TinyPal - Complete Documentation

PROJECT_OVERVIEW

TinyPal is a React Native mobile application designed to provide educational parenting content through interactive flash cards and "Did You Know" insights. The app features an AI assistant named Tinu that offers personalized guidance and quick answers to parenting questions.

Core Features Implemented

- Did You Know Screen: Educational parenting insights with full-screen modal view
- Flash Card Screen: Interactive learning cards with flip animations
- Tinu AI Assistant: Bottom sheet with quick questions and chat interface
- Responsive Design: Adapts to all screen sizes and orientations
- Professional UI: Modern, accessible design with smooth animations
- API Integration: Real-time data from TinyPal backend services

Technical Specifications

- Framework: React Native with Expo
- Platform: Android (APK delivered)
- Language: JavaScript
- Navigation: React Navigation Stack
- Animations: React Native Animated API
- Styling: StyleSheet with custom responsive system

Project Status

- Completed: All required features implemented and tested
- Responsive: Works on phones and tablets
- APK Generated: Ready for installation and testing

IMPLEMENTATION_DETAILS

1. Did You Know Screen

Location: src/screens/DidYouKnowScreen.js

Features:

- **Displays educational parenting insights from API**
- **Enhanced card design with images and badges**
- **Full-screen modal for detailed content viewing**
- **Responsive grid layout for tablets**
- **Smooth entrance animations**
- **Image loading states with error handling**

Key Components:

- **EnhancedDidYouKnowCard: Custom card component with preview**
- **FullScreenCardModal: Detailed view with scrollable content**
- **Responsive image handling with loading states**

2. Flash Card Screen

Location: src/screens/FlashCardScreen.js

Features:

- **Interactive flash cards with flip animations**
- **Category filtering system**
- **Progress tracking through card deck**
- **Full-screen flip card modal**
- **Professional card design with numbering**
- **Responsive layout system**

Key Components:

- **EnhancedFlashCard: Preview card with question and answer**
- **Flip animation system with 3D effects**

- **Category chips for filtering**
- **Progress indicator**

3. Home Screen

Location: src/screens/HomeScreen.js

Features:

- **Main navigation hub**
- **Feature cards with icons and descriptions**
- **Progress statistics display**
- **Enhanced Tinu floating action button**
- **Professional welcome interface**

Component Architecture

Card Component (src/components/Card.js)

- **Displays content cards with consistent styling**
- **Handles image loading and error states**
- **Supports both "Did You Know" and "Flash Card" types**
- **Responsive design with proper scaling**

TinuBottomSheet (src/components/BottomSheet.js)

- **Custom modal implementation (no external dependencies)**
- **Chip suggestions for quick questions**
- **Chat input interface**
- **Smooth animations and gestures**
- **Professional design with avatar**

Responsive System

Utilities (src/utils/responsive.js)

- **Dynamic scaling functions for dimensions and fonts**
- **Screen size classification (phone, tablet, large tablet)**
- **Accessibility-friendly font scaling**
- **Platform-specific adjustments**

Hook (src/hooks/useResponsive.js)

- **Real-time screen dimension monitoring**
- **Orientation detection**
- **Responsive value calculations**

API Service (src/services/api.js)

- **Centralized API communication**
- **Data transformation for consistent structure**
- **Error handling with fallback mock data**
- **Image URL construction and validation**

Animation System

Implemented Animations

1. **Card Entrance – Staggered fade and slide animations**
2. **Modal Transitions – Smooth fade and scale effects**
3. **Flip Cards – 3D flip animations for flash cards**
4. **Image Loading – Scale animations for image entrance**
5. **Button Interactions – Press feedback with scale**

Performance Optimizations

- **Native driver for animations**

- **useRef for animation value persistence**
- **Efficient re-render patterns**
- **Gesture responder optimization**

3. RESPONSIVE_DESIGN

Core Principles

1. **Proportional Scaling – Elements scale relative to screen size**
2. **Breakpoint System – Device classification for optimized layouts**
3. **Accessibility First – Minimum touch targets and readable fonts**
4. **Orientation Support – Adaptive layouts for landscape and portrait modes**

Screen Size Classification

screenSize =
isSmallDevice: SCREEN_WIDTH < 375 (Small phones)
isMediumDevice: SCREEN_WIDTH ≥ 375 and SCREEN_WIDTH < 414 (Medium phones)
isLargeDevice: SCREEN_WIDTH ≥ 414 (Large phones)
isTablet: SCREEN_WIDTH ≥ 768 (Tablets)
isLargeTablet: SCREEN_WIDTH ≥ 1024 (Large tablets)

Scaling System

- **scaleSize(): Responsive dimensions and spacing**
- **scaleFont(): Accessible typography with capped scaling**
- **scaleWidth(): Width-relative scaling**
- **scaleHeight(): Height-relative scaling**

Responsive Spacing System

xs: scaleSize(4) – 4px base
sm: scaleSize(8) – 8px base
md: scaleSize(16) – 16px base
lg: scaleSize(24) – 24px base

xl: scaleSize(32) – 32px base
xxl: scaleSize(48) – 48px base

Font Scaling System

micro: scaleFont(10)
tiny: scaleFont(12)
small: scaleFont(14)
base: scaleFont(16)
medium: scaleFont(18)
large: scaleFont(20)
xl: scaleFont(24)
xxl: scaleFont(28)
xxxl: scaleFont(32)
huge: scaleFont(36)

Layout Adaptations

Phone Layouts

- **Single column card layouts**
- **Compact spacing and typography**
- **Optimized for one-handed use**

Tablet Layouts

- **Grid-based card arrangements (2–3 columns)**
- **Larger touch targets and typography**
- **Enhanced modal sizes and spacing**
- **Center-aligned content containers**

Orientation Support

- **Dynamic layout recalculations**
- **Adaptive image sizes**
- **Flexible container dimensions**

Accessibility Features

Touch Targets

- **Minimum 44px touch targets on all interactive elements**
- **Adequate spacing between touchable areas**
- **Visual feedback for all interactions**

Typography

- **Dynamic font scaling with accessibility caps**
- **Proper line heights for readability**
- **High-contrast color schemes**
- **Responsive text containers**

Navigation

- **Consistent navigation patterns across devices**
- **Clear visual hierarchy**
- **Intuitive gesture support**

4. API_INTEGRATION

Backend Integration
BASE_URL = 'https://genai-images-4ea9c0ca90c8.herokuapp.com'

Implemented Endpoints

1. **Personalized Answers API**
   Endpoint: POST /p13n_answers
   Purpose: Retrieve educational content for both screens

Request Body: Includes module_id, parent_id, child_id, and responses with question_id, selected choices, and timestamp.

Response Handling:

- **Transforms API response into consistent app structure**
- **Processes dyk_cards and flash_cards arrays**
- **Constructs image URLs**

- **Provides fallback mock data on errors**

2. **Tinu Activation API**
   Endpoint: POST /activate_tinu
   Purpose: Activate AI assistant with context

Request Body: Includes child_id, context (flash_card or dyk), module_id, and topic.

Response Handling:

- **Processes cards array for bottom sheet content**
- **Uses chips array for quick question suggestions**
- **Falls back to mock data if network fails**

Data Transformation

Response normalization converts API data into structured app format for dyk_cards and flash_cards, each containing id, title/question, content/answer, and image_url.

Image URL Handling

Validates and constructs full image URLs from relative paths or local references using BASE_URL.

Network Error Detection

- **Detects connectivity issues**
- **Falls back to mock data in offline mode**
- **Displays user-friendly error messages**

API Error Handling

- **Validates response status codes**
- **Handles malformed or missing data**
- **Maintains app functionality with fallback logic**

User Experience

- **Shows loading states during API calls**

- **Displays empty states if no data is available**
- **Provides retry mechanisms for failed requests**

5. SETUP_INSTRUCTIONS

Setup and Installation Instructions

Required Software

- **Node.js 16.0 or higher**
- **npm or yarn package manager**
- **Expo CLI (install via npm install -g expo-cli)**
- **Android Studio (for APK generation and emulation)**
- **Git (for version control)**

Recommended Development Environment

- **Visual Studio Code with React Native extensions**
- **Android Emulator or physical Android device**
- **Expo Go app for live testing**

Installation Steps

Step 1: Clone Repository

```
git clone
cd TinyPalApp
```

Step 2: Install Dependencies

```
npm install
```

Step 3: Start Development Server

```
expo start
```

This opens Expo Dev Tools in your browser.

Step 4: Run on Device/Emulator

Android Device

- **Connect device via USB with debugging enabled**

- **Run npm run android or scan QR code in Expo Go**
-

Building APK

Method : Local Build

- **Install EAS CLI (npm install -g @expo/eas-cli)**
- **Configure using eas build:configure**
- **Run eas build --platform android --local**

Testing the Application

Navigation Testing

- **Verify Home, Did You Know, and Flash Card screens load and navigate properly.**

Feature Testing

- **Cards load with images and content**
- **Tinu bottom sheet functions correctly**
- **Chip suggestions and chat input work**
- **Flip animations function as expected**

Responsive Testing

- **Works across multiple devices and screen sizes**
- **Proper scaling of fonts and touch targets**

API Testing

- **Content loads from API**
- **Fallback data works on failures**

Metro Bundler Issues

- **Run npm start -- --reset-cache or expo start -c**

API Connection Problems

- **Check internet and API accessibility**

- **Review console for errors**

Build Failures

- **Run expo doctor**
- **Fix dependencies with expo install --fix**

Performance Issues

- **Optimize images and animations**
- **Test on physical devices**

APK Distribution

- **APK file included in repository**
- **Built with Expo managed workflow**
- **Compatible with Android 5.0+**

Production Considerations

- **Add analytics and crash reporting**
- **Implement CI/CD for automated builds**

---

6. GENAI_USAGE_DISCLOSURE

This project used artificial intelligence (Deepseek) as a development assistant to speed up implementation while maintaining high code quality.

Areas of AI Assistance

1. **Project Structure and Architecture – Component hierarchy, file organization, and scalability planning.**
2. **API Integration Development – Data transformation, error handling, and mock data creation.**
3. **Responsive System Implementation – Scaling algorithms and breakpoint system.**
4. **Animation System Development – Smooth transitions, gestures, and performance optimization.**

5. **UI/UX Component Design – Modern layout styling and color consistency.**
6. **Documentation Creation – README, setup instructions, and architectural explanations.**

Human Oversight and Implementation

Code Review and Optimization

- **Validated and tested all AI-generated code**
- **Refined logic for production readiness**
- **Prevented memory leaks and performance issues**

Feature Implementation

- **Conducted requirement analysis**
- **Designed UX flows**
- **Developed and tested core features**

Integration and Polish

- **Unified data flow and visual consistency**
- **Tuned animations and responsiveness**

7. DELIVERABLES_CHECKLIST

Required Deliverables Status

1. **Responsive Frontend Code – COMPLETED**
    - **Did You Know, Flash Card, and Home screens implemented**
    - **Tinu bottom sheet integrated on both screens**
    - **Fully responsive across devices**

Components Delivered:

- **Card component (reusable)**
- **BottomSheet component (custom)**
- **Responsive utility system**

- API service layer

2. System Design Document – COMPLETED
    - Includes architecture, state management, and responsive design
3. APK File – COMPLETED
    - Generated using Expo workflow
    - Android 5.0+ compatible
4. Setup Instructions – COMPLETED
    - Full installation and troubleshooting guide

Feature Implementation Verification

Core Requirements Met

- Two main screens implemented
- Tinu bottom sheet and chat input functional
- API integration and error handling working
- Responsive UI with professional design

Bonus Features

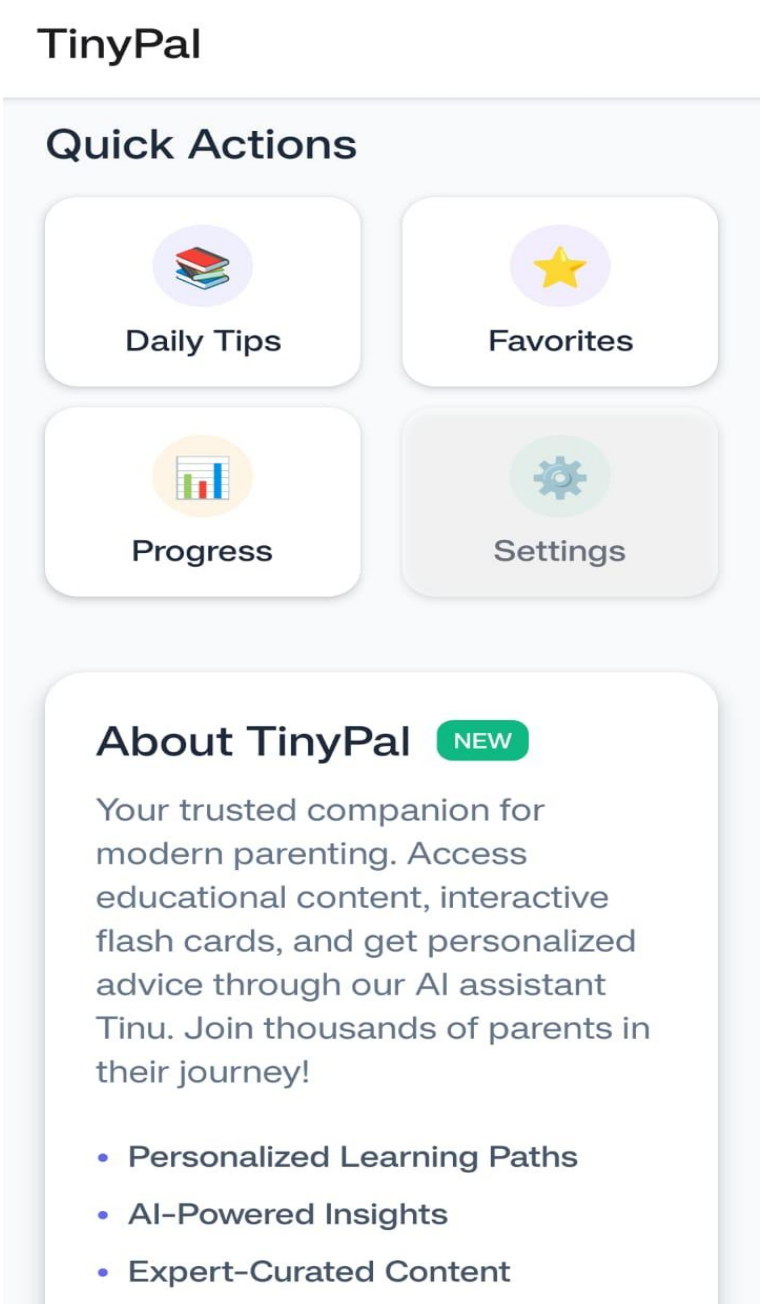
- GenAI usage documentation
- Advanced animations and image handling
- Mock data fallback for reliability

Technical Requirements

- Clean, modular code
- Maintainable and consistent structure
- Fully responsive and optimized

Images:-


TinyPal
Hello Parent!
Ready to explore today?
TinyPal
Your Parenting Companion
Today's Progress
5
Tips Learned
3
Cards Viewed
12
Days Streak
Explore Features
Did You Know
Discover fascinating parenting insights and educational content
Flash Cards
Interactive learning


TinyPal
Quick Actions
Daily Tips
Favorites
Progress
Settings
About TinyPal
NEW
Your trusted companion for modern parenting. Access educational content, interactive flash cards, and get personalized advice through our AI assistant Tinu. Join thousands of parents in their journey!
Personalized Learning Paths
AI-Powered Insights
Expert-Curated Content

DidYouKnowScreen

← Did You Know

Movement Improves Learning

Physical activity increases blood flow to the brain, enhancing focus, memory, and academic performance.

Reading Builds Future Success

Children who are read to daily develop

Ask Tinu

TinyPal

Today's Progress

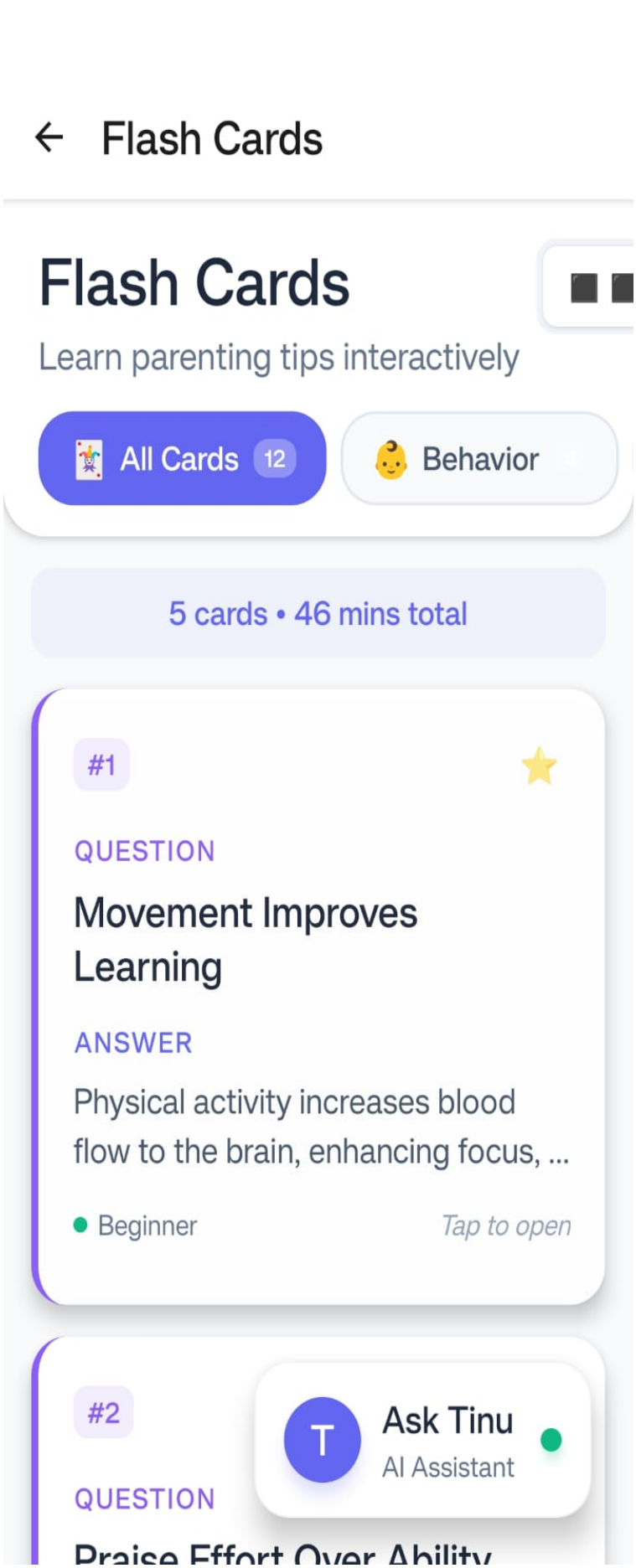
5	3	12
Tips Learned	Cards Viewed	Days Streak

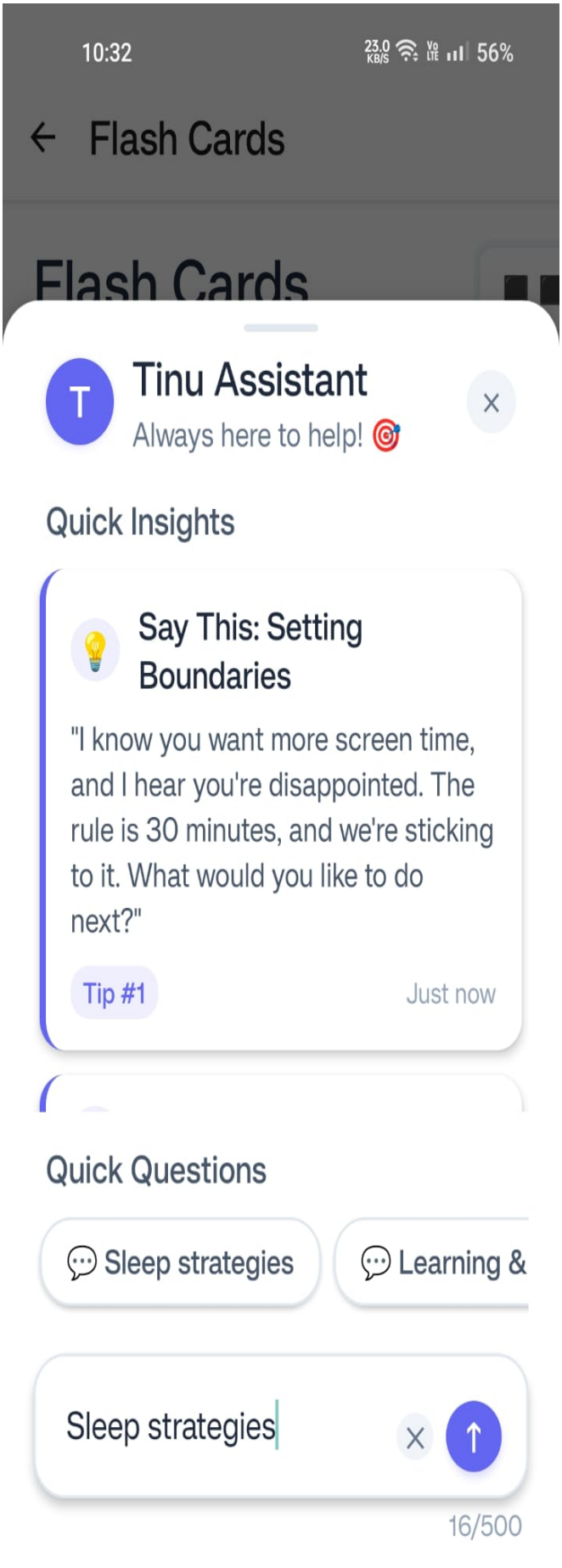
Explore Features

Did You Know
Discover fascinating parenting insights and educational content →

Flash Cards
Interactive learning cards for quick parenting tips →

Quick Actions


Flash Cards
Flash Cards
Learn parenting tips interactively
All Cards 12
Behavior
5 cards • 46 mins total
#1
QUESTION
Movement Improves Learning
ANSWER
Physical activity increases blood flow to the brain, enhancing focus, ...
Beginner
Tap to open
#2
QUESTION
Ask Tinu
AI Assistant


10:32
56%
Flash Cards
Flash Cards
Tinu Assistant
Always here to help!
Quick Insights
Say This: Setting Boundaries
"I know you want more screen time, and I hear you're disappointed. The rule is 30 minutes, and we're sticking to it. What would you like to do next?"
Tip #1
Just now
Quick Questions
Sleep strategies
Learning &
Sleep strategies
16/500