

# 30-Day Plan to Ship Radiant to App Stores

---

**Goal:** Launch Radiant on iOS (TestFlight/App Store) and Android (Google Play) within 30 days.

**Current Status:** MVP complete with all core features working

---

## Week 1: Polish & Prepare (Days 1-7)

### Day 1: Testing & Bug Fixing

**Time:** 3-4 hours

- ☐ **Test complete user flow**
  - Fresh install → Onboarding → Create journal → View journal
  - Test on iOS simulator
  - Test on Android emulator
  - Document any bugs or crashes
- ☐ **Test data persistence**
  - Add journal entries
  - Close app completely
  - Reopen and verify data is still there
  - Test with longer entries (stress test)
- ☐ **Create bug list**
  - Prioritize: Critical (blocks launch) vs Nice-to-fix
  - Focus only on critical bugs this week

**Deliverable:** List of critical bugs to fix

---

### Day 2: Fix Critical Bugs

**Time:** 4-5 hours

- ☐ **Fix all critical bugs from Day 1**
  - Work through list one by one
  - Test each fix immediately
  - Don't add new features (tempting but stay focused!)
- ☐ **Test on real device if possible**
  - Use Expo Go app on your phone
  - Feels more real than simulator
  - Note any performance issues

**Deliverable:** All critical bugs resolved

---

## Day 3: UI Polish & Consistency

**Time:** 3-4 hours

- ☐ **Visual consistency check**
  - All buttons same style?
  - Colors consistent across screens?
  - Spacing feels uniform?
  - Font sizes make sense?
- ☐ **Empty states**
  - What happens when journal is completely empty?
  - Clear messaging for new users
  - Helpful prompts to get started
- ☐ **Loading states**
  - Add loading indicators where needed
  - Smooth transitions between screens
  - No jarring jumps

**Deliverable:** Polished, consistent UI

---

## Day 4: App Icons & Splash Screen

**Time:** 2-3 hours

- ☐ **Design app icon** (or use AI to generate)
  - 1024x1024px base image
  - Simple, recognizable at small sizes
  - Matches app aesthetic (peach/sage green theme)
  - Tools: Canva, Figma, or Midjourney/DALL-E
- ☐ **Create splash screen**
  - Simple logo or app name
  - Peach background matching brand
  - Update in app.json
- ☐ **Update app.json metadata**
  - App name: "Radiant"
  - Slug: "radiant-app"
  - Description (short and compelling)
  - Version: 1.0.0

**Deliverable:** Professional app icon and splash screen

---

## Day 5: App Store Preparation - Research

**Time:** 3-4 hours

- ☐ **Create Apple Developer Account** (\$99/year)
  - Go to [developer.apple.com](https://developer.apple.com)
  - Sign up (takes 24-48 hours for approval)
  - Start this ASAP - don't wait!
- ☐ **Create Google Play Developer Account** (\$25 one-time)
  - Go to [play.google.com/console](https://play.google.com/console)
  - Sign up (usually approved within hours)
- ☐ **Research App Store requirements**
  - Screenshots needed (6.5" iPhone, 12.9" iPad for iOS)
  - Privacy policy requirements
  - Age rating questionnaire
  - Read App Store Review Guidelines
- ☐ **Research Google Play requirements**
  - Screenshots needed (phone, 7" tablet, 10" tablet)
  - Feature graphic (1024x500)
  - Privacy policy
  - Content rating questionnaire

**Deliverable:** Developer accounts created (or in progress), requirements documented

---

## Day 6: Marketing Assets - Screenshots

**Time:** 3-4 hours

- ☐ **Take app screenshots**
  - Onboarding screens (show the value)
  - Journal input screens (show ease of use)
  - Journal view (show the beautiful result)
  - Use simulator "Save Screenshot" feature
- ☐ **Create promotional screenshots** (optional but recommended)
  - Add text overlays explaining features
  - "Track your affirmations"
  - "Build your morning routine"
  - "Visualize your goals"

- Tools: Canva, Figma, or screenshots.pro
- ☐ **Write app descriptions**
  - Short description (80 characters for Google Play)
  - Full description (4000 characters max)
  - Focus on benefits, not features
  - Include keywords: gratitude, journal, affirmations, self-improvement

**Deliverable:** Screenshots and app descriptions ready

---

## Day 7: Privacy Policy & Legal

**Time:** 2-3 hours

- ☐ **Create Privacy Policy**
  - Required by both app stores
  - Use generator: [app-privacy-policy-generator.firebaseapp.com](https://app-privacy-policy-generator.firebaseapp.com)
  - Be honest about data collection (you're using AsyncStorage = local only)
  - Host on simple webpage or GitHub Pages
- ☐ **Review what data you collect**
  - User data: Journal entries (stored locally)
  - Analytics: None yet (keep it simple for MVP)
  - Third-party services: None yet
- ☐ **Create Terms of Service** (optional for MVP)
  - Can use template generator
  - Not strictly required but professional

**Deliverable:** Privacy policy live on a URL

---

## Week 2: Build & Submit (Days 8-14)

### Day 8: Build Configuration - iOS

**Time:** 3-4 hours

- ☐ **Update app.json for production**
  - Set correct bundle identifier: com.yourname.radiant
  - Set version: "1.0.0"
  - Set build number: 1
  - Add iOS-specific config
- ☐ **Install EAS CLI** (Expo Application Services)

```
npm install -g eas-cli  
eas login
```

- ☐ **Configure EAS Build**

```
eas build:configure
```

- Choose iOS and Android
- Creates eas.json

- ☐ **Run first iOS build** (this can take 20-30 minutes)

```
eas build --platform ios --profile preview
```

- Creates IPA file for testing
- Fix any build errors that come up

**Deliverable:** Successful iOS build

---

## Day 9: Build Configuration - Android

**Time:** 3-4 hours

- ☐ **Configure Android build settings**
  - Update package name in app.json
  - Set version code and version name
  - Add adaptive icon configuration
- ☐ **Create app signing key** (for Google Play)

```
eas build --platform android --profile preview
```

- EAS handles this automatically
- Keep credentials safe!

- ☐ **Run first Android build**
  - Test the APK on a real device or emulator
  - Fix any Android-specific issues

**Deliverable:** Successful Android build

---

## Day 10: TestFlight Setup (iOS Beta)

**Time:** 2-3 hours

- ☐ **Create app in App Store Connect**
  - Go to [appstoreconnect.apple.com](https://appstoreconnect.apple.com)
  - Click + to create new app
  - Fill in basic info (name, bundle ID, SKU)

- ☐ **Build for TestFlight**

```
eas build --platform ios --profile production
```

- ☐ **Upload to App Store Connect**
  - EAS can do this automatically with credentials
  - Or manual upload via Transporter app
- ☐ **Submit for TestFlight review** (usually 1-2 days)
  - Fill in beta review notes
  - Add test information

**Deliverable:** App submitted to TestFlight

---

## Day 11: Internal Testing

**Time:** 2-3 hours

- ☐ **Test the TestFlight build yourself**
  - Install via TestFlight on your iPhone
  - Run through complete user flow
  - Document any issues
- ☐ **Invite 3-5 friends/family to test**
  - Add their emails in App Store Connect
  - They get TestFlight invite
  - Ask them to use it for 2-3 days
- ☐ **Create feedback form**
  - Google Form or Typeform
  - Questions: What's confusing? What's missing? Would you use this daily?
  - Share with testers

**Deliverable:** 3-5 people testing on real devices

---

## Day 12: Google Play Console Setup

**Time:** 3-4 hours

- ☐ **Create app in Google Play Console**
  - Go to [play.google.com/console](https://play.google.com/console)
  - Create new app
  - Fill in app details
- ☐ **Upload first build (Internal Testing)**

```
eas build --platform android --profile production
```

- Upload AAB file to Play Console
  - Set up internal testing track
- ☐ **Configure store listing**
  - Upload screenshots
  - Add app description
  - Add icon and feature graphic
  - Fill in categorization
  - Set content rating (likely "Everyone")

**Deliverable:** Android app in internal testing

---

## Day 13: Gather & Implement Feedback

**Time:** 4-5 hours

- ☐ **Review tester feedback**
  - What are the top 3 complaints?
  - Any critical bugs?
  - Any confusing UX?
- ☐ **Prioritize fixes**
  - Critical bugs: Must fix before launch
  - UX improvements: Fix if quick (< 2 hours)
  - Feature requests: Add to "post-launch" list
- ☐ **Implement critical fixes**
  - Don't get distracted by "nice-to-haves"
  - Focus on what blocks a good first impression

**Deliverable:** Updated app with critical feedback implemented

---

## Day 14: App Store Preparation - Final Details

**Time:** 3-4 hours

- ☐ **Write compelling app descriptions**
  - **Headline (iOS subtitle):** "Daily Gratitude & Self-Transcendence"
  - **Description:** Focus on transformation, not features
  - Include keywords naturally
  - Tell a story: Before (scattered) → After (aligned)
- ☐ **Prepare app preview video** (optional but powerful)
  - 15-30 second screen recording
  - Show onboarding → journal entry → view result
  - Add text overlays and music
  - Tool: CapCut (free, easy)
- ☐ **Fill in all metadata**
  - Keywords (iOS): gratitude, journal, affirmations, goals, habits
  - Category: Health & Fitness or Lifestyle
  - Age rating: 4+
  - Support URL (can be your email for now)
  - Marketing URL (optional)

**Deliverable:** All app store metadata complete

---

## Week 3: Submit for Review (Days 15-21)

### Day 15: Final Build - iOS

**Time:** 2-3 hours

- ☐ **Create production build**
- ```
eas build --platform ios --profile production
```
- ☐ **Test the production build thoroughly**
    - Install via TestFlight
    - Go through every screen
    - Test all features one final time
    - Check for any last-minute issues
  - ☐ **Bump version number**
    - Update to 1.0.0 in app.json
    - Rebuild if you made any changes



**Deliverable:** Final iOS production build ready

---

## Day 16: Submit iOS App for Review

**Time:** 2-3 hours

- ☐ **Submit to App Store Review**
  - In App Store Connect, click "Submit for Review"
  - Fill in App Review Information
  - Add demo account if needed (not needed for Radiant)
  - Add notes for reviewer explaining the app
- ☐ **Review submission checklist**
  - All screenshots uploaded
  - Description complete
  - Privacy policy linked
  - Age rating set
  - Pricing set (Free for now)
- ☐ **Wait for review** (typically 24-48 hours)
  - Apple will email you when it's "In Review"
  - Then when it's "Approved" or "Rejected"

**Deliverable:** iOS app submitted for review

---

## Day 17: Final Build - Android

**Time:** 2-3 hours

- ☐ **Create production build**

```
eas build --platform android --profile production
```

- ☐ **Test production build**
  - Install AAB on test device
  - Complete user flow test
  - Check performance
- ☐ **Upload to Play Console**
  - Go to Production track (or Open Testing if you want beta first)
  - Upload AAB file
  - Fill in release notes

**Deliverable:** Final Android production build uploaded

---

## Day 18: Submit Android App for Review

**Time:** 2-3 hours

- ☐ **Complete Play Console content rating**
  - Answer questionnaire honestly
  - Will likely be "Everyone" or "Everyone 10+"
- ☐ **Fill in pricing & distribution**
  - Free app
  - Available countries (all or select)
  - Distributed on Google Play
- ☐ **Submit for review**
  - Click "Send for Review"
  - Google review typically faster (few hours to 1 day)

**Deliverable:** Android app submitted for review

---

## Day 19-21: Review Period & Preparation for Launch

**Time:** 2-3 hours/day

While waiting for app review approvals:

- ☐ **Prepare launch announcement**
  - Tweet draft
  - Reddit post draft (r/SideProject, r/IndieDev)
  - LinkedIn post
  - Personal network email
- ☐ **Create simple landing page** (optional)
  - Single page explaining the app
  - Screenshots
  - App Store badges
  - Can use Carrd.co (free, super easy)
- ☐ **Plan your launch strategy**
  - Who will you tell first?
  - What communities might care?
  - Friends/family launch party?
- ☐ **Monitor review status**
  - Check App Store Connect daily

- Check Play Console daily
- Respond quickly if reviewers ask questions

**Common rejection reasons:**

- Missing privacy policy (you have this!)
- Crash on launch (you tested this!)
- Incomplete functionality (you built a complete MVP!)
- Misleading screenshots (be honest!)

**If rejected:** Don't panic. Read feedback, fix issue, resubmit. Usually approved on 2nd try.

**Deliverable:** Launch content ready, apps in review

---

## Week 4: Launch & Iterate (Days 22-30)

### Day 22: App Approval & Release

**Time:** 2-4 hours

- ☐ **If approved - Release to App Store**
  - iOS: Click "Release this Version" in App Store Connect
  - Android: Usually auto-released after approval
- ☐ **Verify apps are live**
  - Search for "Radiant" in App Store
  - Search in Google Play Store
  - Install on your own device from the store
- ☐ **Test the public version**
  - Delete your dev version
  - Download from store
  - Make sure everything works!

**Deliverable:** Radiant live on both app stores!

---

### Day 23: Launch Day!

**Time:** 3-4 hours

- ☐ **Announce to personal network**
  - Text/email close friends and family
  - Post on social media (Instagram, Twitter, LinkedIn)
  - Share in relevant communities
- ☐ **Post on Reddit** (carefully - follow subreddit rules)

- r/SideProject (great for indie makers)
  - r/IndieDev
  - r/Apps
  - r/Productivity
  - Be genuine, don't spam
- ☐ **Product Hunt launch** (optional)
  - Create account on producthunt.com
  - Submit your app
  - Engage with comments throughout the day
- ☐ **Share your journey**
  - Write a post about building it
  - "I built my first app with AI in 30 days"
  - People love these stories!

**Deliverable:** 50-100 people know about your app

---

## Day 24-25: Monitor & Support Early Users

**Time:** 2-3 hours/day

- ☐ **Check reviews daily**
  - App Store reviews
  - Google Play reviews
  - Respond to every review (good or bad)
- ☐ **Monitor crash reports**
  - Check App Store Connect analytics
  - Check Google Play Console vitals
  - Fix any critical crashes immediately
- ☐ **Gather user feedback**
  - DM people who downloaded it
  - Ask: "What's missing? What's confusing?"
  - Take notes for next update
- ☐ **Track metrics**
  - Downloads per day
  - Active users (from store analytics)
  - Review ratings

**Deliverable:** Understanding of how users are using the app

---

## Day 26-27: First Update Planning

**Time:** 3-4 hours

- ☐ **Prioritize user feedback**
  - What are the top 3 requests?
  - What bugs are people hitting?
  - What features would increase retention?
- ☐ **Plan version 1.1**
  - Pick 2-3 small improvements
  - Don't over-commit
  - Focus on polish, not new features
- ☐ **Create update roadmap**
  - Version 1.1: Bug fixes + 1-2 small features
  - Version 1.2: Next bigger feature
  - Keep it simple and achievable

**Deliverable:** Clear plan for next update

---

## Day 28-29: Implement Quick Wins

**Time:** 4-5 hours

- ☐ **Fix reported bugs**
  - Prioritize crashes and blocking issues
  - Test thoroughly
- ☐ **Add one highly-requested feature**
  - Something quick (2-3 hours max)
  - Shows users you're listening
  - Examples: Export journal, dark mode toggle, custom colors
- ☐ **Improve onboarding**
  - If users are dropping off, make it clearer
  - Add tooltips or hints
  - Test with a new user

**Deliverable:** Version 1.1 ready to ship

---

## Day 30: Reflection & Next Steps

**Time:** 2-3 hours

- ☐ **Submit version 1.1 update**
  - Build new version
  - Upload to both stores
  - Write clear release notes
- ☐ **Review your metrics**
  - Total downloads
  - Active users
  - Review rating (aim for 4.0+)
  - Retention (how many come back?)
- ☐ **Document lessons learned**
  - What went well?
  - What would you do differently?
  - What surprised you?
- ☐ **Plan for next 30 days**
  - Marketing strategy
  - Feature roadmap
  - Monetization exploration (if appropriate)
- ☐ **Celebrate!**
  - You shipped an app to production
  - Most people never get this far
  - This is a huge accomplishment!

**Deliverable:** Radiant v1.0 shipped, v1.1 in review, clear plan for growth

---

## Success Metrics (What "Shipped" Means)

By Day 30, you should have:

### ☒ **Technical:**

- App live on iOS App Store
- App live on Google Play Store
- No critical bugs
- 4.0+ star rating (if you have reviews)

### ☒ **Traction:**

- 50-100 downloads minimum
- 10-20 active users
- At least 3 reviews or feedback comments
- Understanding of who your users are

### ☑ **Process:**

- Know how to build, test, and deploy updates
- Comfortable with App Store Connect and Play Console
- Established update cadence (every 2-3 weeks)

### ☑ **Learning:**

- Understand your code (even if AI wrote it)
- Can debug common issues
- Know what users want vs what you built

---

## Common Roadblocks & Solutions

"I found a critical bug on Day 20"

**Solution:** Fix it immediately. Delay submission by 1-2 days. Better to ship late than ship broken.

"Apple rejected my app"

**Solution:** Read their feedback carefully. Usually fixable in 1 day. Resubmit. Don't give up.

"I don't know how to design screenshots"

**Solution:** Use Canva templates. Search "app store screenshots". Copy styles you like.

"No one downloads my app"

**Solution:** That's normal for Day 1. Focus on 10 engaged users, not 1000 downloads. Quality > quantity early on.

"I'm overwhelmed"

**Solution:** Pick the 3 most critical tasks each day. Skip optional items. Shipping > perfection.

---

## Daily Time Commitment

**Weekdays:** 2-4 hours/day **Weekends:** 4-6 hours/day

**Total:** ~90-100 hours over 30 days

If you have less time:

- Extend to 45-day plan
- Focus on iOS only first (skip Android until Day 35-45)
- Cut optional tasks (video preview, landing page)

---

## Emergency Contacts & Resources

**If stuck on build issues:**

- Expo forums: [forums.expo.dev](https://forums.expo.dev)
- Expo Discord: [chat.expo.dev](https://chat.expo.dev)

**If stuck on App Store:**

- Apple Developer Forums
- App Store Review Guidelines: [developer.apple.com/app-store/review/guidelines/](https://developer.apple.com/app-store/review/guidelines/)

**If stuck on Google Play:**

- Play Console Help Center
- Android Developers Slack

**If stuck on code:**

- Ask Claude Code (me!)
  - Stack Overflow
  - Reddit [r/reactnative](https://www.reddit.com/r/reactnative)
- 

## The Most Important Rule

**SHIP ON DAY 30. NO MATTER WHAT.**

- Even if it's not perfect
- Even if you want to add one more feature
- Even if the icon isn't quite right

**Done is better than perfect.**

You can always update. You can't learn from users until you ship.

---

## Post-Launch (Day 31+)

After shipping:

**Week 5-8: Establish rhythm**

- Update every 2-3 weeks
- Respond to all reviews
- Post weekly about your progress
- Aim for 200-500 downloads

**Week 9-12: Add monetization**

- Once you have 100+ active users
- Add premium features (\$3-5/month)
- Test what people will pay for

**Week 13-24: Grow to \$1K/month**

- Focus on user acquisition



- Improve retention
- Build sustainable business

But that's for another plan. For now:

**Focus on Day 1. Then Day 2. Then Day 3.**

One day at a time. You've got this.

---

*Created: December 24, 2025 For: Radiant App - Journey from MVP to App Stores Let's ship this! 🚀*