Ideal Candidate Profile

(Full-Stack Developer with AI Interest and General AI Experience)

Core Skills and Experience:

Full-Stack Development:
Strong expertise in front-end and back-end development.
Proficiency in React, React Native, or similar JavaScript frameworks.

AI Development:
Interest and experience in AI integration with full-stack projects.
Familiarity with Python for AI and JavaScript for web development.

Key Qualities:

Curiosity and Learning:
Naturally curious and eager to learn new technologies.
Quick to pick up new skills and integrate them effectively.

Problem-Solving and Adaptability:
Strong problem-solving skills and adaptability.
Ability to connect diverse technical areas and find unique solutions.

Professionalism and Reliability:
Dependable with a professional attitude.
Sets realistic expectations and understands personal strengths and weaknesses.

Timeliness and Quality Focus:
Delivers high-quality work on time.
Pays attention to detail and maintains high standards.

This profile seeks a technically proficient candidate in full-stack and AI development, with a strong learning mindset, adaptability, reliability, and a focus on quality and timeliness.

TASK (12-14 hours):

Problem Task for Full Stack Web Developer with Basic AI Knowledge: Blog Project

Create a full-stack blog application that allows users to create, read, update, and delete blog posts. The application should also include an AI-powered feature to generate blog post summaries.

Task Details:

Frontend:
Framework: Use a modern JavaScript framework/library like React, Vue, or Angular.
User Interface:
Home Page: Display a list of all blog posts with their titles and summaries.
Post Page: Display the full content of a selected blog post with options to edit or delete.
Create Page: Provide a form to create a new blog post.
Edit Page: Provide a form to update an existing blog post.
Routing: Implement routing to navigate between different pages.

Backend:
Server: Set up a server using Node.js and Express.
Database: Use a database like MongoDB, PostgreSQL, or MySQL to store blog posts.

API Endpoints:
POST /posts: Create a new blog post.
GET /posts: Retrieve all blog posts.
GET /posts/:id: Retrieve details of a specific blog post.
PUT /posts/:id: Update a specific blog post.
DELETE /posts/:id: Delete a specific blog post.

AI Integration:
Summary Generation: Implement a basic AI model to generate summaries of blog posts.
Libraries: Utilize a library like OpenAI's GPT-3 or a similar text summarization tool.
Data Extraction: Extract relevant summary information from the blog content.

CRUD Operations:
Create: Allow users to create new blog posts and generate summaries.
Read: Display a list of all blog posts and allow viewing of full content for each post.
Update: Provide functionality to update the content of an existing blog post.
Delete: Allow users to delete blog posts.

Detailed Steps:

Frontend Implementation:
Set up the project using Create React App, Vue CLI, or Angular CLI.
Create components for Home, Post, Create, and Edit pages.
Implement forms for creating and editing blog posts.
Use Axios or Fetch API to communicate with the backend API.
Style the application using CSS, Bootstrap, or Material-UI.

Backend Implementation:
Set up a Node.js project with Express.
Connect to a database (MongoDB, PostgreSQL, or MySQL).

Define Mongoose models or Sequelize schemas for storing blog posts.
Implement API endpoints for handling CRUD operations.
Integrate the AI model for generating summaries in the POST /posts endpoint.

AI Integration:
Install necessary AI libraries (e.g., OpenAI's GPT-3).
Implement a function to generate summaries from blog content.
Integrate this function in the blog post creation API endpoint.

Testing and Deployment:
Test the application thoroughly to ensure all CRUD operations work as expected.
Deploy the frontend and backend applications to a cloud platform (e.g., Heroku, Vercel, AWS).
Ensure the application is secure and performs efficiently.

Skills to be Judged based on Task

Frontend Development
Framework Proficiency: Use of React, Vue, or Angular.
Routing and Navigation: Implementing navigation between pages.
Form Handling: Managing forms for creating and editing posts.

Backend Development
Server Setup: Using Node.js and Express.
Database Management: Connecting and managing a database (MongoDB, PostgreSQL, MySQL).
API Development: Implementing CRUD API endpoints.

AI Integration
AI Libraries: Using libraries like OpenAI's GPT-3 for text summarization.
AI Model Implementation: Generating summaries from blog content.
Backend Integration: Integrating AI functions in API endpoints.

Based on the submission there will be a technical round and a HR round interview.