মোবাইল-ভিত্তিক স্বয়ংক্রিয় অ্যাপ্লিকেশন ইঞ্জিন: পিসি ছাড়া ক্লাউড এবং এআই প্রযুক্তি ব্যবহার করে অ্যান্ড্রয়েড এপিকে (APK) জেনারেশনের কৌশলগত বিশ্লেষণ

আধুনিক সফটওয়্যার ইঞ্জিনিয়ারিং এবং ক্লাউড কম্পিউটিংয়ের অভাবনীয় অগ্রগতি এমন এক নতুন দিগন্তের উন্মোচন করেছে, যেখানে একটি পূর্ণাঙ্গ অ্যাপ্লিকেশন ডেভেলপমেন্ট সাইকেল পরিচালনার জন্য প্রথাগত কম্পিউটার বা ল্যাপটপের প্রয়োজনীয়তা ক্রমশ হ্রাস পাচ্ছে। কৃত্রিম বুদ্ধিমত্তা (AI) চালিত কোড জেনারেটর এবং ক্লাউড-নেটিভ কম্পাইলেশন টুলসের সমন্বয়ে বর্তমানে একটি সাধারণ অ্যান্ড্রয়েড স্মার্টফোন ব্যবহার করেই অ্যাপ্লিকেশন ডিজাইন, কোড লিখন এবং তা এক্সিকিউটেবল অ্যান্ড্রয়েড প্যাকেজ কিট (APK) ফাইলে রূপান্তর করা সম্ভব।
এই গবেষণা প্রতিবেদনে একটি স্বয়ংক্রিয় 'অ্যাপ্লিকেশন ইঞ্জিন' তৈরির কৌশলগত আর্কিটেকচার বিশ্লেষণ করা হয়েছে। এই ইঞ্জিনের মূল উদ্দেশ্য হলো—ব্যবহারকারীর সরবরাহকৃত ওয়েব অ্যাপ্লিকেশনের জিপ (ZIP) ফাইল অথবা কোনো অ্যাপ্লিকেশনের ব্লুপ্রিন্ট বা আইডিয়া গ্রহণ করে, এআই ডেভেলপারের মাধ্যমে স্বয়ংক্রিয়ভাবে কোড জেনারেট করা এবং চূড়ান্ত পর্যায়ে সেটিকে সরাসরি অ্যান্ড্রয়েডে ইনস্টলযোগ্য APK ফাইলে রূপান্তর করা। সম্পূর্ণ প্রক্রিয়াটি পিসি ছাড়া শুধুমাত্র অ্যান্ড্রয়েড ডিভাইসে সম্পন্ন করার সম্ভাব্যতা, গুগল এআই স্টুডিও-এর সক্ষমতা, প্রয়োজনীয় থার্ড-পার্টি ক্লাউড টুলস, পাইপলাইন আর্কিটেকচার এবং এটি বাস্তবায়নের জন্য ব্যবহারকারীর করণীয় সম্পর্কে এই প্রতিবেদনে বিস্তারিত আলোচনা করা হয়েছে।

গুগল এআই স্টুডিও: সক্ষমতা, ফ্রি টিয়ার এবং বাংলাদেশ প্রেক্ষিত

গুগল এআই স্টুডিও (Google AI Studio) বর্তমানে ডেভেলপার এবং নন-ডেভেলপারদের জন্য অন্যতম শক্তিশালী একটি প্রোটোটাইপিং প্ল্যাটফর্ম হিসেবে আবির্ভূত হয়েছে। এর অন্তর্নিহিত জেমিনি (Gemini) মডেলগুলো ব্যবহার করে অত্যন্ত দ্রুততার সাথে আইডিয়া থেকে অ্যাপ্লিকেশনে রূপান্তর ঘটানো সম্ভব।

বিলিং একাউন্ট ব্যতীত ফ্রি টিয়ারের আওতা এবং প্রাপ্যতা

গবেষণা ও বিশ্লেষণে দেখা যায়, গুগল এআই স্টুডিও ব্যবহার করার জন্য কোনো সাবস্ক্রিপশন, সিট-ভিত্তিক ফি বা প্রাথমিক পে-ওয়ালের (Paywall) সম্মুখীন হতে হয় না। বাংলাদেশ থেকে এই প্ল্যাটফর্মটি ব্যবহার করতে কোনো ক্রেডিট কার্ড, ডুয়াল-কারেন্সি কার্ড বা গুগল ক্লাউড বিলিং একাউন্ট অ্যাক্টিভেশনের প্রয়োজন নেই। বাংলাদেশ আনুষ্ঠানিকভাবে গুগল এআই স্টুডিও এবং জেমিনি এপিআই-এর সমর্থিত অঞ্চলের (Supported Regions) তালিকাভুক্ত হওয়ায় কোনো ভিপিএন (VPN) বা জিও-লক বাইপাস কৌশল ছাড়াই এটি সরাসরি ব্যবহারযোগ্য।
ফ্রি টিয়ারের আওতায় একজন ব্যবহারকারী জেমিনি ১.৫ ফ্ল্যাশ (Gemini 1.5 Flash) এবং জেমিনি ১.৫ প্রো (Gemini 1.5 Pro) মডেলগুলো ব্যবহার করে প্রতিদিন উল্লেখযোগ্য সংখ্যক রিকোয়েস্ট (RPD) এবং টোকেন (TPM) সম্পূর্ণ বিনামূল্যে লাভ করেন। উদাহরণস্বরূপ, প্রো মডেলের ক্ষেত্রে প্রতিদিন প্রায় ১০০টি এবং ফ্ল্যাশ মডেলের ক্ষেত্রে ২৫০টি রিকোয়েস্ট বিনামূল্যে পাওয়া যায়, যা ব্যক্তিগত প্রোটোটাইপিং এবং স্বয়ংক্রিয় ইঞ্জিন পরিচালনার জন্য যথেষ্ট। তবে এই ফ্রি টিয়ারের একটি গুরুত্বপূর্ণ শর্ত হলো, ব্যবহারকারীর ইনপুট এবং জেনারেট করা ডেটা গুগলের পণ্য উন্নয়নে ব্যবহৃত হতে পারে; ডেটা প্রাইভেসি নিশ্চিত করতে চাইলে পেইড এপিআই টিয়ারে আপগ্রেড করার প্রয়োজন হয়।

'Antigravity Agent' এবং স্বয়ংক্রিয় অ্যান্ড্রয়েড অ্যাপ জেনারেশন

গুগল আই/ও (I/O) ২০২৪ ইভেন্টে এআই স্টুডিওতে একটি যুগান্তকারী ফিচার যুক্ত করা হয়েছে, যার মাধ্যমে সরাসরি টেক্সট প্রম্পট থেকে নেটিভ অ্যান্ড্রয়েড অ্যাপ তৈরি করা সম্ভব। এই ফিচারটি 'Antigravity Agent' এর ওপর ভিত্তি করে

কাজ করে। এর প্রযুক্তিগত বৈশিষ্ট্যগুলো নিম্নরূপ:

- নেটিভ কোড আউটপুট: বাজারে প্রচলিত অন্যান্য অনেক এআই বিল্ডার যেখানে ওয়েব-র‍্যাপার বা 'React Native' ব্যবহার করে, সেখানে গুগল এআই স্টুডিও সরাসরি 'Kotlin' প্রোগ্রামিং ভাষা এবং 'Jetpack Compose' ইউআই ফ্রেমওয়ার্ক ব্যবহার করে প্রোডাকশন-রেডি নেটিভ কোড তৈরি করে।
- ইন-ব্রাউজার এমুলেটর এবং ক্লাউড রান: মোবাইল ব্রাউজারেই একটি এমুলেটর রান করে অ্যাপটির প্রিভিউ দেখা যায়। ব্যবহারকারীরা ব্রাউজার থেকেই হার্ডওয়্যার সেন্সর যেমন—জিপিএস, ব্লুটুথ এবং অ্যাক্সিলেরোমিটার এক্সেস করতে পারেন।
- সরাসরি ডিভাইস ইন্সটলেশন: ওয়েবইউএসবি (WebUSB) এবং অ্যান্ড্রয়েড ডিবাগ ব্রিজ (ADB) প্রোটোকল ব্যবহার করে ব্রাউজার থেকে সরাসরি ডেটা ক্যাবল দিয়ে যুক্ত ফোনে অ্যাপ ইন্সটল করা যায়।
- **APK** এক্সপোর্ট এবং জিপ ডাউনলোড: এআই স্টুডিও স্বয়ংক্রিয়ভাবে অ্যাপের সোর্স কোড এবং মেটাডেটা একটি জিপ (ZIP) ফাইল হিসেবে ডাউনলোড করার সুযোগ দেয়।

এআই স্টুডিওর সীমাবদ্ধতা এবং **"Parsing the Package"** ত্রুটি

যদিও এআই স্টুডিও জিপ ফাইল এক্সপোর্ট করে এবং অনেক ক্ষেত্রে জিপ ফাইলের build/outputs/apk/debug/ ডিরেক্টরিতে একটি প্রাক-কম্পাইলড app-debug.apk ফাইল পাওয়া যায়, তথাপি এটি সরাসরি মোবাইলে ইনস্টল করার ক্ষেত্রে ব্যাপক সীমাবদ্ধতা রয়েছে।
সবচেয়ে সাধারণ সমস্যাটি হলো "There was a problem parsing the package" (পার্সিং এরর)। এই ত্রুটিটি সাধারণত ঘটে যখন:

1. APK ফাইলটি আংশিকভাবে করাপ্টেড থাকে।
2. ডিভাইসের সিপিইউ আর্কিটেকচারের (যেমন: armeabi-v7a বনাম arm64-v8a) সাথে কম্পাইল করা বিল্ডের অমিল থাকে।
3. অ্যান্ড্রয়েডের মিনিমাম এপিআই লেভেল (Minimum SDK) মিসম্যাচ হয় অথবা ফাইলটিতে সঠিক ক্রিপ্টোগ্রাফিক সিগনেচার (Keystore signature) থাকে না।

বিশ্লেষণমূলক অন্তর্দৃষ্টি: গুগল এআই স্টুডিও কি এমন কোনো অ্যাপ তৈরি করতে পারবে যা নিজে একটি APK জেনারেটর হিসেবে কাজ করবে? প্রযুক্তিগতভাবে এর উত্তর হলো 'না'। এআই স্টুডিও বর্তমানে শুধুমাত্র ক্লায়েন্ট-সাইড (Client-side) অ্যাপ তৈরি করতে পারে। একটি APK কম্পাইলার তৈরি করতে হলে সার্ভার-সাইড প্রসেসিং, জাভা ডেভেলপমেন্ট কিট (JDK), ডালভিক বাইটকোড কম্পাইলার (DEX), এবং গ্র্যাডল (Gradle) বিল্ড সিস্টেমের প্রয়োজন হয়, যা এআই স্টুডিওর বর্তমান আর্কিটেকচারে ক্লায়েন্ট-সাইড অ্যাপের ভেতরে এম্বেড করা সম্ভব নয়।
অতএব, একটি সম্পূর্ণ স্বয়ংক্রিয় এবং ত্রুটিমুক্ত অ্যাপ্লিকেশন ইঞ্জিন তৈরি করতে হলে গুগল এআই স্টুডিওর কোড জেনারেটর সক্ষমতাকে একটি বাহ্যিক ক্লাউড কম্পাইলার বা অটোমেটেড পাইপলাইনের (যেমন গিটহাব অ্যাকশনস) সাথে যুক্ত করা অপরিহার্য।

পিসি ছাড়া ওয়েব জিপ থেকে এপিকে **(APK)** কম্পাইল করার ক্লাউড টুলস

যদি এআই স্টুডিও বা অন্য কোনো কোড জেনারেটর একটি ওয়েব অ্যাপ্লিকেশনের জিপ ফাইল (HTML, CSS, JS) তৈরি করে দেয়, তবে সেটিকে পিসি ছাড়া মোবাইলেই নেটিভ অ্যান্ড্রয়েড অ্যাপ বা APK-তে রূপান্তর করার জন্য বেশ কয়েকটি নির্ভরযোগ্য ক্লাউড-ভিত্তিক প্ল্যাটফর্ম রয়েছে। নিচে এই টুলসগুলোর একটি তুলনামূলক বিশ্লেষণ প্রদান করা হলো।

প্ল্যাটফর্মের নাম	প্রযুক্তিগত কৌশল ও আউটপুট	ফ্রি টিয়ারের সুবিধা ও সীমাবদ্ধতা	ইঞ্জিন ইন্টিগ্রেশনের উপযোগিতা
html2app.dev	HTML5 জিপ থেকে সরাসরি Native APK/AAB তৈরি করে। কোনো Android Studio প্রয়োজন হয় না।	প্রতিদিন ৫টি বিল্ড ফ্রি, তবে অ্যাপের সাইজ লিমিট ১০ মেগাবাইট এবং অ্যাপে ওয়াটারমার্ক যুক্ত থাকে।	অফলাইন-রেডি ওয়েব অ্যাপ্লিকেশনের জন্য অত্যন্ত কার্যকর, ব্রাউজার থেকেই জিপ আপলোড করা যায়।

প্ল্যাটফর্মের নাম	প্রযুক্তিগত কৌশল ও আউটপুট	ফ্রি টিয়ারের সুবিধা ও সীমাবদ্ধতা	ইঞ্জিন ইন্টিগ্রেশনের উপযোগিতা
Newly.app	React Native/Expo সোর্স কোড গ্রহণ করে ক্লাউড সার্ভারে কম্পাইল করে সাইন করা APK প্রদান করে।	প্রথম বিল্ড ফ্রি, কিন্তু নিয়মিত ব্যবহারের জন্য মাসিক ২৫ ডলার সাবস্ক্রিপশন প্রয়োজন।	সম্পূর্ণ স্বয়ংক্রিয়, কিন্তু রিঅ্যাক্ট নেটিভ কোডবেস প্রয়োজন এবং পেইড হওয়ার কারণে ফ্রি ইঞ্জিনের জন্য অনুপযুক্ত।
AppMint / Web2Apk	ওয়েবসাইট ইউআরএল (URL) বা আপলোড করা HTML থেকে Android WebView র‍্যাপার তৈরি করে।	সম্পূর্ণ ফ্রি, আনলিমিটেড অ্যাপ তৈরি সম্ভব এবং AdMob ইন্টিগ্রেশনের সুবিধা রয়েছে।	নেটিভ কোডের পারফরম্যান্স পাওয়া যায় না, তবে সাধারণ ব্রোশিওর বা টুলস অ্যাপের জন্য দ্রুততম সমাধান।
Appilix	ওয়েব URL থেকে স্বয়ংক্রিয়ভাবে Native iOS এবং Android ফাইল জেনারেট করে।	১৫ দিনের ট্রায়াল প্রদান করে, যেখানে পুশ নোটিফিকেশন সুবিধা অন্তর্ভুক্ত।	কোড লেভেলে কাস্টমাইজেশন সম্ভব নয় এবং ট্রায়াল শেষে প্রিমিয়াম প্ল্যান বেশ ব্যয়বহুল।
RapidNative	এআই প্রম্পট থেকে সরাসরি React Native কোড তৈরি করে এবং Expo সার্ভারে বিল্ড করে।	ফ্রি টিয়ার উপলব্ধ, সরাসরি Expo Go অ্যাপের মাধ্যমে মোবাইলে লাইভ প্রিভিউ দেখা যায়।	সরাসরি APK ডাউনলোডের চেয়ে Expo ক্লাউড বিল্ডের (EAS) ওপর বেশি নির্ভরশীল।

এই বিশ্লেষণ থেকে এটি স্পষ্ট যে, একটি ওয়েব-ভিত্তিক অ্যাপ্লিকেশনের জিপ ফাইল থাকলে **html2app.dev** অথবা **AppMint** সবচেয়ে দ্রুত এবং মোবাইল-বান্ধব সমাধান প্রদান করে। এই প্ল্যাটফর্মগুলোতে সরাসরি মোবাইল ব্রাউজার থেকে জিপ ফাইল আপলোড করে কয়েক মিনিটের মধ্যে ইনস্টলযোগ্য APK ডাউনলোড করা সম্ভব, যা পিসি-বিহীন আর্কিটেকচারের জন্য আদর্শ।

আইডিয়া থেকে কোড জেনারেট করার মোবাইল-ফ্রেন্ডলি এআই প্ল্যাটফর্ম

গুগল এআই স্টুডিও ছাড়াও আরও বেশ কিছু প্ল্যাটফর্ম রয়েছে যা মোবাইলের ব্রাউজার থেকে সরাসরি আইডিয়া বা ব্লুপ্রিন্ট গ্রহণ করে সোর্স কোড জেনারেট করতে সক্ষম। এই প্ল্যাটফর্মগুলো ইঞ্জিনের 'ব্রেইন' বা লজিক জেনারেটর হিসেবে কাজ করতে পারে।

এআই প্ল্যাটফর্ম	মূল ফোকাস এবং আউটপুট	মোবাইল-বান্ধব বৈশিষ্ট্য
Bolt.new (StackBlitz)	প্রম্পট থেকে অত্যন্ত দ্রুতগতিতে React, Vue বা React Native কোড তৈরি করে।	ব্রাউজার-ভিত্তিক লাইভ প্রিভিউ এবং সম্পূর্ণ সোর্স কোড রিপোজিটরি গিটহাবে এক্সপোর্টের সুবিধা দেয়।
Lovable.dev	ফুল-স্ট্যাক ওয়েব অ্যাপ তৈরি করে, যার সাথে Supabase ডাটাবেস ইন্টিগ্রেটেড থাকে।	নন-ডেভেলপারদের জন্য আদর্শ, মোবাইল ব্রাউজারে ইউআই এডিট করা সহজ।
FlutterFlow AI	প্রম্পট থেকে ভিজ্যুয়াল ইউজার ইন্টারফেস এবং ডার্ট (Dart) কোড জেনারেট করে।	মোবাইল ব্রাউজার থেকে ভিজ্যুয়াল বিল্ডারে কাজ করা যায় এবং সরাসরি এপিকে বা আইপিএ (IPA) এক্সপোর্ট করা যায়।
Manus AI	স্বয়ংক্রিয় এআই এজেন্ট যা ফ্রন্টএন্ড, ব্যাকএন্ড এবং ডাটাবেস সম্পূর্ণ নিজে তৈরি করে।	সরাসরি মোবাইল ডিভাইসে প্রিভিউ এবং ইন্সটল করার সুযোগ প্রদান করে।
Replit Agent	ক্লাউড হোস্টিং সহ ফুল-স্ট্যাক ডেভেলপমেন্ট এনভায়রনমেন্ট প্রদান করে।	মোবাইল থেকে ব্রাউজারে কোড এডিট করা এবং সরাসরি এপিআই ডিপ্লয় করা সম্ভব।

**অন্তর্নিহিত সম্পর্ক বিশ্লেষণ: এই প্ল্যাটফর্মগুলো সরাসরি কোড তৈরি করলেও, ফিজিক্যাল ডিভাইসে রান করার জন্য এপিকে ফাইলে রূপান্তর করতে একটি কম্পাইলার প্রয়োজন। Bolt বা Lovable এর মতো ওয়েব-ফার্স্ট প্ল্যাটফর্ম থেকে জেনারেট করা কোডকে VibeDeploy-এর মতো সেবার মাধ্যমে লাইভ করা যায়। তবে অ্যান্ড্রয়েড এপিকে পাওয়ার জন্য এই সোর্স কোডগুলোকে গিটহাব অ্যাকশনস বা Termux এর মতো কম্পাইলেশন পাইপলাইনের মধ্য দিয়ে নিয়ে যেতে হবে।

এআই জেনারেটেড কোড এবং এপিকে কম্পাইলারের মধ্যে পাইপলাইন স্থাপনের কৌশল

ব্যবহারকারীর কাঙ্ক্ষিত 'অ্যাপ্লিকেশন ইঞ্জিন' বাস্তবে রূপ দেওয়ার জন্য এআই (যে লজিক ও কোড লিখবে) এবং কম্পাইলার (যে কোডকে APK বানাবে)-এর মধ্যে একটি স্বয়ংক্রিয়, নিরবচ্ছিন্ন সংযোগ স্থাপন করতে হবে। যেহেতু পিসি ব্যবহার করা হবে না, তাই এই পাইপলাইনটি সম্পূর্ণ ক্লাউড-নেটিভ (Cloud-native) বা অন-ডিভাইস (On-device) হতে হবে। এটি বাস্তবায়নের জন্য দুটি প্রধান আর্কিটেকচারাল কৌশল রয়েছে।

কৌশল ১: গিটহাব অ্যাকশনস এবং টেলিগ্রাম বট ভিত্তিক ক্লাউড পাইপলাইন (সর্বোচ্চ স্বয়ংক্রিয়)

এটি সবচেয়ে স্বয়ংক্রিয়, নির্ভরযোগ্য এবং পিসি-মুক্ত কৌশল। এটি মোবাইল ব্রাউজার থেকেই সম্পূর্ণ নিয়ন্ত্রণ করা যায় এবং কম্পাইলেশনের জন্য মোবাইলের হার্ডওয়্যারের ওপর কোনো চাপ ফেলে না।

1. কোড জেনারেশন এবং পুশ: গুগল এআই স্টুডিও-তে প্রম্পট দিয়ে অ্যাপের কোড তৈরি করা হয়। এআই স্টুডিও থেকে কোডটি জিপ হিসেবে ডাউনলোড না করে সরাসরি গিটহাব (GitHub) রিপোজিটরিতে এক্সপোর্ট করা হয় (অথবা ব্রাউজার থেকে জিপ ফাইলটি গিটহাবে আপলোড করা হয়)।
2. ক্লাউড কম্পাইলেশন ট্রিগার: গিটহাব রিপোজিটরিতে আগে থেকেই একটি .github/workflows/android.yml ফাইল কনফিগার করা থাকে। যখনই এআই স্টুডিও বা ব্যবহারকারী নতুন কোড গিটহাবে পুশ করে, গিটহাব অ্যাকশনস (GitHub Actions) এর উবুন্টু (Ubuntu) সার্ভার স্বয়ংক্রিয়ভাবে রান করে। এটি ক্লাউডে জাভা ডেভেলপমেন্ট কিট (JDK) সেটআপ করে, Gradle ব্যবহার করে কোডটিকে কম্পাইল করে এবং একটি সাইন করা (Signed) রিলিজ বা ডিবাগ APK তৈরি করে।
3. অটোমেটেড ডেলিভারি: গিটহাব অ্যাকশনসের পাইপলাইনের শেষ ধাপে একটি টেলিগ্রাম বট এপিআই (Telegram Bot API) বা কার্ল (cURL) কমান্ড সংযুক্ত থাকে। সার্ভারে কম্পাইল হওয়া APK ফাইলটি অ্যাকশনস সরাসরি ব্যবহারকারীর টেলিগ্রাম আইডিতে একটি মেসেজ হিসেবে পাঠিয়ে দেয়।

এই পাইপলাইনের ফলে ব্যবহারকারীকে কোড, কমান্ড লাইন বা ফাইল ম্যানেজমেন্ট নিয়ে কাজ করতে হয় না। শুধু এআই স্টুডিওতে প্রম্পট দিলে কয়েক মিনিট পর টেলিগ্রামে সরাসরি ইনস্টলযোগ্য APK চলে আসে। এটি পার্সিং এরর সমাধান করে, কারণ গিটহাবের সার্ভার সঠিক আর্কিটেকচার এবং সিগনেচার ব্যবহার করে স্ট্যান্ডার্ড এপিকে তৈরি করে।

কৌশল ২: **Termux** ব্যবহার করে সম্পূর্ণ অন-ডিভাইস **(On-Device)** ইঞ্জিন

যদি ক্লাউড সার্ভারের ওপর নির্ভরতা শূন্যে নামিয়ে আনতে হয় এবং অফলাইনে এপিকে তৈরি করতে হয়, তবে অ্যান্ড্রয়েডের শক্তিশালী টার্মিনাল এমুলেটর **Termux** ব্যবহার করা যায়।

1. এনভায়রনমেন্ট সেটআপ: Termux-এ প্যাকেজ ম্যানেজারের মাধ্যমে OpenJDK 21, Android SDK কমান্ড-লাইন টুলস (API 35), এবং Gradle 8.10.2 ইন্সটল করে একটি লোকাল বিল্ড এনভায়রনমেন্ট তৈরি করা হয়।
2. অটোমেশন স্ক্রিপ্ট: একটি ব্যাশ (Bash) বা পাইথন (Python) স্ক্রিপ্ট তৈরি করা যায় যা এআই স্টুডিও থেকে ডাউনলোড করা জিপ ফাইলটিকে নির্দিষ্ট ফোল্ডারে আনজিপ করবে। এরপর স্ক্রিপ্টটি প্রয়োজনীয় local.properties ফাইল জেনারেট করবে এবং aapt2 পাথ কনফিগার করবে।
3. লোকাল কম্পাইলেশন: স্ক্রিপ্টটি স্বয়ংক্রিয়ভাবে gradlew assembleDebug কমান্ড রান করে লোকালি স্মার্টফোনের স্টোরেজেই APK তৈরি করবে।

এই কৌশলের সীমাবদ্ধতা: Termux ব্যবহার করে কম্পাইল করা অত্যন্ত হার্ডওয়্যার-নির্ভর। আধুনিক অ্যান্ড্রয়েড প্রজেক্টে Gradle ডেমন প্রচুর মেমরি গ্রহণ করে। স্মার্টফোনে ৮ গিগাবাইটের কম র‍্যাম থাকলে 'আউট অফ মেমরি' (OOM)

ক্র্যাশ হওয়ার প্রবল সম্ভাবনা থাকে। অধিকন্তু, Termux-এ aapt2 টুলটির আর্কিটেকচারাল কম্প্যাটিবিলিটি নিয়ে প্রায়ই সমস্যা হয়, যা ম্যানুয়ালি ওভাররাইড করতে হয়। তাই একটি স্বাচ্ছন্দ্যময় এবং ক্র্যাশ-মুক্ত অভিজ্ঞতার জন্য গিটহাব অ্যাকশনস-ভিত্তিক ক্লাউড পাইপলাইনটিই বেশি যুক্তিযুক্ত।

ইঞ্জিনটি তৈরি করার জন্য ব্যবহারকারীর বাধ্যতামূলক ম্যাটেরিয়ালস

সম্পূর্ণ স্বয়ংক্রিয় অ্যাপ্লিকেশন ইঞ্জিনটি সেটআপ করতে ব্যবহারকারীকে (আপনাকে) প্রাথমিকভাবে নিচের উপাদানগুলো (Materials, API Keys, Tools) সংগ্রহ করে প্রস্তুত রাখতে হবে। এই উপাদানগুলো একবার ইঞ্জিন বা পাইপলাইনে যুক্ত করে দিলে পরবর্তীতে আর ম্যানুয়ালি কনফিগার করার প্রয়োজন হবে না।

ম্যাটেরিয়ালের নাম	সংগ্রহের উৎস ও পদ্ধতি	ইঞ্জিনে এর ভূমিকা
গিটহাব একাউন্ট এবং পার্সোনাল এক্সেস টোকেন **(PAT)**	github.com এ একাউন্ট খুলে Settings > Developer Settings থেকে টোকেন তৈরি করতে হবে।	এআই স্টুডিও থেকে কোড গ্রহণ করা এবং ক্লাউডে কম্পাইলেশন প্রসেস রান করার জন্য।
গুগল এআই স্টুডিও এপিআই কি **(API Key)**	aistudio.google.com থেকে "Get API Key" অপশন ব্যবহার করে বিনামূল্যে সংগ্রহ করা যায়।	পাইপলাইনের ভেতরে এআই ডেভেলপার হিসেবে কাজ করে লজিক ও কোড জেনারেট করার জন্য।
টেলিগ্রাম বট টোকেন **(Bot Token)**	টেলিগ্রামে @BotFather বটের মাধ্যমে নতুন বট তৈরি করে টোকেনটি সংগ্রহ করতে হবে।	গিটহাব সার্ভার থেকে স্বয়ংক্রিয়ভাবে কম্পাইল হওয়া APK ব্যবহারকারীর ফোনে ডেলিভারি করার জন্য।
টেলিগ্রাম চ্যাট আইডি **(Chat ID)**	টেলিগ্রামে @userinfobot বা অনুরূপ বট থেকে নিজস্ব একাউন্টের ইউনিক ID সংগ্রহ করতে হবে।	বট যেন নির্দিষ্টভাবে ব্যবহারকারীর ইনবক্সেই APK ফাইলটি পাঠায় তা নিশ্চিত করার জন্য।
Android Keystore ফাইল **(**ঐচ্ছিক**)**	Termux বা অনলাইনে Keytool কমান্ড ব্যবহার করে জেনারেট করা যায়।	প্লে-স্টোরে পাবলিশ করার উদ্দেশ্যে প্রোডাকশন-রেডি রিলিজ APK সাইন (Sign) করার জন্য।

এই ম্যাটেরিয়ালগুলো সংগ্রহ করার পর, গিটহাব অ্যাকশনসের "Secrets" প্যানেলে এগুলো ইনপুট হিসেবে সেভ করতে হবে, যা নিচের ক্লিক-বাই-ক্লিক গাইডে বিস্তারিতভাবে বর্ণনা করা হয়েছে।

ক্লিক-বাই-ক্লিক গাইড: মোবাইলে এপিকে ইঞ্জিন সেটআপ এবং ইন্টিগ্রেশন

নিচে গিটহাব অ্যাকশনস এবং টেলিগ্রাম বট ব্যবহার করে সম্পূর্ণ স্বয়ংক্রিয় ক্লাউড-পাইপলাইন তৈরির একটি ধাপে ধাপে নির্দেশিকা প্রদান করা হলো। এই পুরো প্রক্রিয়াটি পিসি ছাড়াই শুধুমাত্র মোবাইল ব্রাউজার ব্যবহার করে সম্পন্ন করা সম্ভব।

পর্যায় ১: গিটহাব রিপোজিটরি প্রস্তুতকরণ

1. আপনার স্মার্টফোনের ব্রাউজার (যেমন: Chrome) থেকে github.com এ লগইন করুন এবং ব্রাউজার মেনু থেকে "Desktop Site" মোড চালু করুন।
2. স্ক্রিনের উপরের ডানদিকের '+' আইকনে ক্লিক করে "New Repository" নির্বাচন করুন।
3. রিপোজিটরির একটি নাম দিন (যেমন: My-Auto-APK-Engine) এবং প্রাইভেসি সেটিংসে এটিকে "Private" হিসেবে সেট করুন।
4. "Add a README file" বক্সে টিক চিহ্ন দিন এবং নিচে "Create repository" বাটনে ক্লিক করুন।

পর্যায় ২: টেলিগ্রাম বট কনফিগারেশন এবং সিক্রেটস যুক্ত করা

1. টেলিগ্রাম অ্যাপ ওপেন করে সার্চ বারে @BotFather লিখুন এবং বটটি স্টার্ট করুন।
2. /newbot কমান্ড দিয়ে একটি নতুন বট তৈরি করুন এবং বটের নাম ও ইউজারনেম দিন। BotFather

আপনাকে একটি HTTP API Token প্রদান করবে; এটি কপি করে সযত্নে সংরক্ষণ করুন।

3. এরপর টেলিগ্রামে @userinfobot সার্চ করে স্টার্ট করুন। এটি আপনার নিজস্ব Chat ID (একটি সংখ্যার স্ট্রিং) প্রদান করবে, এটিও কপি করুন।
4. এবার ব্রাউজারে আপনার গিটহাব রিপোজিটরিতে ফিরে যান। উপরের মেনু থেকে "Settings" এ প্রবেশ করুন।
5. বাম দিকের সাইডবার থেকে "Secrets and variables" অপশনটি সম্প্রসারিত করে "Actions" এ ক্লিক করুন।
6. "New repository secret" বাটনে ক্লিক করুন।
7. **Name** বক্সে লিখুন TELEGRAM_BOT_TOKEN এবং **Secret** বক্সে আপনার BotFather থেকে পাওয়া টোকেনটি পেস্ট করে "Add secret" এ ক্লিক করুন।
8. পুনরায় "New repository secret" এ ক্লিক করুন। **Name** বক্সে লিখুন TELEGRAM_CHAT_ID এবং **Secret** বক্সে আপনার টেলিগ্রাম চ্যাট আইডিটি পেস্ট করে "Add secret" এ ক্লিক করুন।

পর্যায় ৩: গিটহাব অ্যাকশনস (GitHub Actions) অটোমেশন স্ক্রিপ্ট তৈরি

1. গিটহাব রিপোজিটরির মূল পেজে ফিরে যান।
2. "Add file" ড্রপডাউন মেনু থেকে "Create new file" এ ক্লিক করুন।
3. ফাইলের নাম ও পাথ দেওয়ার জায়গায় হুবহু এটি লিখুন: .github/workflows/build-apk.yml (এটি স্বয়ংক্রিয়ভাবে ফোল্ডার স্ট্রাকচার তৈরি করবে)।
4. নিচের YAML কোডটি সম্পূর্ণ কপি করে এডিটরের বিশাল সাদা বক্সে পেস্ট করুন। এই কোডটি উবুন্টু সার্ভারে জাভা সেটআপ করবে, আপনার অ্যাপ কম্পাইল করবে এবং টেলিগ্রামে পাঠিয়ে দেবে।

```
name: Auto Build and Deliver Android APK
on:
  push:
    branches: [ main ]
  workflow_dispatch:

jobs:
  build-and-deliver:
    runs-on: ubuntu-latest
    steps:
    - name: Checkout Source Code
      uses: actions/checkout@v4

    - name: Set up Java Development Kit (JDK) 17
      uses: actions/setup-java@v4
      with:
        java-version: '17'
        distribution: 'temurin'
        cache: 'gradle'

    - name: Grant execution rights to Gradle
      run: chmod +x ./gradlew

    - name: Compile the App (Assemble Debug)
      run: ./gradlew assembleDebug --stacktrace

    - name: Deliver APK via Telegram Bot
      env:
```

```
    BOT_TOKEN: ${{ secrets.TELEGRAM_BOT_TOKEN }}
    CHAT_ID: ${{ secrets.TELEGRAM_CHAT_ID }}
  run: |
    curl -F chat_id=$CHAT_ID \
    -F document=@"app/build/outputs/apk/debug/app-debug.apk" \
    -F caption="আপনার প্রম্পট থেকে জেনারেট করা অ্যাপটি সফলভাবে কম্পাইল হয়েছে!" \
    https://api.telegram.org/bot$BOT_TOKEN/sendDocument
```

1. স্ক্রিনের ডানদিকে "Commit changes" বাটনে ক্লিক করে ফাইলটি সেভ করুন। আপনার ইঞ্জিন এখন সম্পূর্ণ প্রস্তুত!

পর্যায় ৪: ইঞ্জিনটি ব্যবহার করে অ্যাপ তৈরি করা

1. মোবাইলের ব্রাউজার থেকে aistudio.google.com এ যান এবং "Build an Android app" অপশনটি সিলেক্ট করুন।
2. আপনার অ্যাপের ব্লুপ্রিন্ট বা আইডিয়া প্রম্পট হিসেবে লিখুন। (যেমন: "একটি সাধারণ পার্সোনাল ফাইন্যান্স ট্র্যাকার অ্যাপ তৈরি করো যেখানে প্রতিদিনের খরচ সেভ করা যাবে।")
3. এআই স্টুডিও কয়েক মিনিটের মধ্যে সম্পূর্ণ প্রজেক্টের কোড জেনারেট করবে।
4. প্রজেক্ট সম্পন্ন হলে, এআই স্টুডিওর এক্সপোর্ট (Export) বা ডাউনলোড অপশন থেকে কোডের জিপ ফাইলটি ডাউনলোড করুন।
5. আপনার মোবাইলের গিটহাব রিপোজিটরিতে গিয়ে "Add file" > "Upload files" ব্যবহার করে জিপ ফাইলটি আনজিপ করে তার ভেতরের কন্টেন্টগুলো পুশ করে দিন (অথবা গিটহাব মোবাইল অ্যাপ/টার্মিনাল দিয়ে পুশ করুন)।
6. যেই মুহূর্তে কোড গিটহাবে আপলোড হবে, গিটহাব অ্যাকশনস স্বয়ংক্রিয়ভাবে ট্রিগার হবে। গিটহাবের "Actions" ট্যাবে গিয়ে আপনি লাইভ প্রোগ্রেস দেখতে পারবেন।
7. ৪-৬ মিনিট পর আপনার টেলিগ্রাম বটে সম্পূর্ণ সাইন করা এবং ত্রুটিমুক্ত ইনস্টলযোগ্য .apk ফাইল সরাসরি মেসেজ হিসেবে চলে আসবে, যা আপনি এক ক্লিকেই ফোনে ইনস্টল করতে পারবেন।

পিসি ছাড়া স্বয়ংক্রিয় অ্যাপ্লিকেশন ইঞ্জিনের সীমাবদ্ধতা এবং প্রায়োগিক সম্ভাবনা

পিসি বা ল্যাপটপের সাহায্য ছাড়া শুধুমাত্র একটি স্মার্টফোনে এআই এবং ক্লাউডের সমন্বয়ে একটি স্বয়ংক্রিয় অ্যাপ্লিকেশন ইঞ্জিন তৈরির ধারণাটি উদ্ভাবনী হলেও, এর কিছু সুনির্দিষ্ট সীমাবদ্ধতা এবং কাঠামোগত চ্যালেঞ্জ রয়েছে, যা সতর্কতার সাথে মূল্যায়ন করা আবশ্যক।

প্রযুক্তিগত সীমাবদ্ধতা

১. **"Parsing the Package" ত্রুটির ঝুঁকি:** গুগল এআই স্টুডিওর ডিফল্ট ব্রাউজার এমুলেটর এবং জিপ এক্সপোর্ট সিস্টেম অনেক সময় এমন প্রাক-কম্পাইলড এপিকে তৈরি করে যা নির্দিষ্ট ডিভাইসের আর্কিটেকচারকে সাপোর্ট করে না। ফলে ব্যবহারকারী যখন সরাসরি জিপ থেকে এপিকে ইনস্টল করার চেষ্টা করেন, তখন অ্যান্ড্রয়েড সিস্টেম প্যাকেজ পার্সিং ত্রুটি প্রদর্শন করে। তবে এই প্রতিবেদনে প্রস্তাবিত ক্লাউড-নেটিভ কম্পাইলেশন (গিটহাব অ্যাকশনস) কৌশলটি এই সমস্যার একটি স্থায়ী সমাধান, কারণ এটি স্ট্যান্ডার্ড টুলস ব্যবহার করে জেনেরিক এপিকে তৈরি করে যা যেকোনো আর্কিটেকচারে পার্স হতে সক্ষম।
২. **সার্ভার-সাইড এবং থার্ড-পার্টি ইন্টিগ্রেশন চ্যালেঞ্জ:** গুগল এআই স্টুডিওর 'Antigravity Agent' মূলত ক্লায়েন্ট-সাইড (Client-side) বা ফ্রন্টএন্ড নির্ভর অ্যাপ জেনারেট করতে পারদর্শী। মাল্টিপ্লেয়ার ফাংশনালিটি, সিক্রেটস ম্যানেজমেন্ট, গুগল ওয়ার্কস্পেস এপিআই, অথবা ফায়ারবেস (Firebase) এর মতো জটিল সার্ভার-সাইড রানিং ডাটাবেস ইন্টিগ্রেশন এই মুহূর্তে এআই স্টুডিওর স্বয়ংক্রিয় কোড জেনারেশনে পুরোপুরি সমর্থিত নয়। ফলে জটিল ব্যাকএন্ড সম্বলিত অ্যাপ তৈরি

করতে হলে এআই ইঞ্জিনকে ম্যানুয়ালি গাইড করতে হয় এবং অনেক ক্ষেত্রে অতিরিক্ত এপিআই কি সরবরাহ করে কোড মডিফাই করতে হয়।
৩. ইউজার ইন্টারফেস **(UI)** এবং ডিজাইন কাস্টমাইজেশন**:** 'Vibe Coding' বা প্রম্পট-ভিত্তিক কোডিংয়ে এআই মূলত স্ট্যান্ডার্ড মেটেরিয়াল ডিজাইন (Material Design) প্যাটার্ন অনুসরণ করে। যদি ব্যবহারকারী অত্যন্ত কাস্টমাইজড, পিক্সেল-পারফেক্ট ইউজার ইন্টারফেস বা জটিল অ্যানিমেশন চান, তবে তা প্রম্পটের মাধ্যমে অর্জন করা কঠিন। মোবাইলের ছোট স্ক্রিনে লাইভ প্রিভিউ দেখে সূক্ষ্ম ভিজ্যুয়াল এডিটিং করা এবং বারবার প্রম্পট দিয়ে এআইকে দিয়ে ত্রুটি সংশোধন করানো বেশ সময়সাপেক্ষ ব্যাপার হতে পারে।
৪. অন**-**ডিভাইস কম্পাইলেশনের হার্ডওয়্যার সীমাবদ্ধতা **(Termux** এর ক্ষেত্রে**):** যদি ব্যবহারকারী ক্লাউড ইঞ্জিনের বদলে টার্মাক্স (Termux) ব্যবহার করে অফলাইনে অ্যাপ কম্পাইল করতে চান, তবে ডিভাইসের হার্ডওয়্যারের ওপর প্রচণ্ড চাপ পড়ে। গ্র্যাডল (Gradle) ডেমন এবং জাভা কম্পাইলার প্রচুর মেমরি (RAM) গ্রহণ করে। এন্ট্রি লেভেল বা মিড-রেঞ্জ স্মার্টফোনে 'আউট অফ মেমরি' (OOM) জনিত ক্র্যাশ হওয়ার প্রবল সম্ভাবনা থাকে। এ কারণে গিটহাব অ্যাকশনসের মতো ক্লাউড সার্ভারের ওপর নির্ভর করাই সর্বোত্তম সমাধান।

প্রায়োগিক সম্ভাবনা এবং ভবিষ্যৎ স্কোপ

উপর্যুক্ত সীমাবদ্ধতাগুলো সত্ত্বেও, একটি পিসি-বিহীন অ্যাপ্লিকেশন ইঞ্জিনের প্রায়োগিক সম্ভাবনা অত্যন্ত ব্যাপক।

- দ্রুত প্রোটোটাইপিং**:** উদ্যোক্তা বা ডিজাইনাররা কোনো আইডিয়া মাথায় আসার সাথে সাথেই মোবাইল থেকে প্রম্পট দিয়ে সেটির একটি ফাংশনাল প্রোটোটাইপ তৈরি করে ক্লায়েন্ট বা ইনভেস্টরদের দেখাতে পারবেন।
- পার্সোনাল ইউটিলিটি অ্যাপ**:** ব্যক্তিগত প্রয়োজনের জন্য ছোট ছোট টুলস (যেমন: কাস্টমাইজড টু-ডু লিস্ট, বাজেট ট্র্যাকার, ক্যালকুলেটর) তৈরি করার ক্ষেত্রে এই ইঞ্জিনটি একটি গেম-চেঞ্জার।
- হার্ডওয়্যার সেন্সর ইন্টিগ্রেশন**:** যেহেতু এআই স্টুডিও নেটিভ অ্যান্ড্রয়েড কোড তৈরি করে, তাই এই ইঞ্জিন দিয়ে তৈরি করা অ্যাপগুলো ফোনের নেটিভ জিপিএস, ব্লুটুথ, এনএফসি (NFC) বা ক্যামেরা সেন্সর নিখুঁতভাবে ব্যবহার করতে পারে, যা সাধারণ ওয়েব-র‍্যাপার বা নো-কোড টুলে সম্ভব হয় না।

একবার গিটহাব অ্যাকশনস এবং টেলিগ্রাম বটের ক্লাউড পাইপলাইনটি সেটআপ হয়ে গেলে, ব্যবহারকারীকে পরবর্তীতে আর কোড বা আর্কিটেকচার নিয়ে ভাবতে হবে না। শুধু প্রম্পট প্রদান করলেই ইঞ্জিনটি স্বয়ংক্রিয়ভাবে বাকি কাজ সম্পন্ন করে একটি রেডি-টু-ইউজ অ্যাপ্লিকেশন সরবরাহ করবে। এই পদ্ধতিটি মোবাইল-ভিত্তিক সফটওয়্যার ডেভেলপমেন্টের ক্ষেত্রে এক নতুন যুগের সূচনা করে, যেখানে পিসির সীমাবদ্ধতা সৃজনশীলতাকে আর আটকে রাখতে পারে না।

যেসব কাজের উদ্ধৃতি যোগ করা হয়েছে

1. Google AI Studio Pricing 2026: Free Tier, API Costs & Plans | No Code MBA, https://www.nocode.mba/articles/google-ai-studio-pricing 2. Free Trial and Free Tier Services and Products - Google Cloud, https://cloud.google.com/free 3. Google AI Studio Models: Accessing Gemini & Veo in Bangladesh - MangoMind Blog, https://www.mangomindbd.com/blog/google-ai-studio-bangladesh-access 4. Available regions for Google AI Studio and Gemini API, https://ai.google.dev/gemini-api/docs/available-regions 5. Google Gemini (mobile app) — Grokipedia, https://grokipedia.com/page/Google_Gemini_mobile_app 6. google ai studio How many free requests per day? : r/googlecloud - Reddit, https://www.reddit.com/r/googlecloud/comments/1rc3zmd/google_ai_studio_how_many_free_requests_per_day/ 7. Google AI Studio Free Plans: Free Tier Limits, Subscription Plans, Trial Access, And Feature Availability, https://www.datastudios.org/post/google-ai-studio-free-plans-free-tier-limits-subscription-plans-trial-access-and-feature-availab 8. Bring any idea to life: Google AI Studio at I/O 2026, https://blog.google/innovation-and-ai/technology/developers-tools/google-ai-studio-io-2026/ 9. Google AI Studio: Build Android Apps from Text Prompts - Digital Applied, https://www.digitalapplied.com/blog/google-ai-studio-android-apps-text-prompts 10. 8 Best AI App Studio Options for 2026: Tested & Ranked, https://x1.new/post/ai-app-studio-buyers-guide

11. Android App Development in Google AI Studio: The Ultimate Vibe Coding Guide - incrypted, https://incrypted.com/en/android-app-development-google-ai-studio/ 12. Build Android Apps in Google AI Studio | Gemini API, https://ai.google.dev/gemini-api/docs/aistudio-android 13. Build native Android apps in Google AI Studio | Android Developers' Blog, https://developer.android.com/blog/posts/build-native-android-apps-in-google-ai-studio 14. How to Export Code Files from Google AI Studio | Download & Edit Your App Files, https://www.youtube.com/watch?v=MiytvVzPwW4 15. How to Download Your App from Google AI Studio (Step-by-Step Tutorial) 2026 - YouTube, https://www.youtube.com/watch?v=nXZPs_JjI-c 16. Build Android App in Google AI Studio (Install APK on Phone No Coding) - YouTube, https://www.youtube.com/watch?v=0repQ5Tki6I 17. There was a problem parsing the package - Grokipedia, https://grokipedia.com/page/There_was_a_problem_parsing_the_package 18. There was a problem parsing the package · Issue #329 · google-ai-edge/gallery - GitHub, https://github.com/google-ai-edge/gallery/issues/329 19. "Parse Error : There is a problem parsing the package" while installing Android application, https://stackoverflow.com/questions/1492401/parse-error-there-is-a-problem-parsing-the-package-while-installing-android 20. APK Builder - Generate Android APK Files - Newly.app, https://newly.app/apk-builder 21. 10 Free AI Tools to Automatically Create a Full Mobile App Using Only Prompts - Medium, https://medium.com/@christophorusdryantoro/10-free-ai-tools-to-automatically-create-a-full-mobile-app-using-only-prompts-5352a3e733df 22. Deploy a Google AI Studio app to a custom domain - VibeDeploy, https://vibedeploy.be/for/google-ai-studio 23. Deploy AI-built websites: tool-by-tool guides | VibeDeploy, https://vibedeploy.be/for 24. CI: GitHub Actions build APK on push · Issue #10 · alexeygrigorev/pocketshell, https://github.com/alexeygrigorev/pocketshell/issues/10 25. Automating Android APK & AAB Generation with GitHub Actions - Medium, https://medium.com/@myofficework000/automating-android-apk-aab-generation-with-github-actions-0910888de1b1 26. Automating Android APK Builds with GitHub Actions (The Sane Way) - DEV Community, https://dev.to/ronynn/automating-android-apk-builds-with-github-actions-the-sane-way-1h95 27. Build and Share Android Apps with Ease Using GitHub Actions, https://blog.dhina17.dev/build-and-share-android-apps-with-ease-using-github-actions 28. GitHub Actions workflow for android projects with telegram bots integration : This workflow can notify about pushes and builds in telegram chats and can also send apk to chats, https://gist.github.com/sainak/b32367294fd1381ac8854e67e21064cf 29. Building Android applications in Termux using Gradle - Grokipedia, https://grokipedia.com/page/Building_Android_applications_in_Termux_using_Gradle 30. illegalvoid/termux-build-apps: build APKs with Termux, without needing Proot - GitHub, https://github.com/illegalvoid/termux-build-apps 31. Build android app on termux - GitHub Gist, https://gist.github.com/tmatz/817bf03433e059bf89c63dc33f286ccb 32. How to Make/Build Android .apk in Termux #25793 - GitHub, https://github.com/termux/termux-packages/discussions/25793 33. Build Android Apps Online | No Android Studio Required - RapidNative, https://www.rapidnative.com/android-app-builder-online 34. subinps/TelePlay: A self-hosted media streaming solution that lets you store, organize, and stream your video files from Telegram Servers. Access your media library from anywhere via a web interface or Android TV app / Mobile App. · GitHub, https://github.com/subinps/TelePlay 35. GitHub Actions: Build Your Android App, https://dl.iir.edu.ua/iir-news/github-actions-build-your-android-app-1767647084 36.

How to Upload a Folder to Your GitHub Repository On Android Using Termux,
https://github.com/Jobians/termux-github-upload-tutorial-for-android