WordPress Developer Technical Assessment

Advanced REST API & Automation Task

Estimated Time: 3-4 days
Difficulty: Intermediate-Advanced
Focus Areas: REST API, Advanced Authentication, Automation, Webhooks

🎯 Task Overview

Create a WordPress plugin that can automatically create Bulk pages via a secure REST API endpoint accessible from external applications, with advanced authentication and webhook notifications.

📋 What You Need to Build

A plugin called **"Simple Page Builder"** that can create bulk pages with the following features:

1. **Secure REST API endpoint** accessible from external applications
2. **Advanced API authentication system** (API Keys with token-based auth)
3. **Admin interface** for API key management and monitoring
4. **Webhook system** to notify external services when pages are created

⚠️ **IMPORTANT REQUIREMENTS:**

- **The REST API must be accessible from outside WordPress** - Any external application with valid API credentials should be able to use it
- **NO basic username/password authentication** - Implement proper API key-based authentication
- **Security is critical** - The authentication system must be production-ready and secure

🔧 Requirements

Feature 1: REST API Endpoint (Main Feature)

Endpoint: `POST /wp-json/pagebuilder/v1/create-pages`

🔒 **Authentication Requirements:**

This endpoint MUST be accessible from external applications (outside WordPress admin).

Implement an **API Key authentication system** where:

- API keys can be generated through the WordPress admin interface
- Each API key has a corresponding secret key
- Authentication uses the API key in request headers
- API keys can be revoked/regenerated
- API keys have optional expiration dates
- All API requests are logged with the API key used

Feature 2: API Key Management System

Create a secure system for generating and managing API keys:

API Key Structure:

Each API key should have:

- **API Key:** A unique, randomly generated string (e.g., 32-64 characters)
- **Key Name:** A friendly name for identification (e.g., "Production Server", "Mobile App")
- **Status:** Active or Revoked
- **Created Date:** When the key was generated
- **Expiration Date:** Optional expiration (can be never)
- **Last Used:** Timestamp of last successful request
- **Request Count:** Total number of requests made with this key
- **Permissions:** What the key can do (for this task, just "create_pages")

Security Requirements:

- API keys must be stored securely (hashed in database, similar to passwords)
- Show the API key only once when generated (cannot be retrieved later)
- Implement rate limiting per API key (e.g., 100 requests per hour)
- Log all API key usage (successful and failed attempts)
- Allow administrators to revoke keys instantly

Storage:

Store API keys in WordPress database. You can choose:

- Custom database table (recommended for scalability)
- WordPress options table (acceptable for this task)

Feature 3: Admin Interface

Create an admin page under **Tools** → **Page Builder**

Page Sections:

1. API Keys Management Tab:

- **Generate New API Key** button/form:
  - Input: Key Name (required)
  - Input: Expiration Date (optional)
  - On generate: Show the key ONCE with copy button
  - Warning: "Save this key securely, you won't see it again"
- **API Keys List** table showing:
  - Key Name
  - Key Preview (first 8 chars + "***)
  - Status (Active/Revoked badge)
  - Created Date
  - Last Used
  - Request Count
  - Actions: Revoke button, View Details

2. API Activity Log Tab:

- Table showing recent API requests:
  - Timestamp
  - API Key Used (preview)
  - Endpoint
  - Status (Success/Failed)
  - Pages Created
  - Response Time
  - IP Address
- Filter by: Status, Date Range, API Key
- Export logs as CSV

3. Created Pages Tab:

- Table showing pages created via API:
  - Page Title
  - URL (clickable link)
  - Created Date
  - Created By (API Key name)

4. Settings Tab:

- Default webhook URL
- Rate limit settings (requests per hour per key)
- Enable/disable API access globally
- API key expiration default (30, 60, 90 days, never)

5. API Documentation Tab:

- Display clear documentation on how to use the API
- Include cURL examples with placeholder for API key
- Show the API endpoint URL
- Explain authentication method
- Show request/response examples

Feature 4: Webhook Notification

When pages are created, send a POST request to the webhook URL:

Webhook Payload:

```
{
  "event": "pages_created",
  "timestamp": "2025-10-07T14:30:00Z",
  "request_id": "req_abc123xyz",
  "api_key_name": "Production Server",
  "total_pages": 3,
  "pages": [
    {
      "id": 123,
      "title": "About Us",
      "url": "http://site.com/about"
    },
    {
      "id": 124,
      "title": "Contact",
      "url": "http://site.com/contact"
    }
  ]
}
```

Webhook Security:

- Include signature header: `X-Webhook-Signature`
- Use HMAC-SHA256 for signature generation
- Signature should be based on webhook secret (configurable in settings)
- Include instructions in documentation on how to verify webhook signature

Requirements:

- Use `wp_remote_post()` to send webhook
- Implement retry logic (2 retries with exponential backoff)
- Log webhook delivery status (success/failed)
- 10 second timeout
- Handle errors gracefully (don't fail page creation if webhook fails)

📦 What to Submit

🔗 **Git Repository Requirement:**

You MUST submit your plugin as a public Git repository.

- Create a public repository
- Include all plugin files
- Write a comprehensive README.md
- Use meaningful commit messages
- **Submit the repository URL, NOT a zip file**

🎁 Bonus Points (Optional)

Additional features that will impress us:

- **OAuth 2.0 implementation** instead of API keys **OR**
  JWT (JSON Web Tokens) for authentication
- **Postman collection** for API testing

📧 Submission Instructions

How to Submit:

1. **Create a public Git repository** (GitHub/GitLab/Bitbucket)
2. **Push all your code** with clear commit messages
3. **Write comprehensive README.md** with all documentation
4. **Test your plugin thoroughly** before submitting
5. **Submit the repository URL** via email

Email to: wordpress@thewebops.com
Subject: WordPress Developer Assessment - [Your Name]
Deadline: 1 Week

🎯 Final Notes

This task is designed to assess your ability to:

- Build secure, production-ready WordPress functionality
- Implement proper authentication for external API access
- Follow security best practices
- Write clean, maintainable code
- Document your work clearly
- Use Git for version control

We're looking for quality over quantity. A well-implemented core feature is better than many half-finished features.

Good Luck! 🚀

We're excited to see your solution and look forward to reviewing your work.

Questions? Contact us at: wordpress@thewebops.com