile wensites-mobile web apps with HTML5-android as compepition itself-getting the tools you need-connecting to the google play- android development practices-building the derby app in android.

Choosing a Mobile Web Option

- When building for mobile, selecting the right approach is crucial. The three primary options are:

1. **Adaptive Mobile Websites**:
    - Use CSS media queries to adapt layouts for different devices.
    - Pros: Single codebase, easier maintenance.
    - Cons: Performance may suffer on older devices or slower networks.

1. **Dedicated Mobile Websites**:
    - A separate site specifically designed for mobile devices (e.g., m.example.com).
    - Pros: Optimized for mobile-specific features, faster loading.
    - Cons: Requires separate maintenance from the main desktop site.
2. **Mobile Web Apps**:
    - Built using HTML5, CSS3, and JavaScript to mimic native app behavior.
    - Pros: Cross-platform compatibility, no need for installation.
    - Cons: Limited access to device-specific features compared to native apps.

Mobile Web Apps with HTML5

- HTML5 enables powerful mobile web apps with rich functionality. Key features:
  - **Offline Support**: Using Web Storage and Service Workers for offline capabilities.
  - **Responsive Design**: Media queries and flexible grids for adaptive layouts.
  - **Device Access**: APIs like Geolocation and WebRTC for accessing hardware features.
- **Animations and Interactivity**: CSS3 and JavaScript for smooth user interfaces.

Android as Competition Itself

Android, as a platform, provides:

1. **Native Development Tools**: Android Studio and SDKs for building feature-rich native apps.
2. **Rich Ecosystem**: Millions of apps on Google Play, robust libraries, and frameworks.
3. **Market Share**: With the largest global market share, Android sets high benchmarks for competition.

►However, Android competes with:

- **Hybrid Platforms**: Tools like React Native and Flutter challenge native-only approaches.
- **Mobile Web**: Growing capabilities of mobile browsers reduce the need for native apps.

Getting the Tools You Need

- To develop for Android or mobile web:

1. **For Android Development**:
    - **Android Studio**: Official IDE for Android with built-in tools like emulators.
    - **Java or Kotlin**: Primary programming languages for Android.
    - **Android SDK**: Libraries and tools for app development.

2. For Mobile Web:

- **Code Editors**: Visual Studio Code, Sublime Text.
- **Frameworks**: Bootstrap, AngularJS, or React for front-end development.
- **Browser DevTools**: Chrome or Firefox developer tools for debugging.

Connecting to the Google Play

To publish apps on Google Play:

1. **Create a Developer Account**:
    - Register at Google Play Console.
    - Pay the one-time registration fee of $25.
2. **Prepare Your App**:
    - Generate a signed APK or App Bundle.
    - Add essential files like the AndroidManifest.xml and a keystore for signing.
3. **Upload and Submit**:
    - Complete the app listing with title, description, and screenshots.
    - Comply with Google Play's policies and submit for review.

Android Development Practices

1. **Design for Performance**:
   - Optimize app size and memory usage.
   - Use efficient layouts (e.g., ConstraintLayout).
2. **Follow Material Design Guidelines**:
   - Maintain consistency and usability with Android's design principles.

3. Test Across Devices:

- Use emulators and physical devices to test for compatibility.

4. Handle Permissions:

- Request permissions responsibly with the Android runtime permission model.

5. Ensure Security:

- Use secure APIs, encrypt sensitive data, and protect API keys.

Building the Derby App in Android

- **Step 1: Setting Up the Project**

1. Open Android Studio and create a new project.
2. Choose an empty activity and configure the project settings.

- **Step 2: Defining the UI**

- Use XML to design the interface in res/layout/activity_main.xml.

xml

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">
```

- <TextView
- android:id="@+id/welcomeText"
- android:layout_width="wrap_content"
- android:layout_height="wrap_content"
- android:text="Welcome to Derby App!"
- android:textSize="18sp" />

```
<Button
    android:id="@+id/startButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Start" />
</LinearLayout>
```

- **Step 3: Adding Functionality**

Write the app logic in MainActivity.java or MainActivity.kt.

- java
- package com.example.derbyapp;
- import android.os.Bundle;
- import android.view.View;
- import android.widget.Button;
- import android.widget.Toast;
- import androidx.appcompat.app.AppCompatActivity;

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
```

```
Button startButton = findViewById(R.id.startButton);
    startButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Toast.makeText(MainActivity.this, "Derby Started!", Toast.LENGTH_SHORT).show();
        }
    });
  }
}
```

Step 4: Running the App

1. Connect a physical device or use an emulator.
2. Build and run the app from Android Studio.

MCQ

- 1. Which mobile web option is ideal for providing a seamless user experience across all devices?
  A. Dedicated mobile website
  B. Adaptive mobile website
  C. Static desktop website
  D. Non-responsive web app
  Answer: B. Adaptive mobile website
- 2. What is the primary feature of adaptive mobile websites?
  A. Automatically resizing to fit screen sizes
  B. Providing a single layout for all devices
  C. Creating multiple versions for different devices
  D. Using fixed-width designs
  Answer: A. Automatically resizing to fit screen sizes
- 3. What is a significant drawback of dedicated mobile websites?
  A. High compatibility across devices
  B. Increased maintenance costs
  C. Seamless integration with desktop versions
  D. Faster loading speeds
  Answer: B. Increased maintenance costs

- 4. Which feature is supported by mobile web apps using HTML5?
  A. Offline storage
  B. Native device API integration
  C. Direct installation from app stores
  D. Limited browser support
  Answer: A. Offline storage
- 5. Which of the following is a reason Android is considered a strong competitor in the mobile ecosystem?
  A. Limited customization options
  B. Open-source platform
  C. Exclusive use of Google services
  D. High development costs
  Answer: B. Open-source platform
- 6. Which tool is essential for Android development?
  A. Android Studio
  B. Visual Studio Code
  C. Xcode
  D. Eclipse
  Answer: A. Android Studio

- 7. What is required to publish an app on Google Play Store?
  A. A free account
  B. A Google Play Developer account
  C. An Apple Developer account
  D. A registered trademark
  Answer: B. A Google Play Developer account
- 8. What is a recommended best practice for Android app development?
  A. Hardcoding screen sizes
  B. Testing on multiple devices
  C. Ignoring user feedback
  D. Using fixed layouts
  Answer: B. Testing on multiple devices
- 9. What is a key step in building the Derby app in Android?
  A. Designing the app interface with Xcode
  B. Testing the app with iOS emulators
  C. Using Android-specific components like RecyclerView
  D. Developing exclusively for tablets
  Answer: C. Using Android-specific components like RecyclerView

- 10. Why are adaptive mobile websites preferred by some businesses?
  A. Requires separate URLs for each device
  B. Maintains consistent performance across devices
  C. High development cost
  D. Limited user engagement
  Answer: B. Maintains consistent performance across devices