CHAPTER 3

# **SYSTEM REQUIREMENTS**

## **3.1 Functional Requirements**

## **3.1.1 User Functional Requirements**

**For Business Owners & Marketers:**

### **Content Input & Customization**

* 1. The system shall allow users to provide inputs such as topic, keywords, target audience, and preferred tone.
  2. The system shall support different objectives like brand awareness, lead generation, or direct sales, ensuring that businesses can adapt the tool to their specific goals.

### **Blog & Article Generation**

* 1. The system shall generate detailed, well-structured articles with headings, subheadings, and summaries.
  2. The system shall include SEO optimization by suggesting keyword placement, readability scores, and meta descriptions.

### **Social Media Content Creation**

* 1. The system shall generate short, engaging content suitable for platforms such as Instagram, LinkedIn, Facebook, and Twitter.
  2. The system shall recommend hashtags, best posting times, and call-to-actions to increase engagement.

### **Email Campaigns**

* 1. The system shall generate email content for newsletters, promotions, and customer follow-ups.
  2. The system shall suggest personalized subject lines and provide multiple variations for testing.

### **Product Description Writing**

* 1. The system shall generate compelling product descriptions that highlight product features and benefits.
  2. The tone of descriptions shall be adaptable (professional, casual, or creative) depending on the platform (Amazon, Shopify, Flipkart, etc.).

### **Advertisement Copywriting**

* 1. The system shall generate ad copy for platforms such as Google Ads and Meta Ads.
  2. The system shall ensure compliance with advertising policies, avoiding restricted words or formatting issues.

### **Content Export Options**

* 1. The system shall allow users to download generated content in multiple formats (PDF, DOCX, TXT).
  2. The system shall provide an option to connect accounts and directly publish or schedule posts on external platforms (WordPress, LinkedIn, Instagram).

## **3.1.2 Admin Functional Requirements**

### **User Management**

* 1. The system shall allow admins to create, update, or remove user accounts.
  2. The system shall provide role-based access, classifying users as Admin, Editor, or Standard User.

### **Content Monitoring**

* 1. The system shall allow admins to review flagged or low-quality content.
  2. The system shall provide options to approve, reject, or improve such content before final use.

### **AI Model Management**

* 1. The system shall allow admins to fine-tune and update AI models to improve accuracy.
  2. The system shall maintain version history so older models can be restored if needed.

### **Subscription & Payment Handling**

* 1. The system shall provide subscription plans (Free, Premium, Enterprise) for users.
  2. The system shall handle automatic billing, invoicing, and renewal reminders.

### **Analytics Dashboard**

* 1. The system shall provide admins with an analytics dashboard displaying active users, generated content, and overall usage.
  2. The system shall highlight areas where users face issues, helping to improve performance.

# **3.2 Non-Functional Requirements**

### 1. **Performance**

1. The system shall be able to generate a complete **blog article of 1000–1500 words** within 30–40 seconds using the inputs (keywords, audience, brand voice).
2. The system shall support **simultaneous generation** of at least 200 different pieces of content (blogs, social media posts, emails) without slowing down.
3. The system shall provide quick previews of generated content, with a **response time of less than 3 seconds** when switching between sections.

### 2. **Reliability**

1. The system shall ensure **99.5% uptime** so that businesses, marketers, and agencies can access the tool any time for campaign deadlines.
2. The system shall automatically **backup all generated content and user inputs**, ensuring no loss of data during failures.
3. The system shall include **error handling and recovery** so that if a task fails midway (e.g., long blog generation), it can be resumed instead of starting again.

### 3. **Security**

1. The system shall encrypt sensitive business data, including uploaded brand guidelines, customer information, and login credentials.
2. The system shall use **role-based access** so that Admins, Premium Users, and Free Users have proper permissions.
3. The system shall comply with **data protection laws** like GDPR and CCPA, since it may handle customer-related marketing data.
4. The system shall support secure login with **two-factor authentication (2FA)** for paid business accounts.

### 4. **Scalability**

1. The system shall be capable of **handling more users** when the platform grows (e.g., from 1000 to 10,000 daily users).
2. The system shall allow **new modules** (like AI video scripts or influencer campaign content) to be added without redesigning the whole system.

### 5. **Usability**

1. The system shall have a **clean and simple dashboard** where users can choose what they want to generate (blog, social post, email, etc.) without confusion.
2. The system shall provide **guidance and tooltips** (like “Add 5–10 keywords for better SEO”) to help beginners.
3. The system shall support **mobile-friendly design**, so that marketers can generate posts quickly even on their phones.
4. The system shall be **accessible for differently-abled users**, with readable fonts, proper contrast, and voice support.

### 6. **Internationalization**

1. The system shall support **multiple languages** for generating content (e.g., English, Hindi, Spanish) to help businesses work globally.
2. The system shall adapt **SEO and formats** based on country-specific marketing needs (like US Google Ads vs. Indian Flipkart product descriptions).

### 7. **Maintainability**

1. The system shall use **modular design** (separate modules for Blog, Social Media, SEO, etc.) so that one can be updated without affecting others.
2. The system shall maintain **error logs and activity reports** to track bugs and issues.
3. The system shall support **regular AI model updates** without causing downtime for users.

# **3.3 Software and Hardware Requirements**

## **3.3.1 Software Requirements Specification (SRS)**

### **1. Server-Side Requirements**

#### **Programming Language & Framework:**

* + **FastAPI (Python)** – For backend REST API development.

#### **Database:**

* + **PostgreSQL** – Primary database for structured content and user data.
  + **Redis** – For caching, session management, and faster content retrieval.
  + **Pinecone (Vector Database)** – For semantic search, content embedding, and AI-driven recommendations.

#### **AI Engine:**

* + **OpenAI GPT-4 API or Hugging Face Transformers** – For content generation.
  + **LangChain** – For orchestrating AI workflows and prompt chaining.

#### **Authentication:**

* + **Firebase Authentication** – For secure user sign-in/sign-up with support for email, Google, and social logins.

#### **Cloud Infrastructure:**

* + **AWS ECS/EKS** – Container orchestration and scaling.
  + **AWS S3** – For storing media, generated content, and assets.
  + **AWS CloudWatch** – For monitoring and logging.

### **2. Client-Side Requirements**

#### **Frontend Framework:**

* + **Streamlit** – For building an interactive and user-friendly web interface.

#### **Supported Browsers:**

* + Chrome, Firefox, Edge, Safari (latest versions).

#### **Features:**

* + User dashboard for campaign creation and content management.
  + Input forms for content parameters (keywords, tone, platform).
  + Real-time preview of generated content.

### **3. Mobile Support**

#### **Responsive Web App:**

* + Mobile-friendly design via Streamlit with responsive layout.

#### **Optional Future Scope:**

* + Cross-platform mobile app using **React Native** or **Flutter**, integrated with the same backend APIs

### **4. Development Tools**

#### **IDE/Code Editors:**

* + Visual Studio Code, PyCharm.
* **Version Control:**
  + Git + GitHub/GitLab.

#### **Containerization:**

* + Docker – For packaging backend services.

#### **CI/CD Pipeline:**

* + GitHub Actions / GitLab CI for automated testing and deployment.

#### **API Testing:**

* + Postman or Insomnia.

#### **Dependency Management:**

* + Poetry / Pipenv for Python.

#### **Monitoring & Analytics:**

* + Prometheus + Grafana (for performance monitoring).

### **5. Additional Requirements**

#### **Security:**

* + HTTPS with SSL/TLS.
  + JWT-based token authentication (managed via Firebase).

#### **Scalability:**

* + Load balancing using AWS Application Load Balancer.

#### **SEO Tools Integration:**

* + Google Search Console API, Ahrefs/Semrush (optional).

# **3.3.2 Hardware Requirements**

### **1. Server-Side (for deployment & AI processing)**

* **Processor:** Minimum 8-core CPU (Intel Xeon/AMD EPYC); Recommended: NVIDIA GPU (A100/T4) for model training/fine-tuning
* **RAM:** Minimum 16 GB; Recommended 32–64 GB
* **Storage:** Minimum 500 GB SSD; Recommended 1 TB NVMe SSD
* **Bandwidth:** High-speed internet with low latency (1 Gbps or more for scaling)

### **2. Client-Side (end users)**

* **Processor:** Dual-core CPU or higher
* **RAM:** 4 GB (8 GB recommended for heavy browser use)
* **Storage:** 500 MB (for browser cache and light offline storage)
* **Browser:** Latest version of Chrome, Firefox, or Edge

### **3. Development Environment (Optional for training/fine-tuning models)**

* NVIDIA GPU (RTX 3090 / A100 for large-scale model fine-tuning)
* 64–128 GB RAM for large dataset handling
* High-performance SSD storage for datasets and checkpoints