BudgetBee: Complete Product Documentation

1. High-Level Product Summary

BudgetBee is a conversational AI budgeting assistant designed to prevent overspending among young people (students, early professionals) by addressing four core spending triggers:

- **Cognitive overload spending** from mental fatigue
- **Social-media & influencer triggered spending** (FOMO)
- **Emotional impulsive spending** (stress, sadness, excitement)
- **Social comparison spending** ("everyone has it" pressure)

Core Mechanism: Users manually open ChatBee before making a purchase. The AI analyzes their emotional state, cognitive load, social triggers, and spending patterns to provide real-time chat/voice guidance that prevents bad purchases.

Key Features:

1. Cognitive Load Alerts
2. Social Trigger Detection
3. Emotional Spending Detection
4. Social Comparison Interception
5. Honey Pot (savings jar for skipped purchases)
6. ChatBee – AI conversational assistant with voice + text (Agora-like mock)

---

2. Full UX/UI Breakdown — Every Screen

Screen 1: Onboarding — Budget Setup

Purpose: User sets up their budget with ChatBee's guidance

UI Elements:

- Welcome message from ChatBee (bee avatar + speech bubble)
- Conversational prompts asking:
  - "What's your monthly income?"
  - "What are your essential expenses? (rent, food, utilities)"
  - "How much do you want to save each month?"
  - "What are your spending goals?"

- Input fields that appear conversationally (one at a time)
- "Next" button after each answer
- Progress indicator (bee flying across screen, 4 stops)

Outcome: Budget categories established, savings goal set

Screen 2: Home Dashboard

Purpose: Overview of budget status and quick access to ChatBee

UI Elements:

- Header: "Hi [Name]! 🐝" + current date
- Budget Summary Card (glassmorphic):
  - Remaining budget this month (large number, honey-gold)
  - Progress bar (spent vs. remaining)
  - Days left in month
- Honey Pot Card (glassmorphic):
  - Total saved from skipped purchases
  - Animated honey filling jar visual
  - Tap to see savings breakdown
- Big prominent button: "Talk to ChatBee" (honey-gold, bee icon)
- Bottom nav: Home | Honey Pot | Settings

Interactions:

- Pull to refresh updates budget
- Honey Pot card pulses when new savings added
- ChatBee button has subtle bounce animation

Screen 3: ChatBee Interface (Text Mode)

Purpose: Conversational AI assistant for pre-purchase guidance

UI Elements:

- Header: "ChatBee 🐝" + voice toggle button (right corner)
- Chat interface:

  - User messages (right, white bubbles)
  - ChatBee messages (left, honey-gold bubbles, bee avatar)
  - Typing indicator (animated dots)
- Input field at bottom: "What are you thinking of buying?"
- Send button (bee icon)
- Quick action chips above keyboard:
  - "Check if I can afford it"
  - "I'm feeling impulsive"
  - "Everyone has this..."

Conversation Flow Example:

1. User: "I want to buy new sneakers for $150"
2. ChatBee analyzes and responds based on:
   - Budget remaining
   - Recent spending patterns
   - Emotional tone detection
   - Social trigger keywords ("everyone," "trending," etc.)

AI Detection Modes (triggered by user input):

Cognitive Load Alert:

- Triggered by: time of day (late night), rapid messages, typos, "tired," "exhausted"
- ChatBee response: "I notice it's late and you might be tired. Decisions made when we're exhausted often lead to regret. Sleep on it?"

Social Trigger Detection:

- Triggered by: keywords like "Instagram," "TikTok," "influencer," "everyone," "trending," "saw online"
- ChatBee response: "Sounds like you saw this on social media! Remember, influencers are paid to sell. Is this truly something YOU need, or just FOMO?"

Emotional Spending Detection:

- Triggered by: keywords like "stressed," "sad," "treat myself," "deserve," "excited," "celebrating"
- ChatBee response: "I hear that you're [emotion]. Shopping can feel good temporarily, but will this purchase truly solve what you're feeling? Let's think of free alternatives together."

Social Comparison Interception:

Social Comparison Interception:

- Triggered by: "everyone has," "my friends have," "I'm the only one without," "keeping up"
- ChatBee response: "Comparison is tough! But your financial health matters more than appearances. What would future-you thank you for choosing today?"

Screen 4: ChatBee Interface (Voice Mode)

Purpose: Same as text mode, but voice-activated

UI Elements:

- Large circular waveform animation (honey-gold, pulsing while user speaks)
- "Listening..." or "ChatBee is thinking..." status text
- Tap-to-speak button (large bee icon, center)
- Voice transcription appears below (user's words in real-time)
- ChatBee's voice response plays with animated bee avatar
- Toggle back to text mode (top right)

Voice Interaction:

- User taps bee icon and speaks
- Agora-like mock: audio captured, converted to text
- AI processes text (same detection logic as text mode)
- Response synthesized to speech and played back
- Transcript saved in chat history

Screen 5: Purchase Decision Outcome

Purpose: Final moment of decision after ChatBee conversation

UI Elements:

- Summary card (glassmorphic) showing:
  - Item considered
  - Price
  - ChatBee's recommendation
- Two large buttons:
  - "I'm buying it" (neutral gray)

    - "I'm skipping it" (honey-gold, bee icon)

If User Selects "I'm buying it":

- Modal appears: "Okay! I've logged this purchase. You have $[amount] left this month."
- Return to Home Dashboard
- Budget updated

If User Selects "I'm skipping it":

- Celebration animation (bee does happy dance, confetti)
- Modal: "Great choice! 🎉 $[amount] added to your Honey Pot!"
- Honey Pot visual fills up slightly
- Return to Home Dashboard

Screen 6: Honey Pot Details

Purpose: View all savings from skipped purchases

UI Elements:

- Large honey jar visual (fills based on total saved)
- Total amount saved (big number, honey-gold)
- List of skipped purchases (card format):
    - Item name
    - Amount saved
    - Date skipped
    - Emoji representing reason (🧠 cognitive, 📱 social media, 💔 emotional, 👥 comparison)
- "Withdraw to Savings Account" button (bottom)

Interactions:

- Scroll through skipped purchase history
- Tap card to see full ChatBee conversation from that decision
- Jar animates when new savings added

Screen 7: Settings

Purpose: Manage budget, preferences, notifications

UI Elements:

- Profile section: Name, avatar
- Budget Management:
  - "Edit Budget" → returns to conversational setup with ChatBee
  - "Reset Budget"
- Preferences:
  - ChatBee voice (male/female/non-binary options)
  - Notification settings
- About:
  - How BudgetBee works
  - Privacy policy
  - Logout

---

3. Exact Component List

UI Components Needed:

Navigation:

- Bottom nav bar (3 tabs)
- Back button
- Header bar

Cards:

- Glassmorphic card (reusable)
- Budget summary card
- Honey Pot card
- Skipped purchase card

Buttons:

- Primary CTA button (honey-gold, rounded)
- Secondary button (gray, rounded)
- Icon button (bee icon)
- Quick action chip

Chat Interface:

- Chat bubble (user)
- Chat bubble (ChatBee)
- Typing indicator
- Input field with send button
- Voice waveform animation

Visuals:

- Bee avatar (multiple expressions: neutral, happy, concerned, celebrating)
- Honey jar (SVG, fillable)
- Progress bar
- Animated confetti
- Bee flight animation

Modals:

- Decision outcome modal
- Confirmation modal
- Celebration modal

Forms:

- Text input
- Number input
- Toggle switch

---

4. Interactions, Animations, and Micro-Interactions

Screen Transitions:

- Fade + slide up (opening ChatBee)
- Slide left/right (navigating tabs)
- Modal slide up from bottom

Button States:

- Default: honey-gold with subtle shadow

- Hover/Press: darker honey-gold, slight scale (0.95)
- Disabled: gray, 50% opacity

ChatBee Animations:

- Bee avatar blinks every 3-5 seconds
- Typing indicator: three dots bounce in sequence
- Message appears with fade + slide from left (ChatBee) or right (user)

Honey Pot Animations:

- New savings: honey "pours" into jar from top (2s animation)
- Jar fills gradually (smooth transition)
- When tapped: slight bounce

Celebration Animations (skipped purchase):

- Bee does 360° spin
- Confetti burst from center (1.5s)
- Honey Pot card pulses 3 times

Voice Mode:

- Waveform pulses in rhythm with user's voice
- Bee avatar "listens" (ear perks up)
- When ChatBee responds: mouth animates with speech

Cognitive Load Alert:

- Screen dims slightly (overlay at 10% opacity)
- Bee avatar shows concerned expression
- Message appears with gentle fade

---

5. AI Behavior Flows (Input → Detection → Output)

Flow 1: Cognitive Load Detection

Input: User opens ChatBee at 11 PM, types "wanna buy this jacket saw it online im tired" **Detection:**

- Timestamp: 11 PM (late night)
- Text analysis: "tired" keyword
- Typing pattern: lowercase, missing punctuation → indicates fatigue **Output:** "I notice it's late and you

might be tired. Decisions made when we're exhausted often lead to regret. Sleep on it?"

Flow 2: Social Trigger Detection

Input: "I saw this on TikTok and everyone's getting it" **Detection:**

- Keywords: "TikTok," "everyone's getting it"
- Social media platform mentioned
- FOMO language pattern **Output:** "Sounds like you saw this on social media! Remember, influencers are paid to sell. Is this truly something YOU need, or just FOMO?"

Flow 3: Emotional Spending Detection

Input: "I had a terrible day and I deserve to treat myself" **Detection:**

- Emotion keywords: "terrible day," "deserve," "treat myself"
- Emotional spending pattern recognized **Output:** "I hear that you're having a rough day. Shopping can feel good temporarily, but will this purchase truly solve what you're feeling? Let's think of free alternatives together."

Flow 4: Social Comparison Interception

Input: "All my friends have AirPods and I'm the only one without them" **Detection:**

- Comparison keywords: "all my friends," "I'm the only one"
- Peer pressure language **Output:** "Comparison is tough! But your financial health matters more than appearances. What would future-you thank you for choosing today?"

Flow 5: Multi-Trigger Detection

Input: "Saw this influencer with the cutest bag and I'm so stressed I need it everyone has designer stuff" **Detection:**

- Social trigger: "influencer"
- Emotional: "stressed," "need"
- Social comparison: "everyone has" **Output:** "Whoa, I'm picking up on a few things here:

1. You saw this from an influencer (that's paid marketing!)
2. You're feeling stressed (shopping won't fix that long-term)
3. You're comparing yourself to others (your journey is unique!)

Let's pause. What's really going on today? And is there a free way to feel better?"

Flow 6: Budget Check (No Triggers)

Input: "I need new running shoes, mine are worn out. They cost $80." **Detection:**

- Necessary purchase (need vs. want)
- Reasonable price
- No emotional/social triggers
- Budget check: User has $200 remaining **Output:** "This sounds like a legitimate need! You have $200 left this month, so $80 for running shoes is totally affordable. Go for it! Want me to log this purchase?"

---

6. User Journey Storyboard

Scenario: Emma (21, college student) wants to buy a $200 dress

Step 1: The Trigger

- Emma sees Instagram post of influencer in a dress
- Thinks: "I need this for the party next week"
- Opens BudgetBee

Step 2: Opening ChatBee

- Taps "Talk to ChatBee" on home dashboard
- Sees current budget: $150 remaining for the month

Step 3: Initial Conversation

- Emma types: "I want to buy this dress I saw on Insta it's $200 and so cute everyone will have something nice"
- ChatBee detects: social trigger (Instagram) + social comparison (everyone)

Step 4: AI Intervention

- ChatBee responds: "I see you found this on Instagram! That's influencer marketing. Also noticed you mentioned 'everyone' — let's focus on YOUR needs, not comparisons. You have $150 left this month. This dress would put you $50 over budget. How important is this really?"

Step 5: Reflection

- Emma thinks for a moment
- Types: "I guess I just feel left out"

- Types: "I guess I just feel left out"
- ChatBee: "That's honest! Feeling left out is hard. But spending money you don't have won't solve that feeling. What if you wore something you already own with confidence? Or borrowed from a friend?"

Step 6: Decision

- Emma: "You're right. I'll skip it."
- Selects "I'm skipping it" button

Step 7: Celebration

- Bee dances, confetti bursts
- "$200 added to your Honey Pot!"
- Emma smiles, feels proud

Step 8: Outcome

- Returns to dashboard
- Honey Pot now shows $200 saved
- Emma feels empowered, not deprived

---

7. Technical Architecture

Frontend

Technology: React Native (iOS + Android)

Key Libraries:

- React Navigation (screen routing)
- React Native Voice (Agora-like mock for voice input)
- Lottie (animations: bee, confetti, honey pour)
- React Native SVG (honey jar, bee avatar)
- Expo (rapid prototyping)

State Management:

- React Context API (user budget, chat history, Honey Pot data)

Styling:

- Styled Components
- Glassmorphism: backdrop-blur + semi-transparent backgrounds

- Honey-gold palette: 🟧 `#FDB515` (primary), 🟨 `#FFD700` (accents), ⬜ `#FFF8DC` (backgrounds)

Backend

Technology: Firebase

Firebase Services Used:

1. **Firestore Database:**
    - Collections:
        - `users` (user profile, budget, savings goal)
        - `budgets` (monthly budget, categories, remaining amount)
        - `chat_history` (user-ChatBee conversations)
        - `skipped_purchases` (Honey Pot entries)
2. **Firebase Authentication:**
    - Email/password signup
    - Anonymous auth for demo mode
3. **Firebase Cloud Functions:**
    - `analyzePurchase(userInput)` → returns AI detection results
    - `updateBudget(purchaseAmount)` → deducts from budget
    - `addToHoneyPot(savedAmount)` → adds skipped purchase
4. **Firebase Storage:**
    - User profile pictures
    - Voice recordings (if stored)

AI Logic (Mock Implementation)

Technology: Firebase Cloud Functions + OpenAI API (mock with hardcoded rules for prototype)

Function: **`analyzePurchase(userInput, userContext)`**

Inputs:

- `userInput` (string): User's message
- `userContext` (object):
    - Current budget remaining

- Time of day
- Recent chat history
- User's spending patterns

Processing:

1. **Keyword Detection:**
   - Cognitive: "tired," "exhausted," timestamp > 10 PM
   - Social: "Instagram," "TikTok," "influencer," "saw online," "trending"
   - Emotional: "stressed," "sad," "happy," "deserve," "treat myself," "excited"
   - Comparison: "everyone," "friends have," "only one without," "keeping up"
2. **Budget Check:**
   - Compare purchase price to remaining budget
   - Flag if over budget
3. **Response Generation:**
   - Select appropriate response template based on detected triggers
   - Personalize with user's name, specific item, price
   - Include budget warning if needed

Output:

- `triggers` (array): ["social", "comparison"]
- `response` (string): ChatBee's message
- `recommendation` (string): "skip" or "proceed"

For Prototype: Use hardcoded if-else logic. For production, integrate OpenAI API with custom prompt engineering.

Voice Integration (Agora-like Mock)

Technology: React Native Voice library (speech-to-text) + Text-to-Speech API

Flow:

1. User taps mic button
2. React Native Voice captures audio
3. Audio converted to text (device's native speech recognition)

4. Text sent to `analyzePurchase()` function

5. AI response returned as text

6. Text-to-Speech API converts response to audio

7. Audio plays through device speaker

8. Transcript saved in chat history

For Prototype: Mock voice with placeholder audio files + text display. Real implementation in production.

---

8. Voice/Chat Integration Plan (Agora-like Mock)

Text Chat (Primary Mode)

- Real-time messaging interface
- Messages stored in Firestore `chat_history` collection
- AI responses generated via Cloud Functions
- No external chat SDK needed (custom built)

Voice Chat (Secondary Mode)

Agora-like Mock Implementation:

Phase 1: Prototype (Mock)

- User taps mic → shows animated waveform
- Audio NOT actually recorded
- User types their message instead (behind the scenes)
- Text displayed as "transcription"
- AI responds as usual
- Text-to-speech reads response (native device TTS)
- Simulates Agora voice experience without actual integration

Phase 2: Production (Real Voice)

- Integrate Agora SDK or Deepgram
- Real audio capture → speech-to-text
- AI processing (same as text mode)
- Response → text-to-speech (ElevenLabs or Google Cloud TTS)
- Full voice conversation capability

For Hackathon/Prototype: Use Phase 1 mock. Demonstrate voice UX without complex integration.

---

9. Aesthetics/Style Guide

Color Palette (Honey-Gold Theme)

- **Primary:** `🟧 #FDB515` (Golden Honey) → buttons, accents, bee elements
- **Secondary:** `🟨 #FFD700` (Bright Gold) → highlights, animations
- **Background:** `⬜ #FFF8DC` (Cream) → screen backgrounds
- **Neutral Dark:** `⬛ #3B3B3B` (Charcoal) → text, icons
- **Neutral Light:** `⬜ #F5F5F5` (Off-White) → cards, secondary backgrounds
- **Success:** `🟩 #7CB342` (Honey Green) → positive actions
- **Alert:** `🟥 #FF6F61` (Soft Coral) → warnings, overspending

Typography

- **Headings:** Poppins Bold (friendly, modern)
- **Body:** Inter Regular (clean, readable)
- **ChatBee Messages:** Quicksand Medium (warm, conversational)

Glassmorphism Style

- Semi-transparent backgrounds: `rgba(255, 255, 255, 0.15)`
- Backdrop blur: `blur(10px)`
- Subtle border: `1px solid rgba(255, 255, 255, 0.3)`
- Soft shadow: `0 8px 32px rgba(0, 0, 0, 0.1)`

Bee Accents

- Bee avatar: cute, friendly, non-gendered
- Expressions: neutral, happy, concerned, celebrating, listening
- Honeycomb pattern: subtle background texture on cards
- Hexagonal elements: buttons, icons where appropriate

Icon Style

- Rounded, friendly
- Honey-gold fills with charcoal outlines
- Bee icon variations for different states

Spacing & Layout

- Generous padding: 20px on cards, 16px inside content
- Rounded corners: 16px (cards), 24px (buttons), 12px (inputs)
- Card elevation: subtle, not heavy

---

10. Prototype-Ready Layout Instructions

For Uizard/Figma/Framer:

Artboard Sizes:

- Mobile: 375x812 (iPhone X)
- Create 7 artboards (one per screen)

Screen 1: Onboarding

- Center-align ChatBee avatar (top)
- Stack conversational prompts vertically
- Input fields appear below prompts
- Progress dots at bottom
- Use auto-layout for dynamic content

Screen 2: Home Dashboard

- Header: fixed, 60px height
- Budget card: 200px height, full width - 32px padding
- Honey Pot card: 180px height, same width
- ChatBee button: 64px height, centered, floating above bottom nav
- Bottom nav: fixed, 80px height, 3 equal tabs

Screen 3: ChatBee Text

- Header: 60px fixed
- Chat area: scrollable, flex-grow
- Messages: max-width 80%, align left (ChatBee) / right (user)
- Input bar: fixed bottom, 60px height
- Quick action chips: horizontal scroll above input

Screen 4: ChatBee Voice

- Centered layout
- Waveform: 200x200px, center screen
- Tap-to-speak button: 120x120px, below waveform
- Status text: center, above waveform

Screen 5: Purchase Decision

- Modal: 80% screen height, slide up from bottom
- Summary card: top, 40% of modal height
- Buttons: bottom, 56px height each, 16px gap

Screen 6: Honey Pot Details

- Honey jar: top, 250px height, center
- Total saved: overlaid on jar
- List: scrollable cards below, 80px height each

Screen 7: Settings

- Standard settings list layout
- Section headers: 40px height
- List items: 56px height, tap targets

Component Library Setup (Figma):

1. Create components for:
    - Glassmorphic card (variants: small, medium, large)
    - Button (variants: primary, secondary, disabled)
    - Chat bubble (variants: user, ChatBee)
    - Bee avatar (variants: neutral, happy, concerned, celebrating)
    - Honey jar (layers for fill animation)
2. Color styles:
    - Add all palette colors as styles
    - Name them: Primary/Gold, Background/Cream, etc.
3. Text styles:
    - Heading 1: Poppins Bold 28px
    - Heading 2: Poppins Bold 20px

- Heading 2: Poppins Bold 20px
- Body: Inter Regular 16px
- ChatBee: Quicksand Medium 16px
- Caption: Inter Regular 14px

Interaction Prototyping (Framer/ProtoPie):

1. Screen transitions: 300ms ease-in-out
2. Button press: scale 0.95, 100ms
3. Honey pour animation: 2000ms linear
4. Confetti: particle system, 1500ms
5. Chat message appear: fade + slide, 200ms
6. Voice waveform: loop, amplitude varies

---

11. Pitch-Ready Explanation

The Problem

Young people are overspending, and traditional budgeting apps fail because they don't address the *why* behind bad purchases. The real culprits are:

- Mental fatigue leading to poor decisions
- FOMO from social media and influencers
- Emotional triggers (stress, sadness, excitement)
- Social pressure to keep up with peers

The Solution: BudgetBee

BudgetBee is the first AI budgeting assistant that understands *human psychology*. Instead of just tracking spending after the damage is done, BudgetBee **intervenes in real-time before a bad purchase happens**.

How It Works

1. User is about to buy something
2. They open ChatBee and describe what they want
3. Our AI analyzes their message for psychological triggers
4. ChatBee asks the right questions to make them pause and reflect
5. User decides to buy or skip
6. If they skip, money goes into their "Honey Pot" savings jar

The Magic: AI Detection

BudgetBee doesn't just look at budgets — it reads between the lines:

- **Cognitive Load Detection:** "It's 2 AM and you're tired — sleep on it?"
- **Social Trigger Detection:** "You saw this on TikTok? That's paid marketing, not reality."
- **Emotional Spending Detection:** "You're stressed. Shopping won't fix that. Let's find a free alternative."
- **Social Comparison Detection:** "Your friends have it? Your financial health matters more than appearances."

Why It Works

- **Conversational:** Feels like talking to a supportive friend, not a stern financial advisor
- **Proactive:** Catches bad decisions before they happen
- **Gamified:** Honey Pot savings jar turns skipped purchases into visible wins
- **Voice-enabled:** Quick and easy when you're in the moment

Target Market

18-25 year olds (students, early professionals) who are:

- Struggling with impulse purchases
- Influenced by social media
- Stressed about money but don't use traditional budgeting apps

Business Model (if asked)

- Freemium: Basic ChatBee is free
- Premium: Advanced AI insights, custom savings goals, financial coaching ($4.99/month)
- Partnerships: Credit card companies, financial literacy programs

The Impact

BudgetBee doesn't just save money — it builds healthier financial habits by addressing the psychological roots of overspending. Users learn to recognize their own triggers and make better decisions independently over time.

Summary: What You Have Now

✅ Complete product specification
✅ Every screen defined (7 screens total)
✅ Full component list
✅ Interaction & animation details

✅ AI behavior flows (6 scenarios)
✅ User journey storyboard

✅ Technical architecture (React Native + Firebase)
✅ Voice integration plan
✅ Style guide (honey-gold glassmorphic)
✅ Prototype layout instructions
✅ Pitch-ready explanation

Next Steps:

1. Use this documentation to build UI mockups in Figma/Uizard

2. Implement prototype with React Native + Firebase

3. Mock AI logic with hardcoded rules for demo

4. Create pitch deck using the explanation in section 11

BudgetBee: Smart savings, one conversation at a time. 🐝💰