**Agent Sketch User Stories**

### **User Story 1:Onboarding & Logging in Securely**

As a user, I want to log in securely with my credentials, so that my access to the system is safe, and my personal and organizational data is protected.

**Acceptance Criteria:**

1. User can log into the system using secure authentication mechanisms.
2. User is provided with a welcome message.
3. Concurrent access by multiple users is supported without performance degradation.

Tasks:

1. UI development for Login and Signup page
2. Backend Development for authentication

### **User Story 2: Getting Images**

As a user, I want to be able to ask my queries in natural language so that I can receive responses in the form of images/icons/illustrations.

**Acceptance Criteria:**

1. The user can type questions or queries in plain language (e.g., "I want images for animals").
2. The system understands and processes the query without needing special terms or complex commands.
3. The user receives a relevant response based on it.

### **Tasks:**

|  |  |  |  |
| --- | --- | --- | --- |
| Category | Task | Resource | Timeline |
|  |  | AI |  |
| Frontend | UI development of page  Integration with backend | UI |  |
| Deployment | Application Deployment | infra |  |

### **User Story 3: Accessing Historical Responses (Chat History)**

As a user, I want to save and review my past interactions with the system, so that I can access previous responses or information I retrieved earlier, without having to ask again.

**Acceptance Criteria:**

1. The user can view a history of previous questions they asked and the answers they received.
2. The user can easily search through past interactions to find specific information they’ve requested before.
3. The user can click on or open past responses to read them again or ask to follow up questions.

|  |  |  |
| --- | --- | --- |
| Category | Task | Resource |
| Data Pipeline | Storing Response in database | DE |
| Frontend | Integration with frontend | UI |

### **User Story 4: Download Generated Images**

As a user, I want to see the source of the information that Agent AI provides, so that I can download image and save it in my device.

**Acceptance Criteria:**

1. Generated image can be downloaded in our device.

|  |  |  |
| --- | --- | --- |
| Category | Task | Resource |
| Data Pipeline | Embedding Links while vectordb creation | DE |
| Database (MinIO/S3) | To store the image | AI |
| Frontend | UI development of page  Integration with backend | UI |
| Deployment | Application Deployment | infra |

### **User Story 5: UI Design**

As a UI/UX designer, I want to develop a user interface based on approved user flow, so that users can interact with Agent AI in an intuitive and user-friendly way.

**Acceptance Criteria:**

1. The UI is based on the approved user flow and is intuitive to use.
2. Users can easily input queries and receive responses from Agent AI.
3. UI provides a clean, organized layout, with easy access to chat history and settings.

|  |  |  |
| --- | --- | --- |
| Category | Task | Resource |
| Frontend | UI development | fE |
| integration | Backend integration | fe |
| Deployment | Deployment of the features | infra |

**Development**

### **User Story 1:Onboarding & Logging in Securely**

### Backend Implementation

Using FastAPI ,

* Set up authentication endpoints (signup, login, logout).
* Implement OAuth/JWT-based authentication.
* Store user details securely with hashing (e.g., bcrypt for passwords).
* Implement concurrent user session handling.

### **User Story 2: Getting Images**

### Step 1

* **Data Collection & Preprocessing (IN PROGRESS)**
* **Labeling & Metadata Creation**
* **Image Preprocessing**

### Step 2

**Fine-Tuning the Stable Diffusion Model:** We can start with a pre-trained Stable Diffusion model like SD3 , SD small v1 . We will select a model checkpoint that best fits the organization's use case.We will implement and choose from the following finetuning techniques

* **LoRA** (Low-Rank Adaptation): Efficient tuning with fewer parameters, suitable for style-specific modifications.
* **DreamBooth**: Helps the model learn new object identities and fine-tune on specific branding.
* **Textual Inversion**: Adds embeddings for new visual styles rather than modifying the entire model.

**Prepare Training Pipeline**

Hardware Setup: Use s for faster training

* Software Stack: Use PyTorch, Hugging Face Diffusers, and Accelerate for efficient training.
* Training Dataset Formatting: Convert images and captions into formats compatible with Stable Diffusion training.

**Train the Model:** Define the hyperparameters like learning rate, batch size, GPU memory, steps/epochs and train the model using mixed precision (FP16) for faster processing. Save checkpoints regularly to avoid losing progress.

**Model Evaluation & Iteration:** Human Evaluation: Compare generated images with real company assets.Fine-Tune Further: Adjust hyperparameters and continue training if needed.

Step 3

**Set Up Fast API Backend:** Deploying the Fine-Tuned Model in FastAPI. Convert model for Efficient Inference. Reduce model size while maintaining quality for faster response time.

* Create an API with the following endpoints:
* POST /generate-image: Accepts a text query and returns a generated image.
* GET /history: Retrieves previously generated images.

### **User Story 4: Download Generated Images**

As a user, I want to see the source of the information that Agent AI provides, so that I can download image and save it in my device.

**AI**

**Steps:**

Set up MinIO/S3 for storing generated images.

Generate signed URLs (valid for a specific period) if download links should expire.

Store this URL in VectorDB to link it with the query.