### ****Project: Digital Dropshipping Platform for Keys****

#### ****1. Introduction****

This system will allow users to import stock from any marketplace using its export API and automatically resell it on the same or other marketplaces with price adjustments and automatic synchronization.

#### ****2. Key Features****

* Integration with APIs from any marketplace that supports stock export.
* Centralized stock management with automatic synchronization.
* Automatic price adjustment with custom rules.
* Digital dropshipping within the same or other marketplaces.
* Automation of sales and key deliveries.
* Monetization through per-key sales fees or monthly subscriptions.

#### ****3. Technical Architecture****

* **Backend:** Node.js (Express) or Django/Python
* **Database:** PostgreSQL / MongoDB
* **Frontend:** React.js or Vue.js
* **Infrastructure:** AWS / DigitalOcean / Google Cloud
* **Payments:** Stripe / PayPal

#### ****4. General API Endpoints****

**User Registration:**  
POST /api/auth/register

**Login:**  
POST /api/auth/login

**Connect Marketplace API:**  
POST /api/providers/connect

**Retrieve Available Stock:**  
GET /api/stock

**Automatically Adjust Prices:**  
POST /api/stock/update-prices

**Publish Stock on Another Marketplace:**  
POST /api/marketplaces/publish

**Publish the Same Stock on G2A with a Different Account:**  
POST /api/marketplaces/publish

**Automatically Reduce Stock:**  
POST /api/sales/update-stock

**Automatic Key Delivery:**  
POST /api/orders/deliver

#### ****5. Monetization Model****

* **Option 1:** Pay-per-key sold (e.g., $0.10 per key).
* **Option 2:** Monthly subscription with access to different tiers.

#### ****6. Development Roadmap****

* **Phase 1:** Backend - Authentication, supplier API integration, stock synchronization.
* **Phase 2:** Frontend - Dashboard and user panel.
* **Phase 3:** Marketplace integrations (Gamivo, Amazon, etc.).
* **Phase 4:** Payment and billing implementation.
* **Phase 5:** Beta testing and optimization.

Hello Tony,

We have conducted a full review of the project, including backend, frontend, and API implementations. After two weeks (half of the agreed four-week period), the progress is highly unsatisfactory. Below is a detailed assessment:

Backend Issues:

APIs Implemented: User, Order, Product, and API Management endpoints are functional, but they are not connected with the frontend.

Missing APIs: There are no endpoints for subscription management (/api/subscription) or dashboard metrics (/api/metrics).

Marketplace Integration: No progress on integrating Amazon or Gamivo marketplaces.

Security Issues: Routes such as /profile and /admin/users are accessible without proper authentication due to errors in auth.js.

Lack of Rate Limiting: There is no protection against brute-force attacks.

Frontend Issues:

API Management Section: The APIIntegration.jsx component is broken and cannot connect to the backend due to issues with axios.post.

Dashboard Issues: Dashboard.jsx displays hardcoded static data instead of fetching real metrics from the backend.

No Subscription Management: There is no interface for managing user plans or subscriptions.

Broken Checkout Process: The payment process in Payment.jsx fails with a server error (500).

Profile Security Flaw: Profile.jsx does not verify authentication tokens, making it possible to view user profiles without logging in.

No Responsive Design: CSS files such as Dashboard.css and Home.css lack media queries, making the site unusable on mobile devices.

Missing Product Publishing: NewProduct.jsx is incomplete and cannot add products.

Poor Form Validation: Forms such as SignUp.jsx and NewAPI.jsx lack proper input validations.

Middleware and Security Issues:

Broken Authentication Middleware: The auth.js file fails to correctly validate tokens, allowing unauthorized access.

No Rate Limiting: The project does not implement any protection against brute-force attacks (express-rate-limit is not configured).

Insufficient Error Handling: The error middleware captures only basic errors and lacks proper logging.

Immediate Action Required (Next 5 Days):

To ensure progress, we require the following actions within the next 5 days:

Code Submission: Upload the complete project to Workana/GitHub immediately.

Demo: Provide a live demonstration of functional API endpoints and a working dashboard.

Backend Fixes:

Implement /api/subscription endpoints for managing user plans.

Implement /api/metrics for real-time dashboard analytics.

Secure routes in auth.js and fix token validation issues.

Add at least one marketplace integration (Amazon or Gamivo).

Implement rate limiting to prevent brute-force attacks.

Frontend Fixes:

Fix APIIntegration.jsx to connect properly to backend endpoints.

Update Dashboard.jsx to show real data from /api/metrics.

Complete NewProduct.jsx to allow product creation.

Add proper authentication protection to Profile.jsx.

Make Dashboard.css and Home.css fully responsive.

Implement proper input validation on all forms. Admin