

Radiant App - App Store Readiness Checklist

Last Updated: December 28, 2025 Current Version: 1.0.0 Status: Pre-launch Development

CRITICAL - Must Have Before Submission

1. App Identity & Metadata

- **Bundle Identifier** - Add to app.json
 - iOS: "bundleIdentifier": "com.yourcompany.radiant"
 - Android: "package": "com.yourcompany.radiant"
 - **Example:** com.radiantapp.ios or com.yourstartup.radiant
 - **Proper App Name**
 - Change from "radiant-app" to "Radiant" in app.json
 - Update "name": "Radiant" and "slug": "radiant"
 - **App Description**
 - Write compelling description for store listing (170 characters for short, 4000 for full)
 - **Suggested Short:** "Transform your mindset with daily gratitude, affirmations, and vision boards. Build streaks and become who you want to be."
 - **Keywords**
 - Research and add relevant keywords
 - **Suggested:** gratitude, affirmations, journal, self-care, mindfulness, mental health, wellness, vision board, daily habits, personal growth
 - **Category**
 - Choose primary category
 - **Recommended:** Health & Fitness (primary) or Lifestyle (secondary)
-

2. Legal Requirements

Privacy Policy (REQUIRED)

- **Create Privacy Policy**
 - Must explain data collection:
 - Journal entries (stored locally)
 - Vision board images (stored locally)
 - Notification preferences (stored locally)
 - No data sent to servers (currently)
 - Must be hosted on a public URL
 - Add link to app.json: "privacyPolicy": "<https://yoursite.com/privacy>"

What to include:

- What data you collect (journal entries, images, settings)
- Where it's stored (locally on device via AsyncStorage)
- Who has access (only the user)
- Data deletion (uninstalling app deletes all data)
- Third-party services (none currently, but mention if you add analytics)

Terms of Service (Recommended)

- **Create Terms of Service**
 - User agreement for app usage
 - Liability limitations
 - Host on public URL

Support Contact

- **Add Support URL/Email**
 - `"supportUrl": "mailto:support@radianthead.com"` or website
 - Must be reachable and responsive
-

3. App Store Assets

App Icon

- **iOS App Icon**
 - 1024x1024 PNG
 - No transparency
 - No rounded corners (iOS adds them automatically)
 - Review current icon.png - may need professional design
- **Android Adaptive Icon**
 - Already have adaptive-icon.png
 - Verify it looks good on different launchers

Screenshots (5-10 required per platform)

iOS Requirements:

- iPhone 6.7" (iPhone 15 Pro Max): 1290x2796 pixels
- iPhone 5.5" (iPhone 8 Plus): 1242x2208 pixels
- Optional: iPad Pro 12.9": 2048x2732 pixels

Android Requirements:

- 16:9 ratio (e.g., 1080x1920 or higher)
- Minimum 2 screenshots, maximum 8

Screenshot Content (Show these screens):

1. Home screen with streak and greeting
2. Journal entry screen with gratitude prompts
3. Vision board grid with images
4. Daily check-in flow
5. Progress card showing insights
6. Settings screen with notifications
7. Slideshow feature (if you have images)

Tips:

- Add text overlays explaining features
- Use device frames to make them look professional
- Show the app in use, not just empty states

Feature Graphic (Android Only)

- **Create Feature Graphic**
 - 1024x500 PNG
 - Showcases app visually
 - Used at top of Play Store listing

4. App Configuration Updates**Current app.json needs these additions:**

```
{
  "expo": {
    "name": "Radiant", // Changed from "radiant-app"
    "slug": "radiant", // Changed from "radiant-app"
    "version": "1.0.0",
    "orientation": "portrait",
    "icon": "./assets/icon.png",
    "userInterfaceStyle": "light",
    "splash": {
      "image": "./assets/splash-icon.png",
      "resizeMode": "contain",
      "backgroundColor": "#ffffff"
    },
    "ios": {
      "bundleIdentifier": "com.yourcompany.radiant", // ADD THIS
      "buildNumber": "1", // ADD THIS
      "supportsTablet": true,
      "infoPlist": {
        "NSPhotoLibraryUsageDescription": "Radiant needs access to your photo library to add images to your vision board.",
        "NSCameraUsageDescription": "Radiant needs access to your camera to take photos for your vision board.",
        "NSUserNotificationsUsageDescription": "Radiant sends daily reminders to"
      }
    }
  }
}
```

```
help you maintain your gratitude practice and keep your streak alive."
        }
    },
    "android": {
        "package": "com.yourcompany.radiant",           // ADD THIS
        "versionCode": "1",                            // ADD THIS
        "adaptiveIcon": {
            "foregroundImage": "./assets/adaptive-icon.png",
            "backgroundColor": "#ffffff"
        },
        "permissions": [
            "CAMERA",
            "READ_EXTERNAL_STORAGE",
            "WRITE_EXTERNAL_STORAGE",
            "POST_NOTIFICATIONS"
        ]
    },
    "plugins": [
        [
            "expo-notifications",
            {
                "icon": "./assets/icon.png",
                "color": "#FF9A76",
                "sounds": []
            }
        ]
    ],
    "extra": {
        "eas": {
            "projectId": "YOUR_PROJECT_ID" // ADD THIS after running eas
        }
    }
}
```

5. Build Configuration

EAS Build Setup

- **Install EAS CLI**

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

- **Login to Expo**

```
eas login
```

- **Configure EAS Build**

```
eas build:configure
```

- This creates `eas.json` file
- Adds projectId to app.json

- **Create Production Build Profile**

- Ensure `eas.json` has production configuration
- Development vs Production builds are different

App Signing

iOS:

- **Apple Developer Account** (\$99/year)
 - Sign up at <https://developer.apple.com>
 - Required for App Store submission
- **Create App Identifier**
 - Must match bundleIdentifier in app.json
- **Generate Certificates** (EAS handles this automatically)
 - Distribution certificate
 - Provisioning profile

Android:

- **Generate Keystore File**
 - EAS can generate and manage this for you
 - OR create manually and upload to EAS
- **Keep Keystore Secure**
 - If you lose it, you can't update your app
 - EAS stores it securely in the cloud

6. Testing Requirements

Device Testing

- **Test on Real iOS Devices**
 - iPhone SE (small screen)
 - iPhone 14/15 Pro (mid-size)

- iPhone 15 Pro Max (large screen)
 - iPad (if supporting tablets)
- **Test on Real Android Devices**
 - Samsung Galaxy (popular)
 - Google Pixel (stock Android)
 - Different Android versions (11, 12, 13, 14)
- ## Feature Testing
- **Journal Features**
 - Create journal entry
 - Edit journal entry
 - Delete journal entry
 - View all entries
 - Data persists after app close
 - **Vision Board Features**
 - Upload image from library
 - Take photo with camera
 - Add label to image
 - Delete image
 - View slideshow
 - Slideshow disabled when no images
 - **Daily Check-In**
 - Complete daily check-in
 - Check-in once per day limit
 - Updates streak counter
 - **Streak Tracking**
 - Streak increases with daily check-ins
 - Streak resets if day missed
 - Displays correctly on home screen
 - **Progress Card**
 - Shows weekly insights
 - Displays entry count
 - Calculates longest streak
 - **Notifications (UNTESTED)**
 - Morning notification sends at correct time
 - Evening notification sends at correct time
 - Streak protection notification works

- Notifications can be toggled on/off
 - Time picker changes notification times
 - Test notification button works
- **Settings**
 - Toggle notifications master switch
 - Toggle individual notification types
 - Change notification times
 - Settings persist after app close

Edge Cases & Error Handling

- **Permissions**
 - Camera permission denied - shows alert
 - Photo library permission denied - shows alert
 - Notification permission denied - shows alert
 - User can still use app without permissions
- **Offline Functionality**
 - App works completely offline (uses AsyncStorage)
 - No network errors when offline
- **Data Limits**
 - Test with 100+ journal entries
 - Test with 50 vision board images
 - App doesn't slow down with lots of data
- **Memory Management**
 - App doesn't crash on low memory devices
 - Images are properly compressed
 - No memory leaks during extended use
- **UI Edge Cases**
 - Very long journal entries display correctly
 - Very long affirmations display correctly
 - App looks good on small screens (iPhone SE)
 - App looks good on large screens (iPhone Pro Max)
 - Keyboard doesn't hide input fields
 - Safe area insets work on all devices

7. App Store Connect / Google Play Console Setup

Apple App Store

- **Create Apple Developer Account**
 - Cost: \$99/year
 - Sign up at <https://developer.apple.com>
 - Enrollment takes 1-2 days
- **Create App in App Store Connect**
 - Go to <https://appstoreconnect.apple.com>
 - Click "My Apps" → "+" → "New App"
 - Fill in app information:
 - Name: Radiant
 - Primary Language: English
 - Bundle ID: (select the one you created)
 - SKU: radiant-ios-app-001
- **Set App Information**
 - Category: Health & Fitness
 - Age Rating: 4+ (no objectionable content)
 - Privacy Policy URL
 - Support URL
- **Prepare for App Review**
 - Review Notes: Explain the app to reviewers
 - Demo Account: Not needed (no login required)
 - Contact Information

Google Play Console

- **Create Google Play Console Account**
 - Cost: \$25 one-time fee
 - Sign up at <https://play.google.com/console>
 - Verification takes 1-2 days
- **Create App in Play Console**
 - Click "Create app"
 - Fill in app details:
 - App name: Radiant
 - Default language: English (United States)
 - App or game: App
 - Free or paid: Free
- **Set Up Store Listing**
 - Short description (80 characters)
 - Full description (4000 characters)
 - Screenshots (minimum 2, maximum 8)

- Feature graphic (1024x500)
 - App icon (512x512)
 - **Content Rating**
 - Complete questionnaire
 - Expected rating: Everyone
 - **Privacy Policy**
 - Add privacy policy URL (required)
-

HIGHLY RECOMMENDED - Should Have

8. User Experience Improvements

Onboarding Flow

- **Create Welcome Screen**
 - Show app value proposition
 - "Welcome to Radiant - Your daily companion for gratitude and growth"
- **Feature Tour**
 - Slide 1: Self-Transcendence Journal
 - Slide 2: Vision Board
 - Slide 3: Daily Check-ins & Streaks
 - Slide 4: Affirmations & Reminders
- **Guided First Experience**
 - Prompt first journal entry
 - Help add first vision board image
 - Show how check-in works
 - Explain streak system
- **Permission Requests at Right Time**
 - Don't ask for all permissions upfront
 - Ask for photo permission when user taps vision board
 - Ask for notification permission after explaining value

Better Empty States

- **Journal Empty State**
 - Current: Basic text
 - Improved: Illustration + motivational copy + clear CTA
- **Vision Board Empty State**

- Current: Basic emoji + text
- Improved: Show example vision board + explain benefits
- **Progress Card Empty State**
 - Show placeholder with "Complete 7 check-ins to unlock insights"

Loading States

- **Image Upload Loading**
 - Show spinner while image picker opens
 - Show progress when saving image
- **Data Saving Feedback**
 - Toast/notification when journal entry saved
 - Visual feedback when check-in completed

Error Handling

- **Network Errors**
 - Not applicable currently (offline-first)
- **Storage Errors**
 - Handle AsyncStorage failures gracefully
 - Show user-friendly message: "Unable to save data. Please ensure you have storage space available."
- **Permission Errors**
 - Better messaging: "We need camera access to take photos for your vision board. You can grant this in Settings."
 - Button to open Settings app

9. Analytics & Monitoring

Crash Reporting

- **Set Up Sentry**
 - Install: `npx expo install @sentry/react-native`
 - Configure with DSN
 - Get alerts when app crashes
 - See stack traces and error context
- **Alternative: Bugsnag**
 - Similar to Sentry

- Good React Native support

Analytics

- **Set Up Mixpanel or Amplitude**
 - Track key events:
 - App opened
 - Journal entry created
 - Vision board image added
 - Daily check-in completed
 - Notification settings changed
 - Slideshow viewed
- **Track User Flows**
 - Onboarding completion rate
 - Feature usage rates
 - Drop-off points
- **Retention Metrics**
 - Day 1 retention
 - Day 7 retention
 - Day 30 retention

Performance Monitoring

- **Firebase Performance Monitoring**
 - Track app startup time
 - Monitor screen load times
 - Identify slow operations
- **Set Performance Budgets**
 - App startup: < 2 seconds
 - Screen transitions: < 300ms
 - Image loading: < 1 second

10. Code Quality

Clean Up Code

- **Remove Console Logs**
 - Search for `console.log` and remove/replace with proper logging
- **Fix TypeScript Errors**
 - Run: `npx tsc --noEmit`

- Fix all type errors
- **Remove Dead Code**
 - Remove unused imports
 - Delete commented-out code
 - Remove unused files
- **Code Formatting**
 - Run Prettier: `npx prettier --write .`
 - Ensure consistent code style

Version Management

- **Ensure Version Consistency**
 - app.json version: 1.0.0
 - package.json version: 1.0.0
 - iOS buildNumber: 1
 - Android versionCode: 1

Documentation

- **Update README**
 - App description
 - Setup instructions
 - Build instructions
- **Code Comments**
 - Document complex logic
 - Explain non-obvious decisions

NICE TO HAVE - Before Launch

11. Additional Features

Data Export

- **Implement Export to PDF**
 - Export all journal entries
 - Include vision board images
 - Formatted nicely for printing
 - **Priority:** High (mentioned in monetization as must-have for paid version)

Dark Mode

- **Add Dark Mode Toggle**
 - Settings screen option

- Store preference in AsyncStorage
- Update all screens to support dark theme
- **Priority:** Medium (users expect this in 2025)

Share Feature

- **Share Affirmations**
 - Generate shareable image with affirmation
 - Share to social media
- **Share Vision Board**
 - Create collage of vision board images
 - Share to Instagram/Facebook

Help & Support

- **Add Help/FAQ Section**
 - "How do streaks work?"
 - "How do I change notification times?"
 - "Can I export my journal?"
 - "Is my data private?"
- **In-App Feedback**
 - Contact support button
 - Bug report feature
 - Feature request form

Engagement Features

- **Rate the App Prompt**
 - Show after 7-day streak
 - Don't show again if dismissed
 - Link to App Store/Play Store review page
- **Achievement Badges**
 - 7-day streak badge
 - 30-day streak badge
 - 100-day streak badge
 - First journal entry badge
 - 10 vision board images badge

12. Performance Optimization

Image Optimization

- **Compress Vision Board Images**
 - Reduce quality to 0.7 (currently 0.8)
 - Resize to max 1024x1024
 - Saves storage space
- **Compress App Assets**
 - Optimize icon.png
 - Optimize splash-icon.png
 - Use WebP format where possible

Bundle Size Reduction

- **Analyze Bundle Size**
 - Run: `npx expo export --dump-sourcemap`
 - Identify large dependencies
- **Remove Unused Dependencies**
 - Check package.json for unused packages
 - Remove if not needed

Performance Testing

- **Test on Older Devices**
 - iPhone 8 (2017)
 - Samsung Galaxy S10 (2019)
 - Ensure smooth 60fps animations
- **Test with Large Datasets**
 - 500+ journal entries
 - 50 vision board images
 - Measure app responsiveness

TIMELINE ESTIMATE

Week 1: Critical Setup (5-7 days)

Goal: Get all legal and identity requirements done

- **Day 1-2:**
 - Register Apple Developer Account (\$99)
 - Register Google Play Console (\$25)
 - Choose and add bundle identifier to app.json
 - Update app name in app.json

- **Day 3-4:**

- Write privacy policy (use template)
- Host privacy policy on website or GitHub Pages
- Write terms of service
- Create support email

- **Day 5-7:**

- Create app screenshots (use simulator + device frames)
 - Review/update app icon
 - Write app description and keywords
 - Create feature graphic for Android
-

Week 2: Testing & Polish (5-7 days)

Goal: Build production version and test thoroughly

- **Day 1-2:**

- Install EAS CLI
- Configure EAS build (`eas build:configure`)
- Create first iOS production build
- Create first Android production build

- **Day 3-5:**

- Install builds on real devices
- Test ALL features on iOS
- Test ALL features on Android
- **CRITICAL:** Test notifications on real devices
- Document and fix bugs found

- **Day 6-7:**

- Implement basic onboarding flow
 - Add loading states
 - Improve error messages
 - Set up crash reporting (Sentry)
-

Week 3: Store Setup & Beta Testing (5-7 days)

Goal: Get app ready for beta testers

- **Day 1-2:**

- Create app in App Store Connect
- Create app in Google Play Console
- Fill in all store listing information
- Upload screenshots and assets

- **Day 3-4:**

- Upload build to TestFlight (iOS)
- Upload build to Internal Testing (Android)
- Invite 5-10 beta testers (friends/family)

- **Day 5-7:**

- Collect beta feedback
 - Fix critical bugs
 - Make UI improvements based on feedback
 - Create new build with fixes
-

Week 4: Final Review & Submission (5-7 days)

Goal: Submit to stores

- **Day 1-3:**

- Final testing on all devices
- Verify all store assets are perfect
- Double-check privacy policy and support email
- Run TypeScript check (`npx tsc --noEmit`)
- Remove all console.logs

- **Day 4:**

- Create final production build
- Upload to App Store Connect
- Submit for App Store review
- Upload to Google Play Console
- Submit for Google Play review

- **Day 5-7:**

- Monitor review status
 - Respond to any review questions
 - **iOS:** Expect 3-7 day review time
 - **Android:** Expect 1-3 day review time
-

Total Timeline: 4 weeks minimum (if working full-time)

If working part-time (nights/weekends): **6-8 weeks**

BIGGEST BLOCKERS RIGHT NOW

🚫 Critical Blockers (Can't Submit Without These)

1. **No Bundle Identifier**

- Impact: Can't build for production
- Fix: Add to app.json (5 minutes)
- Priority: **CRITICAL**

2. No Privacy Policy

- Impact: Instant rejection from both stores
- Fix: Write policy + host it (2-3 hours)
- Priority: **CRITICAL**

3. No Screenshots

- Impact: Can't complete store listing
- Fix: Take screenshots + add frames (3-4 hours)
- Priority: **CRITICAL**

4. No Developer Accounts

- Impact: Can't submit to stores
- Fix: Pay \$99 (Apple) + \$25 (Google)
- Priority: **CRITICAL**

5. Notifications Untested

- Impact: Might not work in production (major feature broken)
- Fix: Build dev client + test on real device (4-6 hours)
- Priority: **HIGH**

RECOMMENDED FIRST 5 TASKS

Start with these TODAY to unblock yourself:

Task 1: Register Developer Accounts

Time: 30 minutes **Cost:** \$124 total

1. Go to <https://developer.apple.com>
2. Enroll in Apple Developer Program (\$99/year)
3. Go to <https://play.google.com/console>
4. Create Google Play Developer account (\$25 one-time)

Why first: Takes 1-2 days for approval, do this early

Task 2: Add Bundle Identifier

Time: 5 minutes **Cost:** Free

Edit [app.json](#):

```
"ios": {  
  "bundleIdentifier": "com.radiantapp.ios",  
  "buildNumber": "1"  
},  
"android": {  
  "package": "com.radiantapp.android",  
  "versionCode": 1  
}
```

Why second: Required for all builds

Task 3: Write Privacy Policy

Time: 2-3 hours **Cost:** Free

Use template (I can help with this) and host on:

- GitHub Pages (free)
- Your website
- Privacy policy generator service

Why third: Required for store submission

Task 4: Set Up EAS Build

Time: 1-2 hours **Cost:** Free (Expo free tier)

```
npm install -g eas-cli  
eas login  
eas build:configure  
eas build --platform ios --profile production  
eas build --platform android --profile production
```

Why fourth: Need production build to test notifications

Task 5: Test Notifications

Time: 4-6 hours **Cost:** Free

1. Install production build on your iPhone
2. Install production build on your Android phone
3. Enable notifications in Settings
4. Set morning notification for 1 minute from now
5. Verify it arrives
6. Repeat for evening and streak protection
7. Fix any bugs found

Why fifth: Critical feature, must work before launch

SUCCESS CRITERIA

You're ready to submit when:

- App builds successfully for iOS and Android
 - All features work on real devices
 - Notifications send at correct times
 - Privacy policy is live and linked
 - Screenshots look professional
 - App Store Connect / Play Console listings complete
 - No critical bugs found in testing
 - TypeScript compiles without errors
 - App has been tested by at least 3 beta users
-

HELPFUL RESOURCES

App Store Submission

- Apple App Store Review Guidelines: <https://developer.apple.com/app-store/review/guidelines/>
- Google Play Policy Center: <https://play.google.com/about/developer-content-policy/>

Privacy Policy

- Privacy Policy Generator: <https://www.privacypolicygenerator.info/>
- Sample Privacy Policies: <https://www.termsfeed.com/blog/sample-privacy-policy-template/>

Screenshots

- Screenshot Frames: <https://www.screely.com/> or <https://mockuphone.com/>
- App Store Screenshot Sizes: <https://help.apple.com/app-store-connect/#/devd274dd925>

Build & Deploy

- Expo EAS Build Docs: <https://docs.expo.dev/build/introduction/>
- Expo Submit Docs: <https://docs.expo.dev/submit/introduction/>

Testing

- TestFlight (iOS): <https://developer.apple.com/testflight/>
 - Google Play Internal Testing: <https://support.google.com/googleplay/android-developer/answer/9845334>
-

NOTES

- **Version Numbering:** Start at 1.0.0, increment patch (1.0.1) for bug fixes, minor (1.1.0) for features
- **Review Times:** iOS takes 3-7 days, Android takes 1-3 days (plan accordingly)

- **Rejections:** Common reasons include missing privacy policy, unclear app purpose, crashes on reviewer device
 - **Updates:** After launch, plan for weekly bug fix releases for first month
 - **Marketing:** Have a plan for launch day (Product Hunt, social media, etc.)
-

Next Steps: Start with the "Recommended First 5 Tasks" and work through the checklist systematically.

Good luck with your launch! 