Backend Requirement Specification for Marketing Content Lab

Technology Stack:

- Backend Language: Node.js

- Database: MongoDB

- Architecture: MVC (Model-View-Controller)

1. Overview

This backend will power the Marketing Content Lab platform, providing API endpoints and services

that support:

- Character-based editing modes

- Vertical market customization

- User preferences and customization

- Preset and configuration management

- Analytics and content feedback

2. API Requirements

2.1 User Management

- POST /api/register

  - Register a new user

  - Payload: { name, email, password }

- POST /api/login

  - Authenticate a user and return a JWT token

  - Payload: { email, password }

- GET /api/user/profile

  - Retrieve user profile (requires auth)

- PUT /api/user/profile

  - Update user profile info

  - Payload: { name, password (optional), preferences }

2.2 Character Modes

- GET /api/characters

  - List all available character modes

- GET /api/characters/:id

  - Retrieve details of a specific character mode

- POST /api/user/character

  - Set preferred character mode for a user

  - Payload: { characterId }

2.3 Preset Bundles & Settings

- GET /api/presets

  - Retrieve all configuration presets (used by characters)

- GET /api/presets/:id

  - Retrieve specific preset details

- POST /api/user/preset

- Assign a specific preset to a user

- Payload: { presetId }

- PUT /api/user/custom-settings

  - Update user-defined custom settings

  - Payload: { grammar, tone, structure, etc. }

2.4 Vertical Market Modules

- GET /api/verticals

  - List all available vertical market configurations

- GET /api/verticals/:id

  - Retrieve details of a specific vertical (jargon, style guide, etc.)

- POST /api/user/vertical

  - Set user's preferred industry vertical

  - Payload: { verticalId }

2.5 Content Editing & Feedback

- POST /api/edit-content

  - Submit content for enhancement via Prose Perfector

  - Payload: { text, characterId, verticalId (optional), customSettings }

- POST /api/feedback

  - Receive analytics and suggestions on submitted content

  - Payload: { contentId, feedbackType }

2.6 Admin Panel APIs

- POST /api/admin/characters

  - Add or update a character mode (Admin only)

- POST /api/admin/presets

  - Create or modify preset bundles (Admin only)

- POST /api/admin/verticals

  - Manage vertical market modules (Admin only)

3. Database Models (MongoDB)

User:

- name

- email

- password (hashed)

- characterId

- verticalId

- customSettings: {}

- presets: []

Character:

- name

- description

- features: [String]

- presetId

Preset:

- name

- settings: {}

VerticalMarket:

- name

- lexicon: [String]

- styleGuide: String

- complianceNotes: String

- features: [String]

Content:

- userId

- originalText

- editedText

- usedCharacter

- usedPreset

- usedVertical

- analytics: {}

4. Authentication & Middleware

- JWT-based authentication for protected endpoints

- Middleware for:

  - Auth check

  - Role-based access (admin/user)

  - Input validation

5. Future Considerations

- Versioning of character/preset configurations

- Analytics dashboard for user progress

- Multi-language support

- Integration with corporate style guides via upload or API

6. Deployment

- Environment: Node.js runtime (Docker container recommended)

- Use environment variables for DB and secret keys

- MongoDB Atlas or self-hosted MongoDB instance

- CI/CD with GitHub Actions or similar

Prepared for client proposal and implementation planning.