

# FinServe – React (Next.js) to Android APK using Capacitor

Step-by-step documentation for APK generation and login issue debugging.

## 1. Project Overview

Frontend: Next.js (React) with static export

Mobile: Android APK using Capacitor

Backend: ASP.NET Core API via AWS API Gateway

Hosting: AWS CloudFront

## 2. Project Structure

frontend/webapp → React + Capacitor project

frontend/webapp/out → Static build output

frontend/webapp/android → Android Studio project

backend/src/FinServe.Api → ASP.NET Core API

## 3. Next.js Configuration

Static export is mandatory for Capacitor.

next.config.js:

output: 'export'

images.unoptimized: true

## 4. Capacitor Final Working Configuration

Root cause of issue was WebView origin being https://localhost.

Final fix was loading app directly from CloudFront.

## 5. capacitor.config.ts (Final)

server.url set to CloudFront domain ensures same origin as web app.

This removes all CORS issues permanently.

## 6. Build & Sync Steps

npm run build

npx cap sync android

npx cap open android

## 7. APK Generation

Debug APK:

gradlew assembleDebug

APK Path:  
android/app/build/outputs/apk/debug/app-debug.apk

## 8. Debugging APK

Enable Developer Mode on phone  
Use adb devices to verify connection  
Use chrome://inspect to debug WebView

## 9. Root Cause & Fix

CORS blocked requests because API Gateway allowed CloudFront only.  
APK origin was localhost.  
Fix: Use CloudFront as server.url in Capacitor.

## 10. Final Result

Login works successfully in APK.  
No CORS errors.  
Production-ready and Play Store safe setup.