

Midnight Magnolia Content Creation Dashboard Setup

Based on your current status on Replit and the need for a dashboard similar to Honeybook, Jasper, and a basic CRM, I'll provide a comprehensive setup guide for the Replit environment. This will focus on creating a Jasper-like AI content generation dashboard for Midnight Magnolia that aligns with your ADHD-friendly requirements and Southern Gothic aesthetic.

1. Dashboard Structure and Setup

Initial Setup (index.js)

javascript

Copy

```
const express = require('express');
const path = require('path');
const bodyParser = require('body-parser');
const session = require('express-session');
const mongoose = require('mongoose');
require('dotenv').config();

const app = express();
const PORT = process.env.PORT || 3000;

// Middleware
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));
app.use(express.static(path.join(__dirname, 'public')));
app.set('view engine', 'ejs');

// Session setup
app.use(session({
  secret: process.env.SESSION_SECRET || 'midnight-magnolia-secret',
  resave: false,
  saveUninitialized: true,
  cookie: { secure: false, maxAge: 24 * 60 * 60 * 1000 } // 24 hours
}));

// MongoDB connection
mongoose.connect(process.env.MONGODB_URI || 'mongodb://
```

```
localhost:27017/midnight-magnolia', {
  useUrlParser: true,
  useUnifiedTopology: true
}).then(() => {
  console.log('Connected to MongoDB');
}).catch(err => {
  console.error('MongoDB connection error:', err);
});
```

// Routes

```
const dashboardRoutes = require('./routes/dashboard');
const contentRoutes = require('./routes/content');
const inventoryRoutes = require('./routes/inventory');
const salesRoutes = require('./routes/sales');
const clientRoutes = require('./routes/clients');
```

```
app.use('/', dashboardRoutes);
app.use('/content', contentRoutes);
app.use('/inventory', inventoryRoutes);
app.use('/sales', salesRoutes);
app.use('/clients', clientRoutes);
```

// Start server

```
app.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`);
});
```

Main Dashboard Structure (views/dashboard.ejs)

html

Copy

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Midnight Magnolia Dashboard</title>
  <link href="https://fonts.googleapis.com/css2?
family=Playfair+Display:wght@400;700&family=Lora:wght@400;500&fa
```

```

mily=Montserrat:wght@400;600&display=swap" rel="stylesheet">
  <link rel="stylesheet" href="/css/styles.css">
</head>
<body class="bg-midnight-blue text-magnolia-white">
  <div class="dashboard-container">
    <!-- Sidebar -->
    <div class="sidebar">
      <div class="logo">
        
        <h2 class="font-heading text-rich-gold">Midnight Magnolia</h2>
      </div>

      <nav class="sidebar-nav">
        <ul>
          <li class="active"><a href="/"><span class="icon"> 🏠 </
span>Dashboard</a></li>
          <li><a href="/content"><span class="icon"> ✨ </span>Content
Creation</a></li>
          <li><a href="/inventory"><span class="icon"> 📚 </span>Digital
Inventory</a></li>
          <li><a href="/sales"><span class="icon"> 💰 </span>Sales &
Orders</a></li>
          <li><a href="/clients"><span class="icon"> 👤 </span>Client
Management</a></li>
          <li><a href="/settings"><span class="icon"> ⚙️ </span>Settings</
a></li>
        </ul>
      </nav>

      <div class="energy-tracker">
        <h3 class="font-accent">Energy Tracker</h3>
        <div class="energy-levels">
          <button class="energy-level high">High</button>
          <button class="energy-level medium">Medium</button>
          <button class="energy-level low">Low</button>
        </div>
      </div>
    </div>
  </div>

```

```

<!-- Main Content -->
<div class="main-content">
  <header>
    <div class="header-left">
      <h1 class="font-heading">Welcome, Latisha</h1>
      <p class="lunar-phase">🌑 Waxing Crescent | Monday, October
30, 2023</p>
    </div>
    <div class="header-right">
      <div class="notifications">
        <span class="icon">🔔</span>
        <span class="counter">3</span>
      </div>
      <div class="user-profile">
        
      </div>
    </div>
  </header>

  <!-- Dashboard Content -->
  <div class="dashboard-content">
    <!-- Stats Overview -->
    <div class="stats-container">
      <div class="stat-card">
        <h3>Revenue</h3>
        <p class="stat-value">$2,450</p>
        <p class="stat-change positive">+18% this month</p>
      </div>
      <div class="stat-card">
        <h3>Orders</h3>
        <p class="stat-value">32</p>
        <p class="stat-change positive">+5 since last week</p>
      </div>
      <div class="stat-card">
        <h3>Products</h3>
        <p class="stat-value">14</p>
        <p class="stat-change neutral">No change</p>
      </div>
    </div>
  </div>

```

```
<div class="stat-card">
  <h3>Goal Progress</h3>
  <p class="stat-value">$2,450/$10,000</p>
  <div class="progress-bar">
    <div class="progress" style="width: 24.5%"></div>
  </div>
</div>
</div>
```

```
<!-- Recent Activity & Tasks -->
<div class="activity-tasks-container">
  <div class="recent-activity">
    <h2 class="section-title font-heading">Recent Activity</h2>
    <ul class="activity-list">
      <li class="activity-item">
        <span class="activity-icon sale">💰</span>
        <div class="activity-details">
          <p class="activity-title">New sale: The Magnolia Reset
Journal</p>
          <p class="activity-time">2 hours ago</p>
        </div>
      </li>
      <li class="activity-item">
        <span class="activity-icon content">📝</span>
        <div class="activity-details">
          <p class="activity-title">Created 5 new affirmation cards</p>
          <p class="activity-time">Yesterday</p>
        </div>
      </li>
      <li class="activity-item">
        <span class="activity-icon client">👤</span>
        <div class="activity-details">
          <p class="activity-title">New client: Maya Johnson</p>
          <p class="activity-time">2 days ago</p>
        </div>
      </li>
    </ul>
    <a href="/activity" class="view-all">View all activity</a>
  </div>
```

```

<div class="tasks">
  <h2 class="section-title font-heading">Today's Tasks</h2>
  <ul class="task-list">
    <li class="task-item">
      <label class="task-checkbox">
        <input type="checkbox">
        <span class="checkmark"></span>
        <span class="task-text">Complete Tarot deck descriptions</
span>
      </label>
      <span class="task-priority high">High</span>
    </li>
    <li class="task-item">
      <label class="task-checkbox">
        <input type="checkbox">
        <span class="checkmark"></span>
        <span class="task-text">Upload new journal templates</
span>
      </label>
      <span class="task-priority medium">Medium</span>
    </li>
    <li class="task-item">
      <label class="task-checkbox">
        <input type="checkbox" checked>
        <span class="checkmark"></span>
        <span class="task-text">Respond to customer inquiries</
span>
      </label>
      <span class="task-priority completed">Completed</span>
    </li>
  </ul>
  <button class="add-task-btn">+ Add New Task</button>
</div>
</div>

```

<!-- Quick Create Buttons -->

```

<div class="quick-create">
  <h2 class="section-title font-heading">Quick Create</h2>

```

```

<div class="quick-buttons">
  <a href="/content/affirmation" class="quick-button">
    <span class="button-icon"> ✨ </span>
    <span class="button-text">Affirmation Card</span>
  </a>
  <a href="/content/tarot" class="quick-button">
    <span class="button-icon"> 🌺 </span>
    <span class="button-text">Tarot Description</span>
  </a>
  <a href="/content/journal" class="quick-button">
    <span class="button-icon"> 📖 </span>
    <span class="button-text">Journal Prompt</span>
  </a>
  <a href="/content/listing" class="quick-button">
    <span class="button-icon"> 🛍️ </span>
    <span class="button-text">Product Listing</span>
  </a>
</div>
</div>
</div>
</div>
</div>

```

```

<script src="/js/dashboard.js"></script>
</body>
</html>

```

2. Content Creation Module (views/content.ejs)

html

Copy


```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Content Creation | Midnight Magnolia</title>
  <link href="https://fonts.googleapis.com/css2?
family=Playfair+Display:wght@400;700&family=Lora:wght@400;500&fa

```

```

mily=Montserrat:wght@400;600&display=swap" rel="stylesheet">
  <link rel="stylesheet" href="/css/styles.css">
</head>
<body class="bg-midnight-blue text-magnolia-white">
  <div class="dashboard-container">
    <!-- Sidebar (same as dashboard) -->

    <!-- Main Content -->
    <div class="main-content">
      <header>
        <div class="header-left">
          <h1 class="font-heading">Content Creation</h1>
          <p class="moon-phase"> Waxing Crescent | Ideal for: New
beginnings, setting intentions</p>
        </div>
        <div class="header-right">
          <!-- Same as dashboard -->
        </div>
      </header>

      <!-- Content Creation Area -->
      <div class="content-area">
        <div class="content-sidebar">
          <h3 class="font-heading">Content Types</h3>
          <ul class="content-types">
            <li class="active"><a href="/content/affirmation">Affirmation
Cards</a></li>
            <li><a href="/content/tarot">Tarot Descriptions</a></li>
            <li><a href="/content/journal">Journal Prompts</a></li>
            <li><a href="/content/listing">Product Listings</a></li>
            <li><a href="/content/social">Social Media</a></li>
            <li><a href="/content/email">Email Templates</a></li>
          </ul>

          <div class="saved-templates">
            <h3 class="font-heading">Saved Templates</h3>
            <ul class="template-list">
              <li><a href="#">Black Feminine Energy Affirmations</a></li>
              <li><a href="#">Recovery Journal Prompts</a></li>

```



```
    <li><a href="#">Sobriety Planner Listing</a></li>
  </ul>
</div>
</div>
```

```
<div class="content-generator">
  <h2 class="font-heading">Create Affirmation Card</h2>
```

```
  <div class="prompt-container">
    <label for="content-prompt" class="font-accent">Enter Your
Prompt:</label>
    <textarea id="content-prompt" class="prompt-textarea"
placeholder="Write a 2-line affirmation rooted in Black Southern
feminine energy. Keep it poetic, healing, and brief enough to fit on a 3x5
card."></textarea>
```

```
  <div class="template-options">
    <h4 class="font-accent">Prompt Template Options:</h4>
    <div class="template-buttons">
      <button class="template-button" data-template="Write a 2-
line affirmation rooted in Black Southern feminine energy about
resilience.">Resilience</button>
      <button class="template-button" data-template="Write a 2-
line affirmation rooted in Black Southern feminine energy about
healing.">Healing</button>
      <button class="template-button" data-template="Write a 2-
line affirmation rooted in Black Southern feminine energy about softness
as strength.">Softness</button>
      <button class="template-button" data-template="Write a 2-
line affirmation rooted in Black Southern feminine energy about
ancestral wisdom.">Ancestral</button>
    </div>
  </div>
```

```
<div class="tone-settings">
  <h4 class="font-accent">Tone & Style:</h4>
  <div class="tone-options">
    <label class="tone-option">
      <input type="checkbox" checked> Poetic
```

```
</label>
<label class="tone-option">
  <input type="checkbox" checked> Healing
</label>
<label class="tone-option">
  <input type="checkbox"> Empowering
</label>
<label class="tone-option">
  <input type="checkbox"> Reflective
</label>
</div>
</div>
```

```
<button id="generate-btn" class="generate-button">Generate
Content</button>
</div>
```

```
<div class="results-container">
  <div class="results-header">
    <h3 class="font-heading">Generated Results</h3>
    <div class="results-actions">
      <button class="results-action-btn refresh"><span
class="icon"><img alt="refresh icon" data-bbox="271 558 298 581"/></span> Regenerate</button>
      <button class="results-action-btn save"><span class="icon">
<img alt="save icon" data-bbox="125 601 152 624"/></span> Save</button>
      <button class="results-action-btn export"><span
class="icon"><img alt="export icon" data-bbox="271 644 298 667"/></span> Export</button>
    </div>
  </div>
```

```
<div class="results-display">
  <div class="result-card">
    <div class="result-content">
      <p>"Your softness ain't weakness, honey;</p>
      <p>It's the river that carved the canyon."</p>
    </div>
    <div class="result-actions">
      <button class="result-action copy" title="Copy to
clipboard"><span class="icon"><img alt="copy icon" data-bbox="463 914 490 937"/></span></button>
```

```
        <button class="result-action edit" title="Edit"><span
class="icon">✎ </span></button>
```

```
        <button class="result-action visualize"
title="Visualize"><span class="icon">👁 </span></button>
```

```
    </div>
```

```
</div>
```

```
<div class="result-card">
```

```
    <div class="result-content">
```

```
        <p>"When ancestors whisper through magnolia blooms,</p>
```

```
        <p>Your healing becomes their living legacy."</p>
```

```
    </div>
```

```
    <div class="result-actions">
```

```
        <button class="result-action copy" title="Copy to
clipboard"><span class="icon">📋 </span></button>
```

```
        <button class="result-action edit" title="Edit"><span
class="icon">✎ </span></button>
```

```
        <button class="result-action visualize"
title="Visualize"><span class="icon">👁 </span></button>
```

```
    </div>
```

```
</div>
```

```
<div class="result-card">
```

```
    <div class="result-content">
```

```
        <p>"Stars that guided runaways now light your path;</p>
```

```
        <p>Their determination flows in your veins."</p>
```

```
    </div>
```

```
    <div class="result-actions">
```

```
        <button class="result-action copy" title="Copy to
clipboard"><span class="icon">📋 </span></button>
```

```
        <button class="result-action edit" title="Edit"><span
class="icon">✎ </span></button>
```

```
        <button class="result-action visualize"
title="Visualize"><span class="icon">👁 </span></button>
```

```
    </div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```

<div class="visualization-preview">
  <h3 class="font-heading">Card Preview</h3>
  <div class="preview-container">
    <div class="card-preview" style="background-color: #0A192F;
color: #FAF3E0;">
      <div class="card-content">
        <p class="card-text">"Your softness ain't weakness,
honey;</p>
        <p class="card-text">"It's the river that carved the canyon."</
p>
      </div>
      <div class="card-footer">
        
        <p class="card-brand">Midnight Magnolia</p>
      </div>
    </div>

    <div class="card-style-options">
      <h4 class="font-accent">Card Styling:</h4>
      <div class="style-options">
        <div class="style-group">
          <label>Background Color:</label>
          <div class="color-options">
            <button class="color-option active" style="background-
color: #0A192F;"></button>
            <button class="color-option" style="background-color:
#0A3B4D;"></button>
            <button class="color-option" style="background-color:
#FAF3E0;"></button>
          </div>
        </div>

        <div class="style-group">
          <label>Text Color:</label>
          <div class="color-options">
            <button class="color-option" style="background-color:
#0A192F;"></button>
            <button class="color-option active" style="background-

```

```
color: #FAF3E0;"></button>
    <button class="color-option" style="background-color:
#D4AF37;"></button>
    </div>
</div>
```

```
<div class="style-group">
    <label>Font:</label>
    <select class="font-select">
        <option value="playfair">Playfair Display</option>
        <option value="lora">Lora</option>
        <option value="montserrat">Montserrat</option>
    </select>
</div>
```

```
<div class="style-group">
    <label>Icon:</label>
    <select class="icon-select">
        <option value="moon">Crescent Moon</option>
        <option value="magnolia">Magnolia Flower</option>
        <option value="none">None</option>
    </select>
</div>
</div>
```

```
<div class="export-options">
    <button class="export-btn">Export as PNG</button>
    <button class="export-btn">Export as JPG</button>
    <button class="export-btn">Export as PDF</button>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
```

```
<script src="/js/content.js"></script>
```

</body>

</html>

3. Styles (public/css/styles.css)

CSS

Copy

```
:root {  
  /* Brand colors */  
  --midnight-blue: #0A192F;  
  --midnight-teal: #0A3B4D;  
  --magnolia-white: #FAF3E0;  
  --rich-gold: #D4AF37;  
  --sage-green: #A3B18A;  
  
  /* Typography */  
  --heading-font: 'Playfair Display', serif;  
  --body-font: 'Lora', serif;  
  --accent-font: 'Montserrat', sans-serif;  
  
  /* Spacing */  
  --spacing-xs: 0.25rem;  
  --spacing-sm: 0.5rem;  
  --spacing-md: 1rem;  
  --spacing-lg: 1.5rem;  
  --spacing-xl: 2rem;  
  --spacing-xxl: 3rem;  
}  
  
/* Base Styles */  
* {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
}  
  
body {  
  font-family: var(--body-font);  
  background-color: var(--midnight-blue);  
  color: var(--magnolia-white);
```

```
    line-height: 1.6;
}

h1, h2, h3, h4, h5, h6 {
    font-family: var(--heading-font);
    color: var(--magnolia-white);
    margin-bottom: var(--spacing-md);
}
```

```
.font-heading {
    font-family: var(--heading-font);
}
```

```
.font-body {
    font-family: var(--body-font);
}
```

```
.font-accent {
    font-family: var(--accent-font);
}
```

```
.text-rich-gold {
    color: var(--rich-gold);
}
```

```
/* Dashboard Layout */
.dashboard-container {
    display: flex;
    min-height: 100vh;
}
```

```
/* Sidebar */
.sidebar {
    width: 280px;
    background-color: var(--midnight-teal);
    padding: var(--spacing-lg);
    display: flex;
    flex-direction: column;
}
```

```
.logo {  
  display: flex;  
  align-items: center;  
  margin-bottom: var(--spacing-xl);  
}
```

```
.logo img {  
  width: 40px;  
  height: 40px;  
  margin-right: var(--spacing-md);  
}
```

```
.logo h2 {  
  font-size: 1.2rem;  
  margin-bottom: 0;  
}
```

```
.sidebar-nav ul {  
  list-style: none;  
}
```

```
.sidebar-nav li {  
  margin-bottom: var(--spacing-md);  
}
```

```
.sidebar-nav a {  
  display: flex;  
  align-items: center;  
  text-decoration: none;  
  color: var(--magnolia-white);  
  padding: var(--spacing-sm) var(--spacing-md);  
  border-radius: 6px;  
  transition: all 0.3s ease;  
}
```

```
.sidebar-nav a:hover {  
  background-color: rgba(250, 243, 224, 0.1);  
}
```



```
.sidebar-nav li.active a {  
  background-color: rgba(250, 243, 224, 0.15);  
  color: var(--rich-gold);  
}
```

```
.sidebar-nav .icon {  
  margin-right: var(--spacing-md);  
}
```

```
.energy-tracker {  
  margin-top: auto;  
  background-color: rgba(10, 25, 47, 0.5);  
  padding: var(--spacing-md);  
  border-radius: 8px;  
}
```

```
.energy-tracker h3 {  
  font-size: 0.9rem;  
  text-transform: uppercase;  
  letter-spacing: 1px;  
  margin-bottom: var(--spacing-sm);  
}
```

```
.energy-levels {  
  display: flex;  
  justify-content: space-between;  
}
```

```
.energy-level {  
  flex: 1;  
  padding: var(--spacing-sm);  
  border: none;  
  background: transparent;  
  color: var(--magnolia-white);  
  font-family: var(--accent-font);  
  border-radius: 4px;  
  cursor: pointer;  
  margin: 0 var(--spacing-xs);  
}
```

```
    transition: all 0.3s ease;
}

.energy-level:hover {
    background-color: rgba(250, 243, 224, 0.1);
}

.energy-level.high {
    border-bottom: 2px solid #4CAF50;
}

.energy-level.medium {
    border-bottom: 2px solid #FFC107;
}

.energy-level.low {
    border-bottom: 2px solid #F44336;
}

.energy-level.selected {
    background-color: rgba(250, 243, 224, 0.15);
}

/* Main Content */
.main-content {
    flex: 1;
    padding: var(--spacing-lg);
    overflow-y: auto;
}

header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin-bottom: var(--spacing-xl);
}

.header-left h1 {
    margin-bottom: var(--spacing-xs);
```

```
}
```

```
.lunar-phase {  
  font-family: var(--accent-font);  
  font-size: 0.9rem;  
  color: var(--sage-green);  
}
```

```
.header-right {  
  display: flex;  
  align-items: center;  
}
```

```
.notifications {  
  position: relative;  
  margin-right: var(--spacing-lg);  
  cursor: pointer;  
}
```

```
.notifications .counter {  
  position: absolute;  
  top: -8px;  
  right: -8px;  
  background-color: var(--rich-gold);  
  color: var(--midnight-blue);  
  font-size: 0.75rem;  
  width: 18px;  
  height: 18px;  
  border-radius: 50%;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  font-family: var(--accent-font);  
}
```

```
.user-profile img {  
  width: 40px;  
  height: 40px;  
  border-radius: 50%;
```

```
    cursor: pointer;
}

/* Dashboard Content Sections */
.dashboard-content {
  display: grid;
  grid-template-columns: 1fr;
  gap: var(--spacing-xl);
}

.section-title {
  font-size: 1.5rem;
  color: var(--rich-gold);
  margin-bottom: var(--spacing-lg);
}

/* Stats Cards */
.stats-container {
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  gap: var(--spacing-lg);
}

.stat-card {
  background-color: var(--midnight-teal);
  padding: var(--spacing-lg);
  border-radius: 8px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}

.stat-card h3 {
  font-family: var(--accent-font);
  font-size: 0.9rem;
  text-transform: uppercase;
  letter-spacing: 1px;
  color: var(--magnolia-white);
  margin-bottom: var(--spacing-sm);
}
```

```
.stat-value {  
  font-family: var(--heading-font);  
  font-size: 1.8rem;  
  color: var(--magnolia-white);  
  margin-bottom: var(--spacing-sm);  
}
```

```
.stat-change {  
  font-family: var(--accent-font);  
  font-size: 0.8rem;  
}
```

```
.stat-change.positive {  
  color: #4CAF50;  
}
```

```
.stat-change.negative {  
  color: #F44336;  
}
```

```
.stat-change.neutral {  
  color: #FFC107;  
}
```

```
.progress-bar {  
  height: 8px;  
  background-color: rgba(250, 243, 224, 0.1);  
  border-radius: 4px;  
  overflow: hidden;  
  margin-top: var(--spacing-sm);  
}
```

```
.progress {  
  height: 100%;  
  background-color: var(--rich-gold);  
}
```

```
/* Activity and Tasks */
```

```
.activity-tasks-container {  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
  gap: var(--spacing-xl);  
}
```

```
.recent-activity, .tasks {  
  background-color: var(--midnight-teal);  
  padding: var(--spacing-lg);  
  border-radius: 8px;  
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);  
}
```

```
.activity-list, .task-list {  
  list-style: none;  
}
```

```
.activity-item, .task-item {  
  display: flex;  
  align-items: center;  
  padding: var(--spacing-md) 0;  
  border-bottom: 1px solid rgba(250, 243, 224, 0.1);  
}
```

```
.activity-item:last-child, .task-item:last-child {  
  border-bottom: none;  
}
```

```
.activity-icon {  
  width: 36px;  
  height: 36px;  
  border-radius: 50%;  
  background-color: rgba(250, 243, 224, 0.1);  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  margin-right: var(--spacing-md);  
}
```

```
.activity-details {  
  flex: 1;  
}
```

```
.activity-title {  
  margin-bottom: 4px;  
}
```

```
.activity-time {  
  font-size: 0.8rem;  
  color: rgba(250, 243, 224, 0.7);  
}
```

```
.view-all {  
  display: block;  
  text-align: center;  
  margin-top: var(--spacing-md);  
  color: var(--rich-gold);  
  font-family: var(--accent-font);  
  text-decoration: none;  
}
```

```
/* Tasks */
```

```
.task-checkbox {  
  display: flex;  
  align-items: center;  
  cursor: pointer;  
  flex: 1;  
}
```

```
.task-checkbox input {  
  position: absolute;  
  opacity: 0;  
  cursor: pointer;  
  height: 0;  
  width: 0;  
}
```

```
.checkmark {  
  position: relative;  
  display: inline-block;  
  width: 20px;  
  height: 20px;  
  background-color: rgba(250, 243, 224, 0.1);  
  border-radius: 4px;  
  margin-right: var(--spacing-sm);  
}
```

```
.task-checkbox input:checked ~ .checkmark {  
  background-color: var(--rich-gold);  
}
```

```
.checkmark:after {  
  content: "";  
  position: absolute;  
  display: none;  
}
```

```
.task-checkbox input:checked ~ .checkmark:after {  
  display: block;  
}
```

```
.task-checkbox .checkmark:after {  
  left: 7px;  
  top: 3px;  
  width: 5px;  
  height: 10px;  
  border: solid white;  
  border-width: 0 2px 2px 0;  
  transform: rotate(45deg);  
}
```

```
.task-text {  
  flex: 1;  
}
```



```
.task-checkbox input:checked ~ .task-text {  
  text-decoration: line-through;  
  color: rgba(250, 243, 224, 0.6);  
}
```

```
.task-priority {  
  font-size: 0.75rem;  
  padding: 2px 8px;  
  border-radius: 12px;  
  font-family: var(--accent-font);  
}
```

```
.task-priority.high {  
  background-color: rgba(244, 67, 54, 0.2);  
  color: #F44336;  
}
```

```
.task-priority.medium {  
  background-color: rgba(255, 193, 7, 0.2);  
  color: #FFC107;  
}
```

```
.task-priority.low {  
  background-color: rgba(76, 175, 80, 0.2);  
  color: #4CAF50;  
}
```

```
.task-priority.completed {  
  background-color: rgba(250, 243, 224, 0.2);  
  color: rgba(250, 243, 224, 0.6);  
}
```

```
.add-task-btn {  
  width: 100%;  
  padding: var(--spacing-sm);  
  margin-top: var(--spacing-md);  
  background-color: transparent;  
  border: 1px dashed rgba(250, 243, 224, 0.3);  
}
```

```
color: var(--rich-gold);
border-radius: 4px;
cursor: pointer;
font-family: var(--accent-font);
transition: all 0.3s ease;
}
```

```
.add-task-btn:hover {
  background-color: rgba(250, 243, 224, 0.05);
}
```

```
/* Quick Create */
.quick-create {
  background-color: var(--midnight-teal);
  padding: var(--spacing-lg);
  border-radius: 8px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}
```

```
.quick-buttons {
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  gap: var(--spacing-md);
}
```

```
.quick-button {
  display: flex;
  flex-direction: column;
  align-items: center;
  padding: var(--spacing-lg);
  background-color: rgba(250, 243, 224, 0.05);
  border-radius: 8px;
  text-decoration: none;
  color: var(--magnolia-white);
  transition: all 0.3s ease;
}
```

```
.quick-button:hover {
  background-color: rgba(250, 243, 224, 0.1);
}
```

```
    transform: translateY(-2px);  
}
```

```
.button-icon {  
    font-size: 2rem;  
    margin-bottom: var(--spacing-sm);  
}
```

```
.button-text {  
    font-family: var(--accent-font);  
    font-size: 0.9rem;  
    text-align: center;  
}
```

```
/* Content Creation Page Styles */
```

```
.content-area {  
    display: flex;  
    gap: var(--spacing-xl);  
    margin-top: var(--spacing-lg);  
}
```

```
.content-sidebar {  
    width: 250px;  
    background-color: var(--midnight-teal);  
    padding: var(--spacing-lg);  
    border-radius: 8px;  
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);  
}
```

```
.content-types {  
    list-style: none;  
    margin-bottom: var(--spacing-xl);  
}
```

```
.content-types li {  
    margin-bottom: var(--spacing-sm);  
}
```

```
.content-types a {
```

```
display: block;
padding: var(--spacing-sm) var(--spacing-md);
text-decoration: none;
color: var(--magnolia-white);
border-radius: 4px;
transition: all 0.3s ease;
}
```

```
.content-types li.active a {
  background-color: rgba(250, 243, 224, 0.15);
  color: var(--rich-gold);
}
```

```
.content-types a:hover {
  background-color: rgba(250, 243, 224, 0.1);
}
```

```
.template-list {
  list-style: none;
}
```

```
.template-list li {
  margin-bottom: var(--spacing-xs);
}
```

```
.template-list a {
  font-size: 0.9rem;
  text-decoration: none;
  color: var(--magnolia-white);
  transition: color 0.3s ease;
}
```

```
.template-list a:hover {
  color: var(--rich-gold);
}
```

```
.content-generator {
  flex: 1;
}
```

```
.prompt-container {  
  background-color: var(--midnight-teal);  
  padding: var(--spacing-lg);  
  border-radius: 8px;  
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);  
  margin-bottom: var(--spacing-xl);  
}
```

```
.prompt-textarea {  
  width: 100%;  
  min-height: 100px;  
  padding: var(--spacing-md);  
  background-color: rgba(10, 25, 47, 0.5);  
  border: 1px solid rgba(250, 243, 224, 0.2);  
  border-radius: 4px;  
  color: var(--magnolia-white);  
  font-family: var(--body-font);  
  margin-bottom: var(--spacing-md);  
  resize: vertical;  
}
```

```
.template-options, .tone-settings {  
  margin-bottom: var(--spacing-lg);  
}
```

```
.template-buttons {  
  display: flex;  
  flex-wrap: wrap;  
  gap: var(--spacing-sm);  
  margin-top: var(--spacing-xs);  
}
```

```
.template-button {  
  padding: var(--spacing-xs) var(--spacing-sm);  
  background-color: rgba(250, 243, 224, 0.1);  
  border: none;  
  border-radius: 4px;  
  color: var(--magnolia-white);  
}
```

```
font-family: var(--accent-font);  
cursor: pointer;  
transition: all 0.3s ease;  
}
```

```
.template-button:hover {  
  background-color: rgba(250, 243, 224, 0.2);  
}
```

```
.tone-options {  
  display: flex;  
  flex-wrap: wrap;  
  gap: var(--spacing-md);  
  margin-top: var(--spacing-xs);  
}
```

```
.tone-option {  
  display: flex;  
  align-items: center;  
  cursor: pointer;  
}
```

```
.tone-option input {  
  margin-right: var(--spacing-xs);  
}
```

```
.generate-button {  
  width: 100%;  
  padding: var(--spacing-md);  
  background-color: var(--rich-gold);  
  color: var(--midnight-blue);  
  border: none;  
  border-radius: 4px;  
  font-family: var(--accent-font);  
  font-weight: 600;  
  text-transform: uppercase;  
  letter-spacing: 1px;  
  cursor: pointer;  
  transition: all 0.3s ease;
```

```
}
```

```
.generate-button:hover {  
  background-color: #C49E32;  
}
```

```
.results-container {  
  background-color: var(--midnight-teal);  
  padding: var(--spacing-lg);  
  border-radius: 8px;  
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);  
  margin-bottom: var(--spacing-xl);  
}
```

```
.results-header {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  margin-bottom: var(--spacing-lg);  
}
```

```
.results-actions {  
  display: flex;  
  gap: var(--spacing-sm);  
}
```

```
.results-action-btn {  
  display: flex;  
  align-items: center;  
  padding: var(--spacing-xs) var(--spacing-sm);  
  background-color: rgba(250, 243, 224, 0.1);  
  border: none;  
  border-radius: 4px;  
  color: var(--magnolia-white);  
  font-family: var(--accent-font);  
  cursor: pointer;  
  transition: all 0.3s ease;  
}
```

```
.results-action-btn:hover {  
  background-color: rgba(250, 243, 224, 0.2);  
}
```

```
.results-action-btn .icon {  
  margin-right: var(--spacing-xs);  
}
```

```
.results-display {  
  display: flex;  
  flex-direction: column;  
  gap: var(--spacing-md);  
}
```

```
.result-card {  
  padding: var(--spacing-md);  
  background-color: rgba(10, 25, 47, 0.5);  
  border-radius: 6px;  
  border-left: 3px solid var(--rich-gold);  
  display: flex;  
  justify-content: space-between;  
}
```

```
.result-content {  
  font-family: var(--heading-font);  
  line-height: 1.6;  
}
```

```
.result-actions {  
  display: flex;  
  flex-direction: column;  
  gap: var(--spacing-xs);  
}
```

```
.result-action {  
  background: transparent;  
  border: none;  
  color: var(--magnolia-white);
```



```
    cursor: pointer;
    opacity: 0.7;
    transition: opacity 0.3s ease;
}
```

```
.result-action:hover {
  opacity: 1;
}
```

```
.visualization-preview {
  background-color: var(--midnight-teal);
  padding: var(--spacing-lg);
  border-radius: 8px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}
```

```
.preview-container {
  display: flex;
  gap: var(--spacing-xl);
}
```

```
.card-preview {
  width: 300px;
  height: 200px;
  border-radius: 8px;
  padding: var(--spacing-lg);
  display: flex;
  flex-direction: column;
  justify-content: space-between;
  box-shadow: 0 4px 15px rgba(0, 0, 0, 0.2);
}
```

```
.card-content {
  flex: 1;
  display: flex;
  flex-direction: column;
  justify-content: center;
  text-align: center;
}
```

```
.card-text {  
  font-family: var(--heading-font);  
  line-height: 1.6;  
}
```

```
.card-footer {  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  font-size: 0.8rem;  
  font-family: var(--accent-font);  
  opacity: 0.8;  
}
```

```
.card-icon {  
  width: 16px;  
  height: 16px;  
  margin-right: var(--spacing-xs);  
}
```

```
.card-style-options {  
  flex: 1;  
}
```

```
.style-options {  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
  gap: var(--spacing-md);  
  margin-bottom: var(--spacing-lg);  
}
```

```
.style-group {  
  margin-bottom: var(--spacing-md);  
}
```

```
.style-group label {  
  display: block;  
  margin-bottom: var(--spacing-xs);  
}
```

```
font-family: var(--accent-font);
font-size: 0.9rem;
}
```

```
.color-options {
  display: flex;
  gap: var(--spacing-xs);
}
```

```
.color-option {
  width: 24px;
  height: 24px;
  border-radius: 50%;
  border: 2px solid transparent;
  cursor: pointer;
}
```

```
.color-option.active {
  border-color: var(--rich-gold);
}
```

```
.font-select, .icon-select {
  width: 100%;
  padding: var(--spacing-xs) var(--spacing-sm);
  background-color: rgba(10, 25, 47, 0.5);
  border: 1px solid rgba(250, 243, 224, 0.2);
  border-radius: 4px;
  color: var(--magnolia-white);
  font-family: var(--accent-font);
}
```

```
.export-options {
  display: flex;
  gap: var(--spacing-sm);
}
```

```
.export-btn {
  flex: 1;
  padding: var(--spacing-sm);
}
```

```
background-color: rgba(250, 243, 224, 0.1);
border: none;
border-radius: 4px;
color: var(--magnolia-white);
font-family: var(--accent-font);
cursor: pointer;
transition: all 0.3s ease;
}
```

```
.export-btn:hover {
  background-color: rgba(250, 243, 224, 0.2);
}
```

```
/* Responsive */
@media (max-width: 1200px) {
  .stats-container {
    grid-template-columns: repeat(2, 1fr);
  }
}
```

```
.quick-buttons {
  grid-template-columns: repeat(2, 1fr);
}
}
```

```
@media (max-width: 992px) {
  .activity-tasks-container {
    grid-template-columns: 1fr;
  }
}
```

```
.preview-container {
  flex-direction: column;
}
```

```
.card-preview {
  margin: 0 auto;
}
}
```

```
@media (max-width: 768px) {
```

```

.dashboard-container {
  flex-direction: column;
}

.sidebar {
  width: 100%;
  padding: var(--spacing-md);
}

.stats-container {
  grid-template-columns: 1fr;
}

.content-area {
  flex-direction: column;
}

.content-sidebar {
  width: 100%;
  margin-bottom: var(--spacing-lg);
}

.style-options {
  grid-template-columns: 1fr;
}

```

4. JavaScript (public/js/content.js)

javascript

Copy

```

document.addEventListener('DOMContentLoaded', function() {
  // Template button functionality
  const templateButtons = document.querySelectorAll('.template-
button');
  const promptTextarea = document.getElementById('content-prompt');

  templateButtons.forEach(button => {
    button.addEventListener('click', function() {
      promptTextarea.value = this.getAttribute('data-template');
    });
  });
});

```

```
});  
});
```

```
// Generate button functionality (simulate AI response)
```

```
const generateBtn = document.getElementById('generate-btn');  
const resultsContainer = document.querySelector('.results-container');
```

```
generateBtn.addEventListener('click', function() {
```

```
  // Show loading state
```

```
  generateBtn.textContent = 'Generating...';
```

```
  generateBtn.disabled = true;
```

```
  // Simulate API call delay
```

```
  setTimeout(() => {
```

```
    // Reset button
```

```
    generateBtn.textContent = 'Generate Content';
```

```
    generateBtn.disabled = false;
```

```
    // Scroll to results
```

```
    resultsContainer.scrollTo({ behavior: 'smooth' });
```

```
  }, 1500);
```

```
});
```

```
// Copy functionality
```

```
const copyButtons = document.querySelectorAll('.result-action.copy');
```

```
copyButtons.forEach(button => {
```

```
  button.addEventListener('click', function() {
```

```
    const contentElement = this.closest('.result-
```

```
card').querySelector('.result-content');
```

```
    const text = contentElement.textContent;
```

```
    navigator.clipboard.writeText(text).then(() => {
```

```
      // Show temporary success indicator
```

```
      const originalIcon = this.innerHTML;
```

```
      this.innerHTML = '<span class="icon">✓</span>';
```

```
    }  
    setTimeout(() => {
```

```
      this.innerHTML = originalIcon;
```

```
    }, 1500);  
  });  
});  
});
```

```
// Card styling preview functionality
```

```
const colorOptions = document.querySelectorAll('.color-option');  
const cardPreview = document.querySelector('.card-preview');  
const cardText = document.querySelectorAll('.card-text');
```

```
// Background color
```

```
colorOptions.forEach(option => {  
  option.addEventListener('click', function() {  
    if (this.closest('.style-  
group').querySelector('label').textContent.includes('Background')) {  
      // Update active state  
      this.closest('.color-options').querySelectorAll('.color-  
option').forEach(opt => {  
        opt.classList.remove('active');  
      });  
      this.classList.add('active');
```

```
// Apply background color
```

```
cardPreview.style.backgroundColor =  
getComputedStyle(this).backgroundColor;  
} else if (this.closest('.style-  
group').querySelector('label').textContent.includes('Text')) {  
  // Update active state  
  this.closest('.color-options').querySelectorAll('.color-  
option').forEach(opt => {  
    opt.classList.remove('active');  
  });  
  this.classList.add('active');
```

```
// Apply text color
```

```
cardText.forEach(text => {  
  text.style.color = getComputedStyle(this).backgroundColor;  
});  
}
```

```
});  
});
```

```
// Font selection
```

```
const fontSelect = document.querySelector('.font-select');
```

```
fontSelect.addEventListener('change', function() {  
  const selectedFont = this.value;  
  let fontFamily;
```

```
  switch(selectedFont) {  
    case 'playfair':  
      fontFamily = 'Playfair Display, serif';  
      break;  
    case 'lora':  
      fontFamily = 'Lora, serif';  
      break;  
    case 'montserrat':  
      fontFamily = 'Montserrat, sans-serif';  
      break;  
    default:  
      fontFamily = 'Playfair Display, serif';  
  }
```

```
  cardText.forEach(text => {  
    text.style.fontFamily = fontFamily;  
  });  
});
```

```
// Energy level selection
```

```
const energyLevels = document.querySelectorAll('.energy-level');
```

```
energyLevels.forEach(level => {  
  level.addEventListener('click', function() {  
    energyLevels.forEach(el => el.classList.remove('selected'));  
    this.classList.add('selected');  
  });  
});  
});
```


5. Workflow Scripts

Content Generation API Integration (server/controllers/aiController.js)

javascript

Copy

```
const axios = require('axios');  
require('dotenv').config();
```

```
// This controller handles AI content generation requests  
// It will connect to OpenAI or other AI APIs based on your configuration
```

```
class AIController {  
  constructor() {  
    this.apiKey = process.env.OPENAI_API_KEY;  
    this.baseUrl = 'https://api.openai.com/v1/chat/completions';  
  }  
}
```

```
async generateContent(req, res) {  
  try {  
    const { prompt, contentType, tone } = req.body;  
  
    if (!prompt) {  
      return res.status(400).json({ error: 'Prompt is required' });  
    }  
  }  
}
```

```
// Craft the system message based on content type and tone  
let systemMessage = 'You are a helpful, creative assistant  
specializing in content creation.';
```

```
switch(contentType) {  
  case 'affirmation':  
    systemMessage = 'You are a specialized content creator focusing  
on affirmations rooted in Black Southern feminine energy. Your tone is  
poetic, healing, and empowering. Create content that reflects ancestral  
wisdom and resilience.';  
    break;  
  case 'tarot':  
    systemMessage = 'You are a tarot card interpreter with expertise
```

in connecting traditional tarot meanings with contemporary Black cultural icons. Your descriptions are insightful, meaningful, and culturally relevant.';

```
break;
```

```
case 'journal':
```

```
  systemMessage = 'You are a trauma-informed journal prompt creator specializing in healing, recovery, and self-discovery prompts. Your writing is gentle, introspective, and growth-oriented.';
```

```
  break;
```

```
case 'listing':
```

```
  systemMessage = 'You are an e-commerce copywriter specializing in SEO-optimized product listings for spiritual and wellness products. Your copy is compelling, authentic, and marketing-savvy.';
```

```
  break;
```

```
}
```

```
// Add tone modifications
```

```
if (tone && tone.length > 0) {
```

```
  systemMessage += ` Focus on being ${tone.join(', ')}`;
```

```
}
```

```
const response = await axios.post(
```

```
  this.baseUrl,
```

```
{
```

```
  model: 'gpt-4',
```

```
  messages: [
```

```
    { role: 'system', content: systemMessage },
```

```
    { role: 'user', content: prompt }]
```

```
,
```

```
  temperature: 0.8,
```

```
  max_tokens: 1000,
```

```
  n: 3 // Generate 3 options
```

```
},
```

```
{
```

```
  headers: {
```

```
    'Content-Type': 'application/json',
```

```
    'Authorization': `Bearer ${this.apiKey}`
```

```
  }
```

```
}
```

```

);

const results = response.data.choices.map(choice =>
choice.message.content);

return res.status(200).json({ results });
} catch (error) {
console.error('AI Generation Error:', error);
return res.status(500).json({
error: 'Error generating content',
details: error.response ? error.response.data : error.message
});
}
}

async generateImage(req, res) {
try {
const { prompt, style } = req.body;

if (!prompt) {
return res.status(400).json({ error: 'Prompt is required' });
}

// For image generation, we'll use DALL-E or similar
const response = await axios.post(
'https://api.openai.com/v1/images/generations',
{
prompt: prompt,
n: 1,
size: '1024x1024',
style: style || 'natural'
},
{
headers: {
'Content-Type': 'application/json',
'Authorization': `Bearer ${this.apiKey}`
}
}
);

```

```

    return res.status(200).json({
      imageUrl: response.data.data[0].url
    });
  } catch (error) {
    console.error('Image Generation Error:', error);
    return res.status(500).json({
      error: 'Error generating image',
      details: error.response ? error.response.data : error.message
    });
  }
}
}
}

```

module.exports = new AIController();

Inventory Management Workflow (routes/inventory.js)

javascript

Copy

```

const express = require('express');
const router = express.Router();
const Product = require('../models/Product');

// Get all products
router.get('/', async (req, res) => {
  try {
    const products = await Product.find().sort({ createdAt: -1 });
    res.render('inventory/index', { products });
  } catch (error) {
    console.error('Error fetching products:', error);
    res.status(500).send('Server error');
  }
});

// Add new product form
router.get('/add', (req, res) => {
  res.render('inventory/add');
});

```

// Create new product

```
router.post('/', async (req, res) => {
  try {
    const {
      name,
      type,
      description,
      price,
      imageUrl,
      productUrl,
      platform,
      tags
    } = req.body;

    const product = new Product({
      name,
      type,
      description,
      price: parseFloat(price),
      imageUrl,
      productUrl,
      platform,
      tags: tags.split(',').map(tag => tag.trim()),
      createdAt: new Date()
    });

    await product.save();

    res.redirect('/inventory');
  } catch (error) {
    console.error('Error creating product:', error);
    res.status(500).send('Server error');
  }
});
```

// Edit product form

```
router.get('/edit/:id', async (req, res) => {
  try {
```

```
const product = await Product.findById(req.params.id);
if (!product) {
  return res.status(404).send('Product not found');
}

res.render('inventory/edit', { product });
} catch (error) {
  console.error('Error fetching product:', error);
  res.status(500).send('Server error');
}
});
```

```
// Update product
router.post('/update/:id', async (req, res) => {
  try {
    const {
      name,
      type,
      description,
      price,
      imageUrl,
      productUrl,
      platform,
      tags,
      status
    } = req.body;
```

```
const product = await Product.findById(req.params.id);
if (!product) {
  return res.status(404).send('Product not found');
}
```

```
product.name = name;
product.type = type;
product.description = description;
product.price = parseFloat(price);
product.imageUrl = imageUrl;
product.productUrl = productUrl;
product.platform = platform;
```

```

    product.tags = tags.split(',').map(tag => tag.trim());
    product.status = status;
    product.updatedAt = new Date();

    await product.save();

    res.redirect('/inventory');
  } catch (error) {
    console.error('Error updating product:', error);
    res.status(500).send('Server error');
  }
});

```

```

// Delete product
router.post('/delete/:id', async (req, res) => {
  try {
    await Product.findByIdAndDelete(req.params.id);
    res.redirect('/inventory');
  } catch (error) {
    console.error('Error deleting product:', error);
    res.status(500).send('Server error');
  }
});

```

```
module.exports = router;
```

Database Models (models/Product.js)

javascript

Copy

```

const mongoose = require('mongoose');

const ProductSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true,
    trim: true
  },
  type: {
    type: String,

```

```
    required: true,
    enum: ['digital', 'physical', 'service'],
    default: 'digital'
  },
  description: {
    type: String,
    required: true
  },
  price: {
    type: Number,
    required: true
  },
  imageUrl: {
    type: String
  },
  productUrl: {
    type: String
  },
  platform: {
    type: String,
    enum: ['etsy', 'shopify', 'patreon', 'other'],
    default: 'other'
  },
  tags: [String],
  status: {
    type: String,
    enum: ['active', 'draft', 'archived'],
    default: 'active'
  },
  createdAt: {
    type: Date,
    default: Date.now
  },
  updatedAt: {
    type: Date
  }
});
```

```
module.exports = mongoose.model('Product', ProductSchema);
```


6. Replit Configuration (.replit)

Copy

```
entrypoint = "index.js"
```

```
modules = ["nodejs-18:v3-20230608-f4cd419"]
```

```
hidden = [".config", "package-lock.json"]
```

```
[nix]
```

```
channel = "stable-23_05"
```

```
[deployment]
```

```
run = ["node", "index.js"]
```

```
deploymentTarget = "cloudrun"
```

```
[languages]
```

```
[languages.javascript]
```

```
pattern = "**/{*.js,*.jsx,*.ts,*.tsx}"
```

```
[languages.javascript.languageServer]
```

```
start = "typescript-language-server --stdio"
```

```
[debugger]
```

```
support = true
```

```
[debugger.interactive]
```

```
transport = "localhost:0"
```

```
startCommand = ["dap-node"]
```

```
[debugger.interactive.initializeMessage]
```

```
command = "initialize"
```

```
type = "request"
```

```
[debugger.interactive.initializeMessage.arguments]
```

```
clientId = "replit"
```

```
clientName = "replit.com"
```

```
columnsStartAt1 = true
```

linesStartAt1 = true
locale = "en-us"
pathFormat = "path"
supportsInvalidatedEvent = true
supportsProgressReporting = true
supportsRunInTerminalRequest = true
supportsVariablePaging = true
supportsVariableType = true

[debugger.interactive.launchMessage]
command = "launch"
type = "request"

[debugger.interactive.launchMessage.arguments]
console = "externalTerminal"
cwd = "."
pauseForSourceMap = false
program = "./index.js"
request = "launch"
sourceMaps = true
stopOnEntry = false
type = "pwa-node"

[packager]
language = "nodejs"

[packager.features]
packageSearch = true
guessImports = true
enabledForHosting = false

[interpreter]
command = ["prybar-nodejs", "-q", "--ps1",
"\u0001\u001B[33m\u0002\u0001\u001B[00m\u0002 ", "-i"]

[unitTest]
language = "nodejs"

[env]

```
XDG_CONFIG_HOME = "$REPL_HOME/.config"
PATH = "$REPL_HOME/.config/npm/node_global/bin:$REPL_HOME/
node_modules/.bin"
npm_config_prefix = "$REPL_HOME/.config/npm/node_global"
```

```
[gitHubImport]
requiredFiles = [".replit", "replit.nix", ".config", "package.json",
"package-lock.json"]
```

```
[[hints]]
regex = 'Error \[ERR_REQUIRE_ESM\]'
message = "We see that you are using require(...) inside your code. We
currently do not support this syntax. Please use 'import' instead when
using external modules. (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/import)"
```

7. Required Dependencies (package.json)

json

Copy

```
{
  "name": "midnight-magnolia-dashboard",
  "version": "1.0.0",
  "description": "Content creation dashboard for Midnight Magnolia",
  "main": "index.js",
  "scripts": {
    "start": "node index.js",
    "dev": "nodemon index.js"
  },
  "dependencies": {
    "axios": "^1.5.0",
    "body-parser": "^1.20.2",
    "dotenv": "^16.3.1",
    "ejs": "^3.1.9",
    "express": "^4.18.2",
    "express-session": "^1.17.3",
    "mongoose": "^7.5.0",
    "multer": "^1.4.5-lts.1"
  },
  "devDependencies": {
```

```
"nodemon": "^3.0.1"
}
}
```

Next Steps

Now that you have a comprehensive setup for your Midnight Magnolia dashboard, here are the next steps to complete the implementation:

1. Create the MongoDB Database:

- Set up a MongoDB database (you can use MongoDB Atlas for a cloud-based solution)
- Create the necessary collections: products, content, clients, orders

2. Set Up OpenAI API Integration:

- Generate an API key from OpenAI
- Create a .env file with your API key and other sensitive information

3. Implement User Authentication:

- Add login/signup functionality for secure access
- Create user roles and permissions

4. Connect to Shopify:

- Integrate with the Shopify API to sync your inventory and orders
- Set up webhooks to automatically update your dashboard when orders are placed

5. Build the CRM Functionality:

- Develop the client management views
- Implement order tracking and history

6. Create Content Export Tools:

- Build exporters for different platforms (Etsy, Shopify, etc.)
- Add batch processing capabilities for multiple content items

7. Implement Energy-Based Task Scheduling:

- Develop the energy tracking system to prioritize tasks
- Create a calendar view that aligns with moon phases

This implementation provides an ADHD-friendly, Southern Gothic-styled dashboard that combines the content capabilities of Jasper AI with the client management of Honeybook and the e-commerce tracking needed for your Midnight Magnolia brand.