QuickCash Android App - Setup and Usage Guide

⚠️ Important Notice: Firebase Configuration Required

The one-month free trial for Firebase has ended, but you can still use Firebase for FREE under the Spark Plan with these limitations:

Firebase Free Tier (Spark Plan) Limits:

- ✅ **Authentication**: Unlimited users
- ✅ **Realtime Database**: 1GB storage, 10GB/month download
- ✅ **Hosting**: 10GB storage, 360MB/day download
- ✅ **Cloud Functions**: 125K invocations/month
- ❌ **Some advanced features require paid plan**

📋 Prerequisites

Required Accounts:

1. **Google Account** (for Firebase)
2. **GitHub Account** (optional, for version control)

Required Software:

- Android Studio (Arctic Fox or later)
- Java JDK 8 or higher
- Android SDK (API 21+)

🚀 Setup Instructions

Step 1: Set Up Firebase Project

1. **Go to Firebase Console**
2. **Click "Create Project"**
   - Project name: QuickCash
   - Disable Google Analytics (not needed for this app)

   - Click "Create Project"
3. **Add Android App to Firebase**
   - In your Firebase project, click "Android icon"
   - Android package name: com.example.quickcash
   - App nickname: QuickCash
   - Debug signing certificate (optional)
   - Click "Register app"
4. **Download** google-services.json
   - Place this file in: app/ folder of your project
   - **DO NOT commit this file to GitHub** (it's in .gitignore)

Step 2: Configure Firebase Authentication

1. **In Firebase Console, go to:**
   - **Authentication** → **Sign-in method**
   - Enable **Email/Password**
   - Click "Save"
2. **Set up Realtime Database:**
   - **Realtime Database** → **Create Database**
   - Start in **test mode** (for development)
   - Location: Choose closest to your region

Step 3: Import and Build Project

1. **Open Android Studio**
2. **Import Project:** File → New → Import Project → Select your project folder
3. **Sync Project with Gradle Files:**
   - Android Studio will automatically sync
   - If errors occur, go to: File → Sync Project with Gradle Files
4. **Build the Project:**
   - Build → Make Project

   - Ensure no compilation errors

📱 How to Use the App

User Registration Flow:

1. **Launch the App**
   - App opens to Create Account screen
2. **Create New Account:**
   - Enter Username
   - Enter valid Email address
   - Create Password (must be 8+ characters with special character)
   - Confirm Password
   - Answer 3 security questions:
     - What is your pet's name?
     - What was the name of your first school?
     - What is your favorite movie?
3. **Account Creation:**
   - Click "Create Account"
   - User is automatically registered in Firebase
   - Redirected to Login screen

Login Flow:

1. **Enter Credentials:**
   - Email address
   - Password
2. **Authentication:**
   - Firebase verifies credentials
   - Successful login redirects to Dashboard
   - Failed login shows error message

Password Recovery:

1. **On Login Screen:**
   - Click "Forgot Password?"

2. **Email Verification:**
    - Enter registered email address
    - Click "Verify Email"
3. **Security Questions:**
    - Answer the 3 security questions you set during registration
    - Enter new password
    - Confirm new password
    - Click "Submit"
4. **Password Reset:**
    - Password is updated in Firebase
    - Redirect to Login screen

Dashboard:

1. **Main Screen:**
    - Displays welcome message
    - Logout button
2. **Logout:**
    - Click "Logout" button
    - Returns to Login screen
    - Firebase session ended

🔧 Firebase Configuration Details

Required Firebase Services:

1. **Authentication**
    - Email/Password provider enabled
    - No additional configuration needed
2. **Realtime Database**
    - Structure: Users/{userId}/
    - Stores: username, email, security answers

Database Structure:

json

```
{
  "Users": {
    "firebase-user-id": {
      "username": "john_doe",
      "email": "john@example.com",
      "password": "encrypted_password",
      "securityAns": {
        "question1": "answer1",
        "question2": "answer2",
        "question3": "answer3"
      }
    }
  }
}
```

🛠️ Troubleshooting Common Issues

Build Errors:

1. **"Failed to find target with hash string"**
   - Install required SDK versions in Android Studio
   - Go to: Tools → SDK Manager
2. **"Google services JSON not found"**
   - Ensure google-services.json is in app/ folder
   - Check file name spelling
3. **Firebase dependencies errors**
   - Clean project: Build → Clean Project
   - Rebuild project: Build → Rebuild Project

Runtime Errors:

1. **"Authentication Failed"**

   - Check internet connection
   - Verify Firebase project configuration
   - Ensure Email/Password auth is enabled in Firebase Console

2. **"Database permission denied"**
   - Check Realtime Database rules in Firebase Console
   - For development, use these rules:

     json

     ```
     {
       "rules": {
         ".read": "auth != null",
         ".write": "auth != null"
       }
     }
     ```

3. **App crashes on startup**
   - Check if google-services.json is properly placed
   - Verify package name matches in manifest and Firebase

📊 Testing the Application

Test Scenarios:

1. **Successful Registration**
   - Fill all fields correctly
   - Should create Firebase user
   - Should store user data in Realtime Database
2. **Duplicate Email Registration**
   - Try registering with same email twice
   - Should show "Email already exists" error
3. **Password Recovery**
   - Use registered email
   - Answer security questions correctly
   - Should allow password reset

4. **Invalid Login**
    - Wrong email/password combination
    - Should show authentication error

💡 Development Notes

Security Considerations:

- Passwords are handled by Firebase Authentication (encrypted)
- Security answers stored in plain text (consider encryption for production)
- Database rules should be tightened for production

Scalability:

- Current structure supports basic user management
- Can be extended with job posting features
- Payment integration can be added later

Customization:

- Modify security questions in activity_create_account.xml
- Change color schemes in square_button.xml and other layout files
- Add additional user fields in UserModel.java

📞 Support

If Firebase Setup Fails:

1. Check Firebase Documentation
2. Verify Google account has no billing restrictions
3. Ensure project is in supported region

Common Firebase Issues:

- **"Billing account required"**: This shouldn't happen for basic authentication and database services
- **"Project creation failed"**: Try different project name or check Google account permissions

🎯 Next Steps for Enhancement

1. **Add email verification** for new registrations
2. **Implement password strength meter**
3. **Add user profile management**

4. **Include job posting functionality**
5. **Add in-app messaging system**

This app provides a solid foundation for a job marketplace application with complete user authentication and password recovery systems using Firebase backend services.