Build Insurance AI MVP using AWS Bedrock & Anthropic Claude

About the Project
We're building Dark Mist, a patent-pending AI system for commercial insurance field underwriting. We need an experienced developer to build our MVP web application using Amazon Bedrock with Anthropic Claude.

What We're Building: A web application that allows insurance underwriters to upload facility photos and receive AI-powered risk assessments in seconds (currently takes 2-4 hours manually).

Current Status:
AWS account active with Bedrock access enabled ✅
Anthropic Claude Sonnet 4.5 approved ✅
Product requirements documented ✅
Patent applications filed (3 provisional patents) ✅
Pilot customers identified ✅

Your Mission: Build a working MVP that demonstrates the core functionality and can be shown to pilot customers within 3 weeks.

Technical Requirements

Core Functionality
Feature 1: Image Upload & Analysis
Simple web interface for uploading facility photos (JPG/PNG)
Integration with Amazon Bedrock (Claude Sonnet 4.5 via boto3)
AI analyzes image for insurance risks across 4 coverage types:
General Liability (slip/fall, equipment hazards)
Commercial Auto (vehicle fleet, loading safety)
Pollution Liability (fluid management, EPA compliance)
Property (fire hazards, building condition)
Return structured JSON with risk scores and recommendations

Feature 2: Risk Assessment Display
Display overall risk score (1-10) with color coding
Show detected hazards by category
List safety equipment status
Provide improvement recommendations

Flag potential compliance issues

Feature 3: PDF Report Generation
Auto-generate professional inspection report
Include uploaded image with AI annotations
Multi-coverage risk breakdown
Recommendations and action items
Company branding (Dark Mist logo placeholder)

Feature 4: Demo Mode
Pre-loaded sample facilities for instant demos
No API calls required for demo (cost savings)
Shows full workflow to prospects

Technology Stack

Required:
Python 3.9+ (backend)
Amazon Bedrock (Anthropic Claude Sonnet 4.5)
AWS Lambda (serverless functions)
Amazon S3 (image storage)
Flask or FastAPI (web framework)
ReportLab or similar (PDF generation)

Preferred (Not Required):
AWS Rekognition Custom Labels (computer vision for hazard detection)
Simple frontend (HTML/CSS/JS or React - mobile-friendly)
AWS API Gateway (RESTful API)
Basic authentication (login/password)

Deliverables
Week 1:
Project structure and AWS infrastructure setup
Bedrock integration working (can analyze single image)
Basic web interface (upload + results display)
Week 2:
Multi-coverage risk scoring implemented
PDF report generation working
Demo mode functional
Basic styling/UX improvements

Week 3:
Deploy to AWS (Lambda + API Gateway)
Testing and bug fixes
Documentation (how to run, deploy, extend)
Handoff session (1 hour walkthrough)

Final Deliverables:
✅ Working web application deployed on AWS
✅ Source code in GitHub repository
✅ Documentation (README, deployment guide, API docs)
✅ 2 rounds of revisions based on feedback
✅ 1 hour handoff/training session

Developer Requirements
Must Have:
✅ Proven experience with AWS (Bedrock, Lambda, S3)
✅ LLM/AI API integration experience (OpenAI, Anthropic, etc.)
✅ Python backend development (Flask/FastAPI)
✅ RESTful API design
✅ AWS serverless architecture (Lambda, API Gateway)
✅ English communication skills (weekly check-ins required)

Nice to Have:
🌟 Experience with insurance/fintech applications
🌟 Computer vision experience (AWS Rekognition)
🌟 Experience with Constitutional AI / responsible AI
🌟 Frontend development (React, Vue, or similar)
🌟 PDF generation libraries (ReportLab)

Red Flags (Will Reject):
❌ No AWS experience
❌ No LLM/AI API experience
❌ Copy-paste portfolio (not original work)
❌ Poor English communication
❌ Unavailable for weekly check-ins

Project Timeline & Budget

Duration: 3 weeks (fixed deadline)
Budget: $5,000 - $10,000 USD (fixed price, not hourly)

Payment Structure:
25% upfront (upon project start)
35% at Week 2 milestone (core features working)
40% upon final delivery and acceptance

Start Date: Immediate (within 3-5 days of hiring)

Availability Required:
Weekly 30-minute check-in call (Slack or Zoom)
Async communication via Slack for questions
Responsive (replies within 24 hours on weekdays)

Success Criteria
This project is successful when:
✅ We can upload a facility photo and get an AI risk assessment in <10 seconds
✅ The risk assessment is insurance-relevant and accurate
✅ We can generate a professional PDF report
✅ We can demo to pilot customers confidently
✅ Code is clean, documented, and maintainable
✅ Deployed on AWS and accessible via URL

Bonus Points:
Application is mobile-friendly (underwriters use phones in field)
Loading states and error handling are polished
Demo mode is impressive for sales presentations
You suggest improvements we haven't thought of

Questions?
If you have questions before applying, please include them in your proposal. Common questions:
Q: What AWS services do you already have set up? A: AWS account active, Bedrock access enabled, Claude Sonnet 4.5 approved. You'll get IAM credentials with appropriate permissions.
Q: Do you have sample facility images for testing? A: Yes, we have 20+ sample images from various facility types (auto dismantlers, repair shops, warehouses).
Q: What does the risk assessment JSON structure look like? A: We'll provide detailed specs, but generally: risk_score (1-10), hazards_detected (array), safety_equipment (object), recommendations (array), compliance_issues (array).

Q: Do you need mobile app or just web? A: MVP is web-only (mobile-responsive). Native mobile app is Phase 2 (future project).
Q: Will you provide the AI prompts for Claude? A: Yes, we have detailed prompts for the risk assessment. You'll integrate them, not write them from scratch.

How We'll Work Together
Communication:
Primary: Slack (we'll add you to Dark Mist channel)
Weekly calls: 30 minutes via Zoom/Google Meet
Code reviews: GitHub pull requests
Project management: Simple (we're not bureaucratic)

Our Working Style:
We're founders who value speed and results over process
We give clear feedback quickly (within 24 hours)
We respect your expertise - you're the technical lead
We're available for questions but won't micromanage

What We Provide:
Clear requirements and user stories
Sample facility images and test data
AWS credentials and access
Quick feedback on demos/prototypes
Domain expertise (insurance underwriting knowledge)

What We Expect:
Proactive communication (flag issues early)
Weekly progress updates (even if just "on track")
Clean, documented code
Ownership mentality (suggest improvements)
Meet the 3-week deadline