Christ Collective App – Replit & App Version Setup Guide

1. Project Setup on Replit

1. Create a new Replit project:
2. Template: Node.js (or HTML/CSS/JS if vanilla)
3. Name: ChristCollectiveApp
4. Install Ionic/Capacitor:

```
npm install -g @ionic/cli
ionic start app-version blank --type=react
cd app-version
npm install @capacitor/core @capacitor/cli
npx cap init "ChristCollectiveApp" "com.christcollective.app"
```

2. Shared Backend Usage

- Use the same backend for both web and app.
- Example API fetch (React/JS):

```
fetch('https://your-backend.com/api/posts')
  .then(res => res.json())
  .then(setPosts);
```

- Both platforms will always see the same content.

3. Detecting Platform (App vs Web)

```
import { Capacitor } from '@capacitor/core';
const isApp = Capacitor.getPlatform() !== 'web';
```

- true for iOS/Android app - false for website

4. Conditional Layouts

Separate Components for App vs Web

```
function HomePage() {
  const isApp = Capacitor.getPlatform() !== 'web';
  return (
    <div>
      {isApp ? <AppHome /> : <WebHome />}
    </div>
  );
}
```

Shared Components with Conditional Rendering

```
function MainContent({ isApp }) {
  return (
    <div>
      {isApp ? <MobileSignupBanner /> : <DesktopInfoGrid />}
      <PostsList />  {/* same data for both */}
    </div>
  );
}
```

5. Using Mock Data (Optional for Preview)

- Create `mockData.js`:

```
export const mockUsers = [
  { id: 1, name: 'Test User 1' },
  { id: 2, name: 'Test User 2' },
];

export const mockPosts = [
  { id: 1, title: 'Welcome!', content: 'This is a mock post.' },
  { id: 2, title: 'Preview Mode', content: 'You can edit this in
Replit!' },
];
```

- Use in app:

```
import { mockUsers, mockPosts } from './mockData.js';

setUsers(mockUsers);
setPosts(mockPosts);
```

6. Mobile Layout Recommendations

- Prioritize signup/login on home page for app.
- Use Ionic components for native feel (`ion-card`, `ion-button`, `ion-content`).
- Single-column layout with larger tap targets.
- Minimize sidebars/menus that are only relevant on desktop.

7. Running in Replit

1. Install dependencies: `npm install`
2. Serve app locally:

```
ionic serve --host 0.0.0.0 --port 3000
```

3. Replit gives a live URL.
4. Edit layouts live and test changes for web; build app separately for iOS/Android.

8. Preparing for Codemagic (iOS Preview)

- Make sure your app build generates a `.app` file for iOS simulator.
- Use mock data or connect to live backend for preview.
- Push the project to GitHub for Codemagic to build.