Module 1 :Homepage Module

---

1. Module Name

Homepage Module

---

2. Purpose of the Module

The Homepage Module is the **entry point of the system**.

Its primary purpose is to:

- Provide a **calm first impression**
- Explain *what the app does* in very simple terms
- Let users **enter the app without pressure**
- Route users to the correct next module (Relax, Focus, Resources, Login)

This module must **reduce anxiety**, not create decisions overload.

---

3. Problems This Module Solves

User Problems

- Users arrive feeling stressed, unfocused, or unsure
- Fear of signups, tracking, or commitment
- Uncertainty about *where to start*

System Problems

- First-time user drop-off
- Users leaving before understanding value
- Overloading users with features too early

---

4. Responsibilities of Homepage Module

The Homepage Module is responsible for:

1. Displaying the **core value proposition**
2. Explaining **Relax vs Focus** clearly
3. Offering **low-commitment entry paths**
4. Providing access to **Login / Signup (optional)**
5. Acting as a **routing hub** to other modules

It does **not** handle:

- Mental wellness logic
- User data storage
- Authentication logic

---

5. Subcomponents of Homepage Module

5.1 Header / Navigation Component

Responsibilities

- Display app identity (logo/name)
- Provide navigation to:
  - Focus Mode
  - Relax Mode
  - Resources
  - Login / Signup

Design Constraints

- Login/Signup visually secondary
- No popups or forced prompts

System View

```
Header
 ├──Logo
 ├──FocusModeLink
 ├──RelaxModeLink
 ├──ResourcesLink
 └──Login/SignupLink
```

5.2 Hero Section Component

Responsibilities

- Communicate the app’s purpose in one glance
- Create emotional safety

Displayed Content

- Headline: *“Your Sanctuary for Focus and Calm”*
- Short description (1–2 lines)
- Primary CTA button

Primary CTA

- “Begin Your Journey”

System Constraint

- CTA should **not** require login

5.3 Secondary CTA Component (Low Commitment Entry)

Responsibilities

- Reduce fear of commitment
- Allow immediate value experience

Examples

- “Try a 2-minute calm session”
- “Start without signing up”

System Behavior

- Directly routes to:
  - Relax Mode OR
  - Focus Mode (quick start)

5.4 Mode Explanation Component (Relax vs Focus)

Responsibilities

- Help users choose *what they need right now*
- Prevent confusion

Displayed Items

- Relax Mode — calm your mind in minutes
- Focus Mode — start tasks without pressure

Design Constraint

- Only one line per mode
- Icons allowed, text minimal

5.5 Trust & Privacy Message Component

Responsibilities

- Build trust immediately
- Reduce fear around data usage

Examples

- “Private by design · No tracking · Free to use”
- “Use without creating an account”

System Constraint

- No legal language
- No long explanations

---

6. Data Handling (Homepage Module)

Data Used

- None (static content only)

Data Stored

- None

Data Sent to Backend

- None

Homepage must load **without backend dependency**

---

7. User Flow (Homepage Module)

First-Time User Flow

User landson Homepage
→Reads headline
→ Choosesone:
├──Begin Journey → Focus/ Relax
├── Try Calm Session → Relax Mode
├── Focus Mode → FocusModule
├── Resources → ResourcesModule
└── Login/ Signup → AuthModule

---

8. Failure & Edge Case Handling

Scenario	System Behavior
Backend down	Homepage still loads
User not logged in	Full homepage usable
User skips CTAs	No forced prompts
Slow network	Static page loads fast

---

9. Security & Privacy Considerations

- No user data collected
- No cookies required
- No tracking scripts
- No analytics in Phase 1

Homepage must be **privacy-neutral**.

---

10. Non-Functional Requirements

- **Performance:** Fast load (<2s)
- **Accessibility:** Clear fonts, readable contrast
- **Responsiveness:** Mobile & desktop friendly
- **Calm UX:** No flashing elements, no animations that demand attention

---

11. Why This Module Design Is Correct

This design:

- Aligns with mental wellness goals
- Minimizes cognitive load
- Encourages exploration without pressure
- Serves as a stable routing layer
- Requires no backend dependency

Module 2: Relax Mode Module

---

1. Module Name

Relax Mode Module

---

2. Purpose of the Module

The Relax Mode Module is designed to help users **calm their mind and nervous system quickly** when they feel stressed, overwhelmed, or emotionally overloaded.

Its purpose is **not therapy**, diagnosis, or long-term treatment.

It provides **short, guided calming experiences** that users can access **without effort, judgment, or commitment**.

---

3. Problems This Module Solves

User Problems

- Feeling mentally overwhelmed or anxious
- Difficulty calming down in the moment
- Lack of simple, immediate stress-relief tools
- Aversion to complex mental health apps

System Problems

- Users leaving before experiencing value
- Overloading users with long sessions or tracking
- Collecting sensitive emotional data unnecessarily

---

4. Responsibilities of Relax Mode Module

The Relax Mode Module is responsible for:

1. Providing **immediate calming experiences**
2. Offering **multiple relaxation techniques** (not one-size-fits-all)
3. Keeping sessions **short and optional**
4. Avoiding data storage and analysis
5. Allowing users to exit at any time without consequence

It does **not**:

- Store emotional data
- Track user behavior
- Diagnose or evaluate mental states
- Require login

---

5. Subcomponents of Relax Mode Module

5.1 Relax Mode Landing Component

Responsibilities

- Gently introduce relaxation options
- Avoid decision overload

Displayed Options

- Breathing
- Visualization
- Grounding / finger-tapping

Design Constraints

- Minimal text
- Calm icons
- No timers shown upfront

System View

```
Relax Mode
 ├── BreathingSession
 ├── VisualizationSession
 └── GroundingSession
```

5.2 Breathing Session Component

Responsibilities

- Guide users through simple breathing patterns
- Regulate physiological stress

Session Characteristics

- Duration: 2–5 minutes
- Visual cues (gentle expansion/contraction)
- Optional audio guidance

System Behavior

- Starts immediately
- Can be stopped anytime
- No completion tracking

Data Handling

- No data stored

- No session history

5.3 Visualization Session Component

Responsibilities

- Help users mentally shift to a calm, safe imagery
- Reduce racing thoughts

Session Flow

1. Soft voice guidance
2. Gentle imagery prompts (nature, safety, calm moments)
3. Optional finger tapping for grounding

Design Constraints

- Non-specific imagery (avoid triggering content)
- Inclusive language

Data Handling

- No saved responses
- No emotional logging

5.4 Grounding / Finger Tapping Component

Responsibilities

- Provide a body-based calming technique
- Help users anchor attention in the present moment

Technique Used

- Guided finger tapping
- Simple rhythmic prompts

Why Included

- Effective for users who struggle with breathing exercises
- Helpful during high anxiety moments

5.5 Session End / Gentle Exit Component

Responsibilities

- Close sessions without pressure
- Reinforce safety and self-compassion

Example Messages

- “Take your time.”
- “You can return whenever you need.”
- “It’s okay to stop here.”

System Constraint

- No “session completed” badges
- No streaks or rewards

---

6. Data Handling (Relax Mode)

Data Used

- None (only real-time interaction)

Data Stored

- None

Data Sent to Backend

- None

Relax Mode is **entirely stateless**.

---

7. User Flow (Relax Mode Module)

Typical User Flow

User enters Relax Mode
→ Chooses technique
→Session begins immediately

→User exits anytime
→Returnsto homepageor closes app

No forced progression.

---

8. Failure & Edge Case Handling

Scenario	System Behavior
User exits mid-session	Session stops silently
Audio unavailable	Visual-only mode
Network down	Sessions still work
User overwhelmed	No prompts or alerts

---

9. Security & Privacy Considerations

- No personal data collected
- No emotional data stored
- No backend calls
- No cookies required

Relax Mode is **privacy-safe by design**.

---

10. Non-Functional Requirements

- **Latency:** Immediate start
- **Accessibility:** Calm visuals, readable text
- **Responsiveness:** Mobile-first friendly
- **UX Safety:** No sudden sounds or animations

---

11. Why This Module Design Is Correct

This design:

- Matches real user stress behavior
- Minimizes cognitive effort
- Avoids medical framing
- Builds trust through non-collection of data
- Provides immediate value

---

12. Module Summary (One Paragraph)

The Relax Mode Module provides short, guided relaxation techniques—breathing, visualization, and grounding—to help users calm their mind in moments of stress. It is entirely stateless, requires no login, stores no data, and allows users to enter or exit sessions freely. The design prioritizes immediacy, privacy, and emotional safety, ensuring users can find relief without pressure or commitment.

Module 3: Focus Mode Module

---

1. Module Name

Focus Mode Module

---

2. Purpose of the Module

The Focus Mode Module helps users **start tasks, reduce procrastination, and regain clarity** when they feel mentally stuck or overwhelmed.

This module is **not a productivity tracker** or task manager.

Its purpose is to **lower the psychological barrier to starting work**, using gentle, time-limited, and non-judgmental tools.

---

3. Problems This Module Solves

User Problems

- “I don’t know where to start”
- Feeling overwhelmed by tasks
- Procrastination driven by anxiety or perfectionism
- Guilt associated with unfinished work

System Problems

- Users abandoning productivity tools due to pressure
- Over-collection of personal task data
- Rigid workflows that don’t fit fluctuating energy levels

---

4. Responsibilities of Focus Mode Module

The Focus Mode Module is responsible for:

1. Reducing mental overload before work begins
2. Helping users define **small, concrete actions**
3. Enabling **time-boxed focus sessions**
4. Supporting gentle, flexible planning
5. Avoiding guilt, scoring, or performance pressure

It does **not**:

- Evaluate productivity
- Enforce schedules
- Compare users
- Permanently store task history by default

---

5. Subcomponents of Focus Mode Module

5.1 Focus Mode Landing Component

Responsibilities

- Offer clear entry points into focus tools
- Avoid overwhelming the user with steps

Displayed Options

- Clear my head (Brain Dump)
- Start a focus session
- Plan for today / tomorrow

Design Constraint

- No forced sequence
- User can choose any starting point

5.2 Brain Dump Component

Responsibilities

- Allow users to externalize thoughts
- Reduce cognitive overload

Functionality

- Free-form text input
- No structure or validation
- Optional auto-clear after session

Data Handling

- Stored temporarily (session or short TTL)
- Never backed up to server

Why It Exists

- Unwritten thoughts create anxiety
- Externalizing restores mental clarity

5.3 Next Physical Action Component

Responsibilities

- Convert vague tasks into concrete actions
- Reduce task initiation paralysis

Functionality

- Prompt user to define the *next visible physical action*
- Reject overly vague inputs
- Provide example hints

Example

- “Study for exam” → “Open notebook and write today’s date”

Data Handling

- Stored locally (temporary)
- Eligible for encrypted backup (optional)

5.4 Time-Boxing / Pomodoro Component

Responsibilities

- Enable focused work without pressure
- Contain effort within a safe time boundary

Functionality

- User selects session length (10, 15, 25 minutes)
- Countdown timer
- Gentle session completion message
- Early stop allowed

Design Constraints

- No streaks
- No failure states
- No forced breaks

5.5 Gentle Planning Component

Responsibilities

- Reduce future procrastination
- Minimize decision fatigue

Sub-features

- Plan tomorrow (1–3 tasks)

- Break tasks into small time blocks
- Mark blocks as done / not done

Design Constraints

- No fixed schedules
- No backlog pressure
- Plans auto-expire after 24–48 hours

5.6 Self-Compassion Closure Component

Responsibilities

- Separate self-worth from productivity
- Prevent shame-based disengagement

Example Messages

- “You showed up. That’s enough.”
- “Stopping is allowed.”
- “You can start again anytime.”

System Constraint

- No performance metrics
- No negative feedback

---

6. Data Handling (Focus Mode)

Data Types

- Brain dump text
- Next actions
- Time blocks
- Completion status

Storage Rules

- Stored locally by default
- Auto-deleted after 24–48 hours
- Optional encrypted backup (login required)

Backend Interaction

- Only when backup is enabled
- Encrypted data only

---

7. User Flow (Focus Mode Module)

Typical User Flow

User enters Focus Mode
→ Choosestool(Brain Dump / Focus / Plan)
→ Worksforshort time
→ Stops freely
→ Data stored temporarily or backedup(optional)

No forced progression, no penalties.

---

8. Failure & Edge Case Handling

Scenario	System Behavior
User stops mid-session	Session ends silently
Data expires	Cleared without alerts
Backend unavailable	Local use unaffected
User skips steps	Fully allowed

---

9. Security & Privacy Considerations

- Sensitive content stays on device by default
- Encrypted before backup
- No analytics tied to tasks
- No behavioral scoring

Focus Mode is **privacy-first by design**.

---

10. Non-Functional Requirements

- **Responsiveness:** Immediate interaction
- **Usability:** Minimal input required
- **Accessibility:** Simple language, readable UI
- **Resilience:** Works offline

---

11. Why This Module Design Is Correct

This design:

- Matches real procrastination psychology
- Reduces cognitive and emotional load
- Avoids toxic productivity patterns
- Encourages repeat use without pressure
- Aligns with mental wellness goals

---

12. Module Summary (One Paragraph)

The Focus Mode Module provides gentle, psychology-informed tools to help users start tasks without pressure. By combining brain dumping, next-action framing, time-boxed focus sessions, and flexible planning, it reduces overwhelm and procrastination. All data is temporary by default, with optional encrypted backup, ensuring privacy, flexibility, and emotional safety.

Module 4: Resources Module

---

1. Module Name

Resources Module

---

2. Purpose of the Module

The Resources Module provides users with **trusted external support options** when self-guided tools are not sufficient.

Unlike Relax and Focus modes, this module uses the **backend to dynamically serve curated resources**.

Its goals are to:

- Provide **up-to-date, trustworthy resources**
- Allow centralized control over content
- Maintain ethical responsibility
- Keep the frontend lightweight and static

---

3. Problems This Module Solves

User Problems

- Not knowing where to seek reliable help
- Overwhelming or unverified information online
- Needing support beyond self-help tools

System Problems

- Hardcoding sensitive resources in frontend
- Difficulty updating helplines or links
- Risk of outdated or incorrect information

---

4. Why Backend Is Used for This Module

This module **intentionally uses the backend** because:

- Resources may change (helplines, websites, books)
- Content needs centralized updates
- Frontend redeployments should not be required for content changes
- Regional resources may be added later

> **This is a read-only, non-personal backend interaction.**

---

5. Responsibilities of Resources Module

The Resources Module is responsible for:

1. Fetching curated resources from the backend
2. Displaying them clearly and safely on the frontend
3. Providing ethical disclaimers and boundaries
4. Ensuring resources are accessible without login

It does **not**:

- Track user clicks
- Store personal data
- Personalize content based on user behavior

---

6. High-Level Architecture (Resources Module)

Frontend (Resources Page)||GET/api/resources|
Backend (Express API)||
MongoDB/Static Config

---

7. Subcomponents of Resources Module

7.1 Resources Landing Component (Frontend)

Responsibilities

- Display introduction text
- Explain purpose of resources
- Trigger backend fetch on load

System Behavior

On page load, frontend sends a request to backend:
GET /api/resources

-

7.2 Backend Resources Service

Responsibilities

- Store and manage curated resource lists
- Return categorized resource data to frontend

Data Source

- MongoDB collection **or**
- Static JSON configuration (initially)

Why MongoDB (Optional)

- Easy updates
- Category-based queries
- Future localization support

7.3 Immediate Help / Helplines Component (Frontend)

Responsibilities

- Display crisis and support helplines
- Emphasize external nature of help

Backend Role

- Backend sends:
  - Helpline name
  - Description
  - Contact method (call/link)
  - Region (optional)

Frontend Constraint

- No tracking
- Links open externally

7.4 Educational Resources Component (Frontend)

Responsibilities

- Display books, blogs, trusted sites

Backend Role

- Backend sends curated lists with:
  - Title
  - Type (book/article/site)
  - Short description
  - External URL

7.5 Disclaimer & Boundaries Component

Responsibilities

- Clearly state limits of the app

Backend Role

- Backend may provide disclaimer text to ensure consistency

---

8. Backend API Design (Resources Module)

API Endpoint

GET /api/resources

Example Response

{"helplines":[{"name":"National Mental Health Helpline","description":"24/7 support","contact":"<https://example.org>","region":"India"}],"education":[{"title":"Managing Stress","type":"article","url":"<https://example.com>"}],"disclaimer":"This app is not a substitute for professional mental health care."}

---

9. Data Handling & Privacy

- No user identifiers sent
- No request parameters tied to user behavior
- Backend logs kept minimal
- No analytics on resource usage

This ensures **zero privacy risk**.

---

10. User Flow (Resources Module)

User opens Resources page
→ Frontend requests resource list
→ Backendreturns curated data
→ Frontend renders categories
→User opensexternal links

---

11. Failure & Edge Case Handling

Scenario	System Behavior
Backend down	Show cached/static resources
Network slow	Loading indicator
Invalid link	Graceful error message

---

12. Non-Functional Requirements

- **Availability:** Backend highly available
- **Maintainability:** Easy content updates
- **Security:** Read-only endpoint
- **Performance:** Fast response (low payload)

---

13. Why This Design Is Correct

Using the backend here:

- Keeps sensitive resource updates centralized
- Prevents outdated or incorrect information
- Avoids frontend redeployment for content changes
- Maintains ethical responsibility

At the same time:

- No personal data is exposed
- No user behavior is tracked

---

14. Module Summary (One Paragraph)

The Resources Module uses the backend to dynamically serve curated, trustworthy external support resources to the frontend. By centralizing content management on the server while keeping the frontend read-only and anonymous, the system ensures resources remain accurate, maintainable, and ethically responsible without compromising user privacy or introducing behavioral tracking.

Module 5 : Login / Sign-Up Module

---

1. Module Name

Login / Sign-Up Module

---

2. Purpose of the Module

The Login / Sign-Up Module provides **optional user authentication** to enable **encrypted cloud backup** and **cross-device continuity**.

This module is intentionally **non-mandatory**.

Users can fully use the app **without creating an account**.

The purpose is **data safety and continuity**, not user tracking or feature gating.

---

3. Problems This Module Solves

User Problems

- Loss of focus data due to device change or browser reset

- Desire to continue work across devices
- Need for a secure backup option

System Problems

- Need to persist data safely for willing users
- Ensuring privacy while supporting continuity
- Avoiding forced identity creation

---

4. Design Philosophy of This Module

> **Authentication exists to protect user data, not to control access.**

Key principles:

- Optional by design
- Minimal data collection
- Clear consent
- Easy exit (delete account & data)

---

5. Responsibilities of Login / Sign-Up Module

This module is responsible for:

1. Registering users securely
2. Authenticating returning users
3. Managing authentication tokens
4. Enabling encrypted backup functionality
5. Allowing users to delete their account and data

It does **not**:

- Track usage behavior
- Store mental health content in plaintext
- Gate core app features
- Personalize content

---

6. High-Level Architecture (Login Module)

```
Frontend(LoginUI)
      |
      | HTTPS (Credentials / Tokens)
      |Backend(AuthAPI-Express)
      |
      |MongoDB(UsersCollection)
```

7. Subcomponents of Login / Sign-Up Module

7.1 Login / Sign-Up UI Component (Frontend)

Responsibilities

- Collect email and password
- Explain why login is optional
- Obtain explicit consent for backups

Design Constraints

- Calm, non-pressuring language
- No dark patterns
- Clear “Continue without login” option

Example Copy

> “Create an account only if you want to save your focus data across devices.”

7.2 Authentication API (Backend)

Responsibilities

- Handle registration
- Handle login
- Issue JWT tokens
- Validate credentials

Endpoints

```
POST /auth/register
POST /auth/login
```

7.3 User Model (Database)

Stored Fields

{"_id":"ObjectId","email":"string","passwordHash":"string","createdAt":"timestamp"}

Design Constraints

- No profile data
- No mental health metadata
- Email stored securely (hashed or encrypted)

7.4 Token Management Component

Responsibilities

- Generate JWT on login
- Store token securely on client
- Attach token to backup requests

Design Constraints

- Short-lived tokens
- HTTPS only
- No token-based tracking

7.5 Account Deletion Component

Responsibilities

- Allow user to delete account
- Remove all backups immediately
- Revoke authentication tokens

System Guarantee

> “Deleting your account deletes all stored data.”

8. Data Handling & Security

Credentials

- Passwords hashed using bcrypt
- Never stored in plaintext

Focus Data

- Encrypted on client
- Stored encrypted on server
- Never decrypted on backend

Retention

- User data stored only while account exists
- Immediate deletion on request

9. User Flow (Login / Sign-Up Module)

Sign-Up Flow

User chooseslogin
→ Sees consent explanation
→ Submits credentials
→ Account created
→ Backup enabled

Login Flow

User logsin
→ JWT issued
→Encrypted backup restored (optional)

Account Deletion Flow

User requests deletion
→ Backups deleted
→ Account removed

→Userloggedout

10. Failure & Edge Case Handling

Scenario	System Behavior
Login fails	Clear error, retry allowed
Backend unavailable	App usable without login
Token expired	Re-login required
Backup restore fails	Local data unaffected

11. Non-Functional Requirements

- **Security:** Strong password hashing
- **Availability:** Auth service isolated
- **Privacy:** Minimal user data
- **Usability:** Login never blocks core features

12. Why This Module Design Is Correct

This design:

- Preserves privacy-first philosophy
- Avoids forced accounts
- Provides real user value (backup)
- Keeps backend scope minimal
- Is legally and ethically defensible

13. Module Summary (One Paragraph)

The Login / Sign-Up Module provides optional authentication solely to support encrypted data backups and cross-device continuity. It collects minimal user information, secures credentials using standard cryptographic practices, and ensures users retain full control over their data, including the ability to delete their account and backups at any time. Core app functionality remains accessible without authentication.