

# Charles Henry Dubreuil

Brooklyn, NY • +1-929-713-9400 • [Charles\\_Dubreuil053100@yahoo.com](mailto:Charles_Dubreuil053100@yahoo.com) •  
<https://www.linkedin.com/in/charles-dubreuil> • <https://github.com/charleshenrydubreuil>

## EDUCATION

CUNY Brooklyn College – Brooklyn, NY

May 2024

### Bachelor of Science in Computer Science

- Relevant coursework: Data Structures and Algorithms, Databases, Operating Systems, Mobile Application Development, Web Development, Game Development

## SKILLS & CERTIFICATIONS

**Languages:** Kotlin, Java, Swift, Dart, Python, TypeScript, JavaScript, Rust, SQL, C, C++, C#, HTML, CSS, GraphQL | **Frameworks:**

Android SDK, Jetpack Compose, Spring Boot, React, Next.js, Flutter, Kotlin Multiplatform (KMP/KMM), Tailwind CSS | **Android & Mobile Technologies:** Jetpack (Compose, ViewModel, LiveData, Navigation, Room, DataStore, WorkManager), Retrofit, OkHttp, Gson, CameraX, Paging 3, ExoPlayer | **APIs & Services:** Google Maps API, Google Places API, Firebase Realtime Database, Firebase Firestore, Google Custom Search API, YouTube API, PayPal SDK, Stripe SDK, Chromecast SDK | **Architecture & Design Patterns:** MVVM, MVI, Clean Architecture, Repository Pattern | **Authentication & Security:** Firebase Authentication, OAuth 2.0, JWT, Google Sign-In, Apple Sign-In, Okta | **Dependency Injection:** Hilt, Dagger | **Testing:** JUnit, Espresso, Mockito, MockK | **Developer Tools:** Git, GitHub, Gradle, Maven, Postman, Android Studio, VS Code, Xcode | **Operating Systems:** macOS, Windows, Linux, Kali Linux

## EXPERIENCE

Techfield - Android Developer Intern | Atlanta, Georgia

September 2025 - Current

- Developing Android features using Kotlin, Jetpack Compose, MVVM, Clean Architecture, Coroutines, Hilt, Retrofit, and Room, improving feature delivery speed by 30% through a cleaner modular architecture.
- Building modern Compose UI screens, handling state via ViewModel + StateFlow, and implementing offline-first functionality using Room persistence.
- Collaborating with designers and backend engineers using Git-based workflows, maintaining feature branches, writing meaningful commits, performing code reviews, and participating in Agile sprints and stand-ups.

## PROJECTS

Modern Dating App | Personal Project

GitHub | [View Project](#)

- Built a Tinder-style Android app using Jetpack Compose, Kotlin, MVVM, and Clean Architecture with repository and mapper layers.
- Integrated Retrofit and Coroutines to fetch nearby users by geolocation, handling network success/error states through domain models.
- Added CameraX for seamless in-app profile photo capture, enabling a fully self-contained 100% in-app onboarding flow with no external apps required.
- Implemented Hilt DI for ViewModels, Use Cases, and repositories; used StateFlow to manage UI reactivity across swiping, profile, and onboarding features.

Country Search App | Personal Project

GitHub | [View Project](#)

- Developed an MVVM-structured Android app consuming a public Countries API using Kotlin, Retrofit, LiveData, and Repository abstraction.
- Used Coroutines to fetch API data and execute Room Database operations asynchronously for offline caching.
- Implemented dynamic search, A-Z sorting, and region filtering; built Room-powered bookmarks with full CRUD support, increasing data navigation efficiency by 50%.
- Built a lifecycle-aware ExoPlayer login screen featuring an auto-looping video background tied to the Activity lifecycle, and implemented Firebase Authentication for secure user login.

Tech News App | Personal Project

GitHub | [View Project](#)

- Built a Kotlin-based Android app using Retrofit, REST APIs, XML UI, and RecyclerView + Coil to fetch and display technology news; implemented an article detail screen using Intents and Activity navigation.
- Added dynamic search, adapter filtering, and API response validation; improved resilience against network errors, empty states, and slow API responses through structured error handling.
- Refactored ViewHolders, adapters, and data models for cleaner architecture; documented code structure and applied best practices for maintainable Android development.