**📌 Project Overview**

The **Banking Bot** is an **AI-powered virtual assistant** developed using **AWS Lex**, which allows users to:  
✅ Check account balances  
✅ Transfer funds securely  
✅ Get account details  
✅ Provide banking-related information

It is integrated with **AWS Lambda** for backend logic and **DynamoDB** for storing user transactions.

**🚀 Features**

* **Conversational AI** using AWS Lex
* **Secure banking transactions**
* **Multi-account support** (Savings & Current)
* **Seamless chatbot integration** via **Kommunicate API**
* **Real-time responses** and **secure database storage**

**🛠️ Technologies Used**

| **Technology** | **Purpose** |
| --- | --- |
| **AWS Lex** | Chatbot framework |
| **AWS Lambda** | Backend logic for processing user requests |
| **DynamoDB** | Stores user balances and transaction history |
| **Kommunicate API** | Provides chat interface on the website/app |
| **AWS S3** | Hosts static assets & logs interactions |

**📸 Screenshots**

Below are key screenshots from the chatbot interaction:

**1️⃣ Banking Bot Welcome Screen**

👉 A computer screen shot of a computer screen

AI-generated content may be incorrect.

**2️⃣ Balance Inquiry Response**

👉 A computer screen shot of a computer

AI-generated content may be incorrect.

**3️⃣ Fund Transfer Confirmation**

👉 A computer screen shot of a computer screen

AI-generated content may be incorrect.

**4️⃣ Account Summary Screen**

👉 A computer screen shot of a chat

AI-generated content may be incorrect.

**📂 Folder Structure**

graphql

CopyEdit

BankingBot/

│── images/ # Folder for screenshots

│ ├── Welcome.jpg

│ ├── Balance.jpg

│ ├── FundTransfer.jpg

│ ├── AccountSummary.jpg

│── lambda\_function.py # AWS Lambda backend logic

│── README.md # Project Documentation

**3️⃣ Set Up AWS Lambda Function**

1️⃣ **Login to AWS Console** → Go to **AWS Lambda**.  
2️⃣ Click **Create Function** → Choose **Author from Scratch**.  
3️⃣ Set **Function Name** as BankingBotHandler.  
4️⃣ Choose **Runtime** as **Python 3.x**.  
5️⃣ In the **Permissions** section, create a new **IAM Role**.  
6️⃣ Click **Create Function**.

7️⃣ **Upload Lambda Code:**

* Open the lambda\_function.py file.
* Copy and paste the code into the AWS Lambda **code editor**.
* Click **Deploy**.

✅ **Lambda Function Created!** 🎉

**4️⃣ Configure AWS Lex Bot**

1️⃣ Open **AWS Lex Console** → Click **Create Bot**.  
2️⃣ Set Bot Name as **BankingBot** → Select **Create a new IAM Role**.  
3️⃣ Click **Next** → Set up the following **intents**:

| **Intent Name** | **Function** |
| --- | --- |
| CheckBalance | Fetch and display account balance |
| TransferFunds | Process money transfers |
| AccountSummary | Show recent transactions |

4️⃣ Under **Fulfillment**, select **AWS Lambda Function** and choose BankingBotHandler.  
5️⃣ Click **Save & Build** → **Test the Bot** in AWS Lex Console.

✅ **Lex Bot is Ready!** 🎉

**5️⃣ Deploy the Chatbot on a Website (Kommunicate API)**

1️⃣ Sign up on **Kommunicate** → Get your **API Key**.  
2️⃣ Add the following script in your **HTML** file:

html

CopyEdit

<script type="text/javascript">

(function(d, m){

var kommunicateSettings = {

"appId":"YOUR\_APP\_ID",

"popupWidget":true,

"automaticChatOpenOnNavigation":true

};

var s = document.createElement("script"); s.type = "text/javascript"; s.async = true;

s.src = "https://widget.kommunicate.io/v2/kommunicate.app";

var h = document.getElementsByTagName("head")[0]; h.appendChild(s);

window.kommunicate = m; m.\_globals = kommunicateSettings;

})(document, window.kommunicate || {});

</script>

3️⃣ Replace "YOUR\_APP\_ID" with your **Kommunicate API Key**.  
4️⃣ Save the file and open it in a browser.

✅ **Chatbot is Live on the Website!** 🎉

A computer screen shot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**📝 Additional Notes**

* AWS Lex chatbot **automatically improves** responses using **Machine Learning**.
* Ensure **IAM permissions** allow Lex to invoke **Lambda functions**.
* **DynamoDB auto-scaling** ensures faster performance.

**📬 Contact & Support**

📧 **Email:** balaji629262@gmail.com