Immigration AI Advisor

Team: Borderless.AI
Sprint 2 Presentation

Agenda

- Team Member Roles and Responsibilities
- Improvements
- Project Description
- Team Working Agreement
- Project Schedule
- Personas
- MVP
- Technologies
- Algorithms
- Diagrams
- Sprint 1 Recap
- Product Backlog
- Sprint 2 Backlog
- Metrics
- Retrospective
- Sprint 3 Planning and Commitment
- Project Demo
- Github link
- Live Demo

BORDERLESS.AI

Team Member Roles and Responsibilities

Matt Borkowski

Developer

SCRUM Master

Miles ZhengFerrari

Developer

Team Leader

Nuvorish Paul

Developer

Raj Rahul Parekh

Developer

Santhosh Charudatta

Developer

All team members collaborate as developers, contributing to coding, testing, and feature development.

BORDERLESS.AI

Improvements

Architecture Diagram

ER Diagram

Project Backlog

BORDERLESS.AI

Project Description

Project Name: Immigration AI Advisor

Team: Borderless.AI

Project Description:

Immigration AI Advisor is a AI-based application designed to assist international students with immigration-related queries by reading documents (I-20, I-94) and providing personalized advice based on immigration rules and policies.

Benefit Outcomes:

- Faster, more accurate immigration advice for international students.
- Reduced burden on international student offices with limited resources.
- Improved efficiency in managing immigration-related queries within budget constraints.

Github Link:https://github.com/htmw/2024F-borderless.ai/wiki

BORDERLESS.AI

Team Working Agreement

Team Working Agreement

To ensure the smooth and successful completion of our project, our team commits to maintaining clear communication, adhering to deadlines, and fostering a collaborative environment. We will be transparent, proactive in addressing challenges, and supportive of each other to ensure the success of the project.

Terms of Agreement:

Communication The team will communicate regularly using Zoom for scheduled meetings and Group Texts for quick discussions or emergencies. For managing tasks and project progress, Jira will be our primary tool.

- **Meetings**: Mandatory Zoom meetings will be held every Monday, Wednesday, and Friday. All team members must attend unless they have communicated their absence ahead of time.
- **Response Time**: Members are expected to respond to messages within 24 hours to ensure timely communication and issue resolution.
- **Project Updates**: Critical project updates will be shared promptly to keep everyone informed, and collaborative efforts should begin early to avoid last-minute workloads.

Work Division and Participation Each team member will take responsibility for their assigned tasks and contribute to the overall project success. Work will be divided fairly according to each member's skill set, and any blockers or difficulties must be communicated as early as possible.

- **Deadlines**: Tasks are expected to be completed by the team's scheduled deadlines. Last-minute handovers or submissions will not be tolerated.
- **Support**: Members should ask for help when needed, well in advance of deadlines. The team will support each other to ensure tasks are completed efficiently.
- **Task Tracking**: All tasks will be tracked and managed through Jira, and members are expected to regularly update their progress to maintain visibility across the team.

Code and Quality Assurance Code must be submitted for peer review at least 48 hours before any major deadline to ensure adequate time for feedback and revisions. Proper testing will be conducted on all code to maintain quality and prevent errors in the final product.

Meetings The team will meet on Zoom three times a week (Monday, Wednesday, and Friday) for project discussions, progress updates, and problem-solving. All members are required to attend unless they've communicated their absence beforehand. If a member misses a meeting, they are responsible for catching up via meeting notes and watching recordings.

- **Retrospectives**: After each sprint, the team will hold a retrospective to reflect on what went well, identify areas for improvement, and set actionable goals for the next sprint.

Respect and Team Culture We will foster a respectful and supportive environment where every team member can share their ideas, give and receive feedback, and collaborate openly. Each member will take ownership of their tasks and contribute to the continuous improvement of our processes and teamwork.

Signed by: *BorderlessAI*

BORDERLESS.AI

Project Schedule

Sprint 0: Understanding the Project (Pre-Development Phase)

Goal: Gather requirements, define project scope, and identify key user needs.

Activities:

Understand the scope of immigration document handling (I-20, I-94).

Research relevant immigration rules and policies.

Define key features for the Immigration AI Advisor (PDF upload, document parsing, personalized advice).

Set up the development environment and select tech stack (Python, Flask, AI framework).

Initial team roles assignment.

BORDERLESS.AI

Project Schedule

Sprint 1: Basic PDF Upload & Parsing

Goal: Build and demonstrate the basic functionality of the Immigration AI Advisor.

Tasks:

Implement PDF upload functionality via the web UI.

Develop code for reading and extracting basic information from I-20 PDFs (e.g., Name, DoB).

Establish Flask as the backend for handling requests.

Test and verify that the I-20 upload and parsing process is functional.

Demo the working upload and basic data extraction.

BORDERLESS.AI

Project Schedule

Sprint 2: AI Component and Chat Box Front End

Goal: Introduce basic AI-driven advice.

Tasks:

Implement basic AI functionality to begin providing personalized advice

Ensure the system is flexible enough to handle more complex parsing as needed.

Front End UI and Chat Box functionality

BORDERLESS.AI

Project Schedule

Sprint 3: AI Component Enhancement

Goal: Improve the AI's capability to provide more complex immigration advice.

Tasks:

Refine the AI logic to handle more nuanced queries based on immigration policies.

Add more advanced decision-making capabilities to the AI component.

Enhance the AI model's ability to analyze and cross-reference multiple document types.

Test the AI's performance and ensure it scales well with increased complexity.

Prepare the Immigration AI Advisor for final testing and potential deployment.

BORDERLESS.AI

Personas

Name: Sarah Johnson

Age: 35

Role: International Student Adviser

Background:

Sarah has been an adviser for 8 years, handling hundreds of students each semester. She's responsible for guiding them through immigration processes, ensuring compliance with visa regulations, and managing I-20 and I-94 forms.

Challenges:

Overwhelmed by repetitive tasks (e.g., checking I-20, I-94 documents). Struggles to keep up with changing immigration rules. Limited time for personalized student interactions due to workload.

Goals:

Automate routine tasks to reduce workload. Improve accuracy and efficiency in advising. Free up time for more complex student cases.

How Immigration AI Advisor Helps:

Automates document review (I-20, I-94).

Provides preliminary, personalized immigration advice.

Reduces manual workload, enabling Sarah to focus on complex cases.

BORDERLESS.AI

Personas

Name: Aakken Lee

Age: 22

Role: International Student, 3rd-year undergraduate

Background:

Aakken is studying Computer Science and is on an F-1 visa. He regularly deals with immigration paperwork like I-20s and I-94s, and needs to stay compliant with visa regulations to maintain his legal status in the U.S.

Challenges:

Needs quick, accurate advice on maintaining visa status. Often faces delays due to overburdened advisers. Uncertainty about document requirements or policy changes causes stress.

Goals:

Access immediate immigration information and advice. Avoid potential issues with immigration status.

How Immigration AI Advisor Helps:

Provides instant, accurate responses to common immigration queries.

Offers clear explanations of I-20 and I-94 documents.

Reduces reliance on advice availability for basic information.

BORDERLESS.AI

Personas

Name: Dr. Maria Rodriguez

Age: 50

Role: VP of Office of Global Services

Background:

Dr. Rodriguez's role focuses on ensuring compliance with immigration regulations while enhancing student support and satisfaction. She's responsible for strategic planning, resource allocation, and policy implementation.

Challenges:

Ensuring the institution stays compliant with evolving immigration laws. Balancing limited resources with the growing needs of international students.

Goals:

Improve efficiency to better serve students. Use data and automation to ensure compliance with immigration rules while respect the limited budget.

How Immigration AI Advisor Helps:

Streamlines immigration document processing, freeing up adviser resources.

Enhances support for international students, improving satisfaction and retention.

BORDERLESS.AI

MVP

Description:

The MVP of the Immigration AI Advisor focuses on delivering essential functionality to assist international students with their immigration-related needs. The MVP allows users to upload their I-20 document, and the system extracts key information such as their Name and Date of Birth. This basic document parsing provides immediate value by automating a critical task, helping students quickly access and verify important information from their immigration paperwork.

Key Features of the MVP:

PDF Upload: Users can upload an I-20 form through a web interface.

Data Extraction: The system automatically extracts and displays basic information (e.g., Name, Date of Birth).

Usability: Users can interact with the system, receive valuable information instantly, and ensure their immigration documents are processed accurately.

Value to Users:

The MVP solves a practical problem for students by saving time and reducing the complexity of manually reading their immigration documents. It provides immediate utility while laying the groundwork for future enhancements, such as more detailed document parsing and AI-based immigration advice.

BORDERLESS.AI

Technologies

Languages & Frameworks:

Python, Flask, FastAPI, Pytest, ReactJS, HTML, CSS

Artificial Intelligence:

TensorFlow, PyTorch, Chat GPT4o mini, BART, Claude

Any additional technology we may discover and add as we move on to each sprints.

BORDERLESS.AI

Algorithms

Regular Expressions/Pattern Matching

To recognize structure of, and extract information from, I-20 formatted forms

AI Large Language Models

LLM's (Large Language Models), minimally trained language models, to provide the basis of a custom-trained, immigration-oriented "knowledge base"

TensorFlow/ PyTorch:

Might be used for statistical/quantitative analysis and transformation, in order to augment LLMs' capabilities

BORDERLESS.AI

Diagrams

Architecture Diagrams

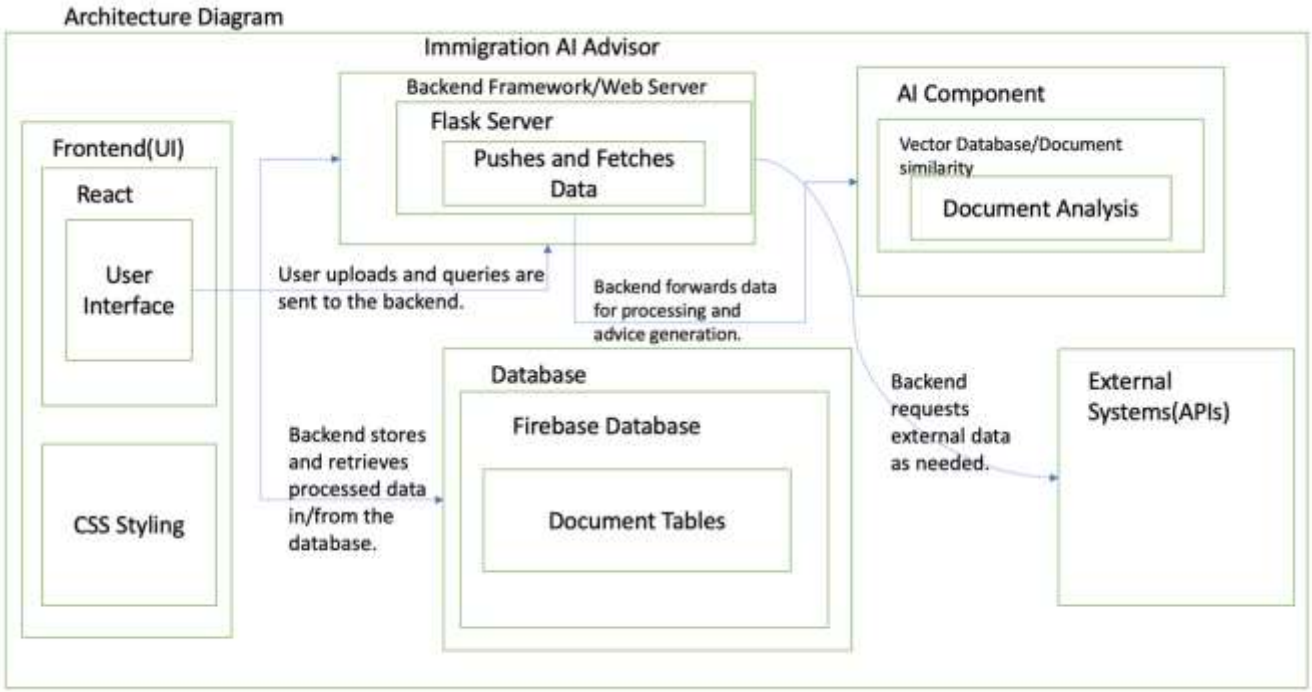
Context Diagram

ER Diagram

Sequence Diagram

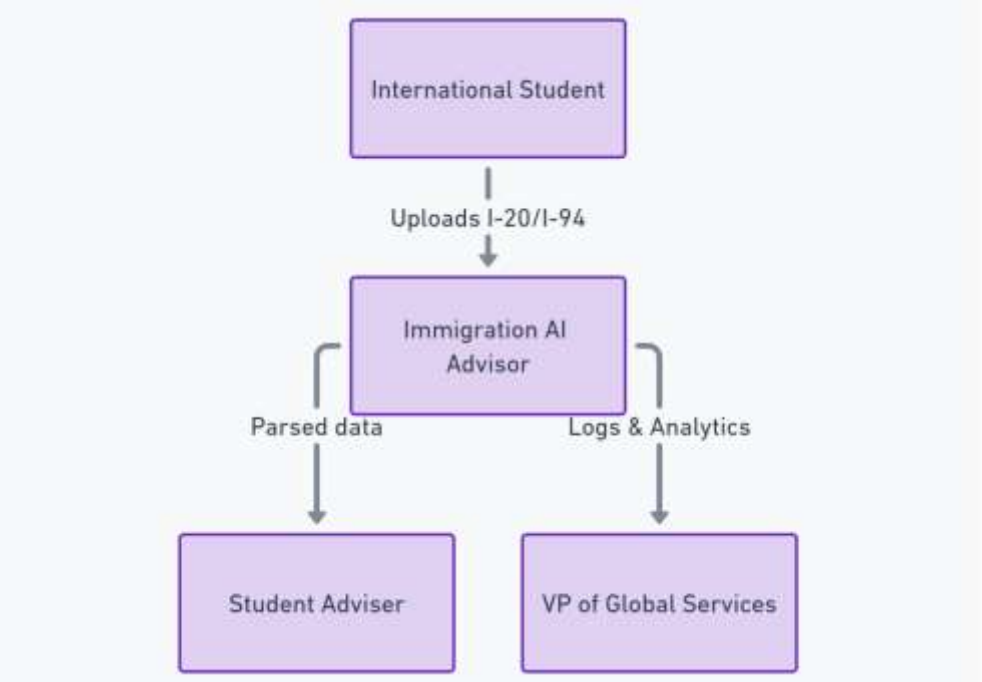
State Diagrams

Architecture Diagram



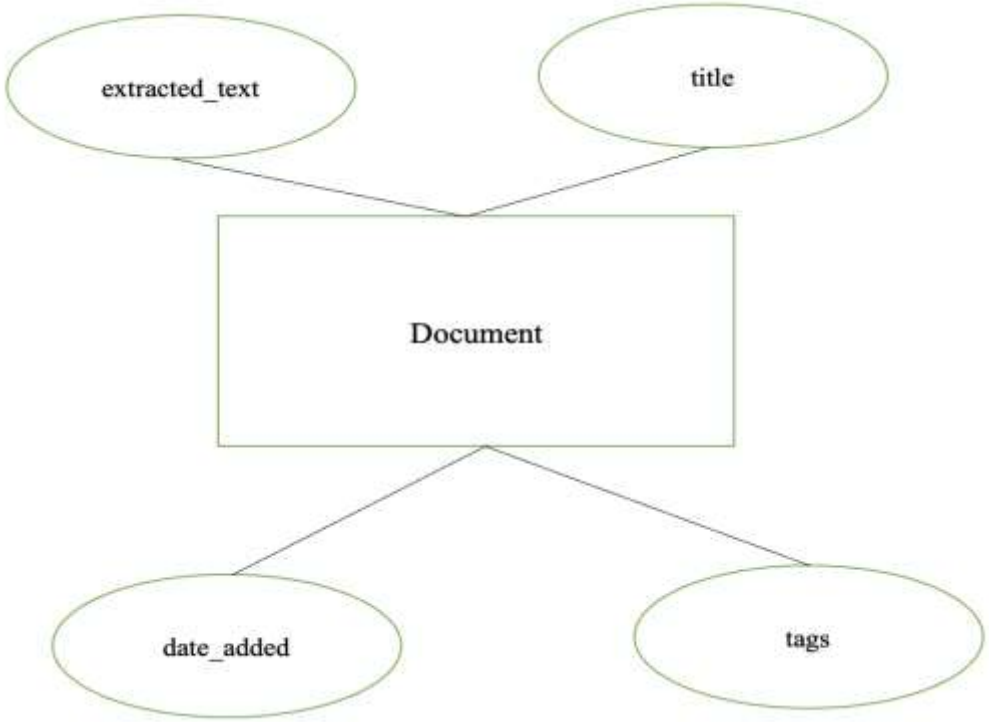
BORDERLESS.AI

Context Diagram

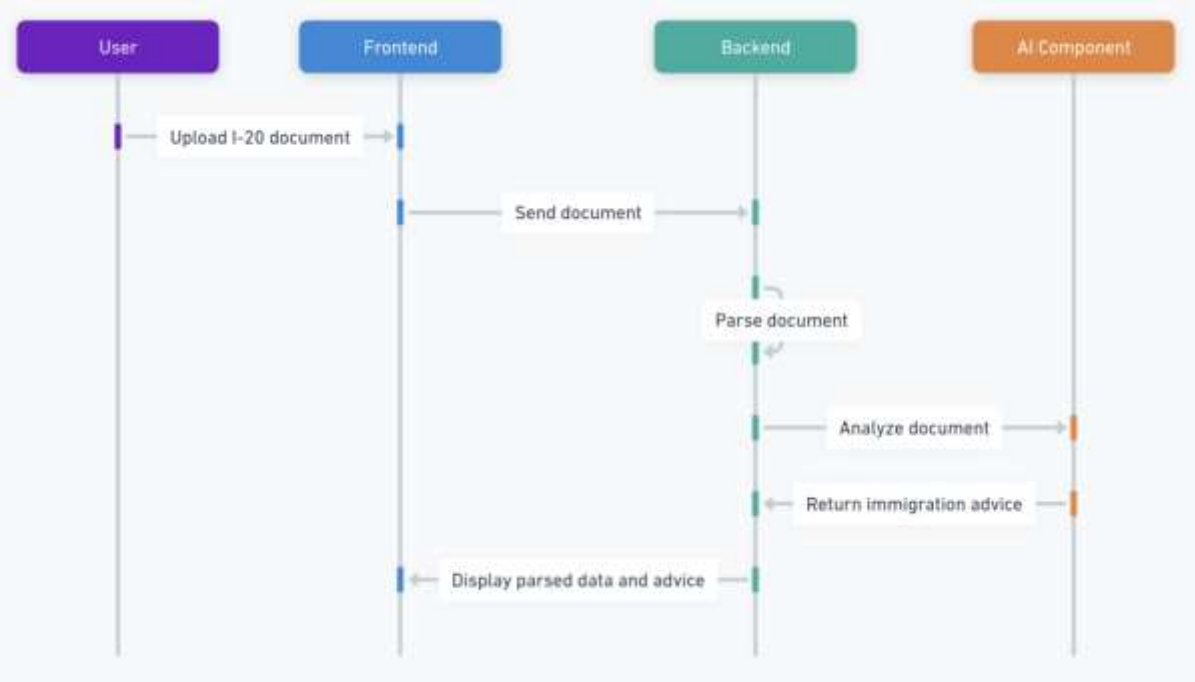


BORDERLESS.AI

ER Diagram (Entity-Relationship Diagram)

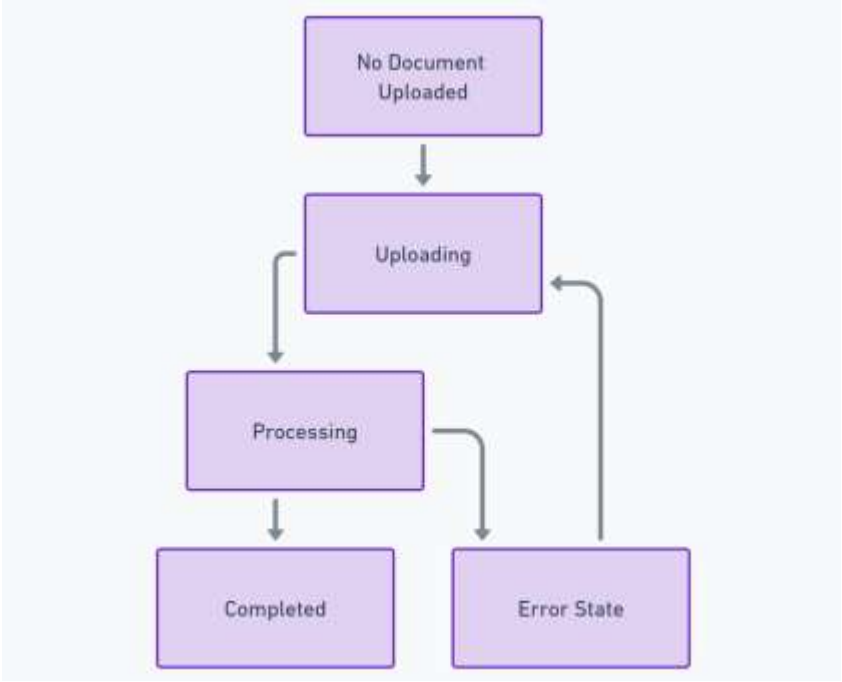


Sequence Diagram



BORDERLESS.AI

State Diagram



Sprint 1 Recap

Sprint Goals:

Objective: Build and demonstrate the basic functionality of the Immigration AI Advisor.

- Enable users to upload I-20 PDFs via the web UI.
- Extract key information (e.g., Name, Date of Birth) from I-20 PDFs.
- Set up Flask as the backend to handle requests and processing.

Key Accomplishments:

- PDF Upload Functionality: Successfully developed and tested a user-friendly interface for document uploads.
- Basic Information Parsing: Implemented code to extract Name and Date of Birth from I-20 PDFs accurately.
- Backend Setup: Established Flask as the backend and created endpoints for handling uploads.
- Testing and Validation: Ensured stability for invalid or corrupted file uploads.

BORDERLESS.AI

Project Backlog
Stories or Tasks for the Whole Project

# Sprint	User Story/Tasks ID	User Story/Tasks	Acceptance Criteria	Status
0	0.1	As a developer, I want to research and understand the immigration document requirements (I-20, I-94), so that I can build the necessary parsing functionality.	Clear understanding of I-20 and I-94 document structures. Document key fields that need to be parsed (e.g., Name, DoB, Visa Status).	Completed
0	0.2	As a developer, I want to set up the development environment and tech stack, so that the team can start building the application.	Development environment is set up (Python, Flask, AI frameworks). Shared repository for the team to collaborate.	Completed
1	1.1	As an International Student, I want to upload my I-20 document, so that I can view my basic information (Name, Date of Birth).	User can upload an I-20 PDF from the web interface. The system extracts and displays Name and Date of Birth.	Completed
1	1.2	As a developer, I want to implement the PDF upload functionality on the frontend and backend, so that students can upload their documents.	Users can upload PDFs from the frontend. The backend processes uploaded documents.	Completed
1	1.3	As a developer, I want to implement basic parsing functionality for I-20 documents, so that key fields (Name, DoB) can be extracted.	System parses I-20 documents and extracts Name and Date of Birth. Extracted information is displayed in the frontend user interface.	Completed
1	1.4	As a developer, I want to ensure the application has error handling during the document upload process, so that users receive appropriate feedback for failed uploads.	Clear error messages for upload issues (e.g., wrong file type, corrupted file). The system logs errors for troubleshooting.	Completed
1	1.5	As a developer, I want to set up the basic structure of the Flask backend, so that I can manage file uploads and communication between the frontend and backend.	Flask is set up with routes for handling file uploads. The backend communicates with the frontend for document uploads.	Completed
2	2.1	As a developer, I want to collect immigration-related documents, so that the team can use them to provide personalized advice.	USCIS documents are collected. Common immigration-related Q&A are compiled.	Completed
2	2.2	As a developer, I want to implement a tagging system, so that we have a structure to sort all the AI training materials which can train AI to provide personalized responses.	Materials are systematically identified and tagged. Document data is centralized and mapped to relevant questions accurately.	Completed
2	2.3	As a developer, I want to develop an AI model (using BERT or an RNN), so that the system can handle user queries and deliver accurate responses.	AI model is trained on a dataset of immigration-related queries and document data. Model achieves an acceptable accuracy threshold in query-response testing.	Completed
2	2.4	As a user, I want to input immigration-related questions or queries into the system, so that I can receive tailored advice.	Users can input and submit queries through the system. Queries are processed and mapped to relevant responses or document data.	Completed

# Sprint	User Story/Tasks ID	User Story/Tasks	Acceptance Criteria	Status
2	2.5	As a user, I want a chatbox on the frontend where I can interact with the Immigration AI Advisor, so that I can easily ask questions and receive responses.	Chatbox is functional and user-friendly. Integrated with backend systems to process queries and display responses.	Completed
2	2.6	As a developer, I want to fine-tune the AI model, so that it delivers more accurate and high-quality responses to user queries.	Fine-tuned model achieves higher performance metrics compared to the baseline. Improvements are documented and validated against test cases.	Not Completed
3	3.1	Generate text responses based on query and knowledge base	The system generates accurate, coherent, and context-aware text responses. Responses are relevant to the query and align with the knowledge base. Handles queries outside the knowledge base gracefully by providing appropriate feedback or a fallback response.	Not Completed
3	3.2	Improve semantic search	Semantic search returns the most contextually relevant results for user queries. Latency for search results is minimized and consistent. Improved precision and recall metrics for search functionality are documented.	Not Completed
3	3.3	Clean up database (look at "extracted_text" field/column in Firebase, clean up)	The extracted_text field/column is normalized and free from redundant or irrelevant data. All unnecessary or malformed entries in Firebase are removed. The cleaned database supports efficient queries and ensures consistency across records.	Not Completed
3	3.4	Improve tokenization and segmentation methods (use something more state-of-the-art)	Tokenization and segmentation methods reduce errors in text parsing and analysis. Benchmarked against baseline methods with measurable improvements in performance. Integrated seamlessly into the existing AI pipeline without causing regressions.	Not Completed
3	3.5	Support users (login/auth)	Users can register and log in securely. Authentication is implemented with industry-standard practices (e.g., OAuth, JWT). User roles (e.g., admin, student) are defined and enforced within the application.	Not Completed

# Sprint	User Story/Tasks ID	User Story/Tasks	Acceptance Criteria	Status
			documented.	
3	3.3	Clean up database (look at "extracted_text" field/column in Firebase, clean up)	The extracted_text field/column is normalized and free from redundant or irrelevant data. All unnecessary or malformed entries in Firebase are removed. The cleaned database supports efficient queries and ensures consistency across records.	Not Completed
3	3.4	Improve tokenization and segmentation methods (use something more state-of-the-art)	Tokenization and segmentation methods reduce errors in text parsing and analysis. Benchmarked against baseline methods with measurable improvements in performance. Integrated seamlessly into the existing AI pipeline without causing regressions.	Not Completed
3	3.5	Support users (login/auth)	Users can register and log in securely. Authentication is implemented with industry-standard practices (e.g., OAuth, JWT). User roles (e.g., admin, student) are defined and enforced within the application.	Not Completed
3	3.6	Store user documents/contents	Uploaded user documents are securely stored in a database or cloud storage. Documents are tagged and indexed for efficient retrieval. Document storage complies with relevant privacy and data protection standards (e.g., GDPR, CCPA).	Not Completed
3	3.7	Response to queries based on existing functionality, but incorporating user-uploaded documents	The system uses user-uploaded documents to generate tailored responses. Queries are processed to include relevant context from the uploaded documents. Provides accurate and personalized advice based on document data.	Not Completed
3	3.8	Deploy a free, private demo version to web	A demo version of the application is deployed to a web server. Users can interact with core functionalities (e.g., upload documents, ask queries). Demo is secured to prevent unauthorized access and data leaks.	Not Completed

Sprint 2 Backlog
Stories or Tasks committed for this Sprint

Task 2.1: As a developer, I want to collect immigration-related documents, so that the team can use them to provide personalized advice.
Acceptance Criteria:

- USCIS documents are collected.
- Common immigration-related Q&A are compiled.
  Story Points: 4

Task 2.2: As a developer, I want to implement a tagging system, so that we have a structure to sort all the AI training materials which can train AI to provide personalized responses.
Acceptance Criteria:

- Materials are systematically identified and tagged.
- Document data is centralized and mapped to relevant questions accurately.
  Story Points: 5

BORDERLESS.AI

Sprint 2 Backlog
Stories or Tasks committed for this Sprint

Task 2.3: As a developer, I want to develop an AI model (using BERT or an RNN), so that the system can handle user queries and deliver accurate responses.
Acceptance Criteria:

- AI model is trained on a dataset of immigration-related queries and document data.
- Model achieves an acceptable accuracy threshold in query-response testing.
  Story Points: 15

User Story 2.4: As a user, I want to input immigration-related questions or queries into the system, so that I can receive tailored advice.
Acceptance Criteria:

- Users can input and submit queries through the system.
- Queries are processed and mapped to relevant responses or document data.
  Story Points: 5

BORDERLESS.AI

Sprint 2 Backlog
Stories or Tasks committed for this Sprint

User Story 2.5: As a user, I want a chatbox on the frontend where I can interact with the Immigration AI Advisor, so that I can easily ask questions and receive responses.
Acceptance Criteria:

- Chatbox is functional and user-friendly.
- Integrated with backend systems to process queries and display responses.
  Story Points: 7

Task 2.6: As a developer, I want to fine-tune the AI model, so that it delivers more accurate and high-quality responses to user queries.
Acceptance Criteria:

- Fine-tuned model achieves higher performance metrics compared to the baseline.
- Improvements are documented and validated against test cases.
  Story Points: 10

BORDERLESS.AI

Sprint 2 Backlog
Test Cases

Task 2.1: As a developer, I want to collect immigration-related documents, so that the team can use them to provide personalized advice.

- Check if USCIS documents are collected.
- Check if Common immigration-related Q&A are compiled.
  Story Points: 4, **TID: 2001**

Task 2.2: As a developer, I want to implement a tagging system, so that we have a structure to sort all the AI training materials which can train AI to provide personalized responses.

- Verify the Materials are systematically identified and tagged.
- Verify if the Document data is centralized and mapped to relevant questions accurately.
  Story Points: 5, **TID: 2002**

BORDERLESS.AI

Sprint 2 Backlog
Test Cases

Task 2.3: As a developer, I want to develop an AI model (using BERT or an RNN in PyTorch or TensorFlow), so that the system can handle user queries and deliver accurate responses.

- Ensure that the AI model is trained on a dataset of immigration-related queries and document data.
- Verify if the Model achieves an acceptable accuracy threshold in query-response testing.
  Story Points: 15, **TID: 2003**

User Story 2.4: As a user, I want to input immigration-related questions or queries into the system, so that I can receive tailored advice.

- Test if the Users can input and submit queries through the system.
- Check if the Queries are processed and mapped to relevant responses or document data.
  Story Points: 5, **TID: 2004**

BORDERLESS.AI

Sprint 2 Backlog
Test Cases

User Story 2.5: As a user, I want a chatbox on the frontend where I can interact with the Immigration AI Advisor, so that I can easily ask questions and receive responses.

- Test the Chatbox to verify that it is functional and user-friendly.
- Check if it is integrated with backend systems to process queries and display responses.
  Story Points: 7, **TID: 2005**

Task 2.6: As a developer, I want to fine-tune the AI model, so that it delivers more accurate and high-quality responses to user queries.

- Does the fine-tuned model achieve higher performance metrics compared to the baseline?
- Are the improvements documented and validated against predefined test cases?
  Story Points: 10,**TID: 2006**

BORDERLESS.AI

Sprint 2 Backlog
Stories completed

Task 2.1: As a developer, I want to collect immigration-related documents, so that the team can use them to provide personalized advice.
Acceptance Criteria:

- USCIS documents are collected.
- Common immigration-related Q&A are compiled.
  Story Points: 4, **Status : Completed**

Task 2.2: As a developer, I want to implement a tagging system, so that we have a structure to sort all the AI training materials which can train AI to provide personalized responses.
Acceptance Criteria:

- Materials are systematically identified and tagged.
- Document data is centralized and mapped to relevant questions accurately.
  Story Points: 5, **Status : Completed**

BORDERLESS.AI

Sprint 2 Backlog
Stories completed

Task 2.3: As a developer, I want to develop an AI model (using BERT or an RNN), so that the system can handle user queries and deliver accurate responses.
Acceptance Criteria:

- AI model is trained on a dataset of immigration-related queries and document data.
- Model achieves an acceptable accuracy threshold in query-response testing.
  Story Points: 15, **Status : Completed**

User Story 2.4: As a user, I want to input immigration-related questions or queries into the system, so that I can receive tailored advice.
Acceptance Criteria:

- Users can input and submit queries through the system.
- Queries are processed and mapped to relevant responses or document data.
  Story Points: 5, **Status : Completed**

BORDERLESS.AI

Sprint 2 Backlog
Stories completed

User Story 2.5: As a user, I want a chatbox on the frontend where I can interact with the Immigration AI Advisor, so that I can easily ask questions and receive responses.
Acceptance Criteria:

- Chatbox is functional and user-friendly.
- Integrated with backend systems to process queries and display responses.
  Story Points: 7, **Status : Completed**

BORDERLESS.AI

Sprint 2 Backlog Stories Not Completed

Task 2.6: As a developer, I want to fine-tune the AI model, so that it delivers more accurate and high-quality responses to user queries.

Acceptance Criteria:

- Fine-tuned model achieves higher performance metrics compared to the baseline.
- Improvements are documented and validated against test cases.

Story Points: 10, **Status : Not-Completed**

Total Story Points: 46 points

Total Completed Stories: 36 points

Total Stories not completed: 10 points

BORDERLESS.AI

Metrics

Team Velocity - Sprint 2

Details:

The team successfully completed 5 user stories, focusing on core functionalities such as collecting immigration-related documents, implementing a tagging system, developing an AI model, enabling user queries, and integrating a functional chatbox. These achievements significantly advanced the system's ability to provide personalized immigration advice. However, the task to fine-tune the AI model for improved accuracy remains incomplete.

The unfinished task involves fine-tuning the AI model to achieve higher performance metrics compared to the baseline and documenting improvements against test cases. This will be carried forward to the next sprint.

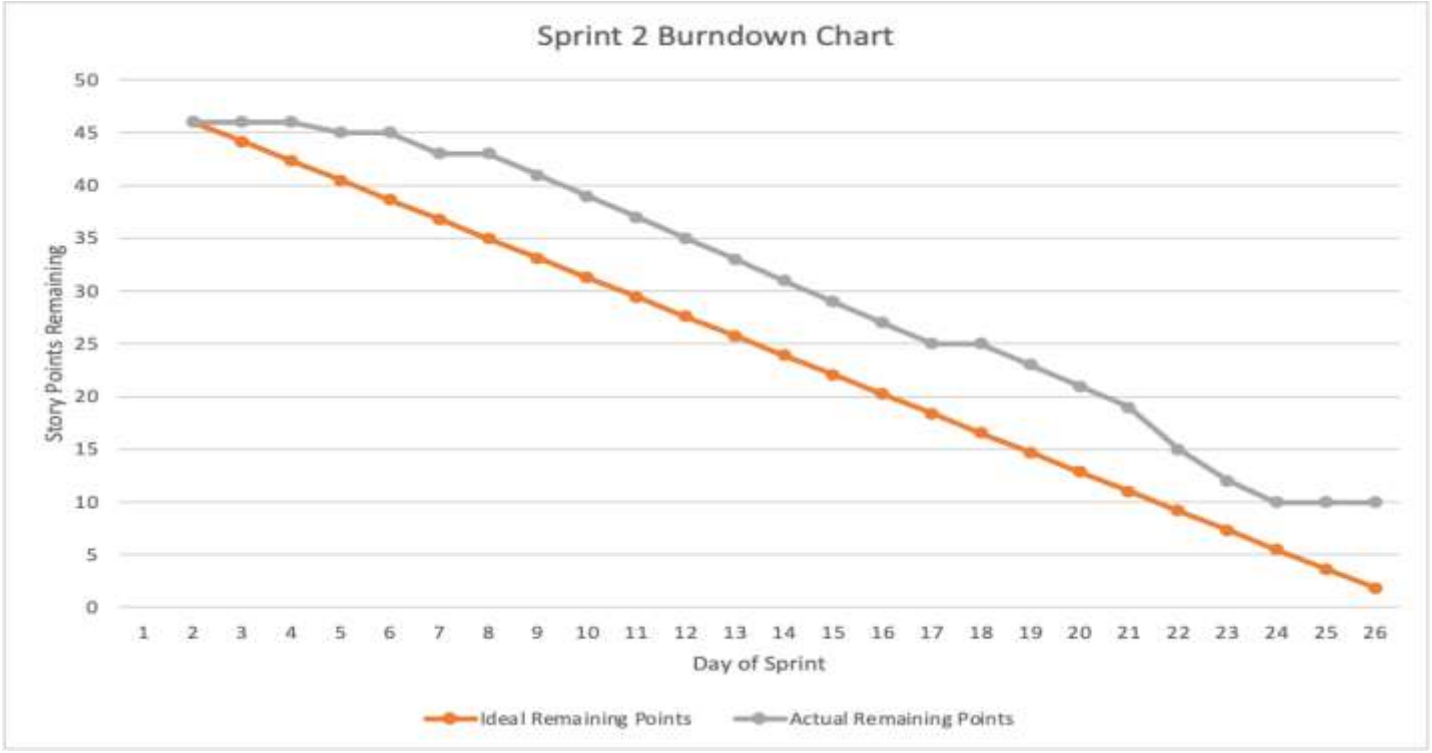
Story points are based on the complexity and effort required for each story, not the number of issues.

Total Story Points: 46

Completed Story Points: 36

Not Completed Story Points: 10

Metrics



Metrics

Completed/Committed Ratio:

36/46=78.28%

BORDERLESS.AI

Retrospective

What Went Well?

Improved the MVP.

Improved the front-end design.

Implemented basic AI functionality.

Successfully collected required documents.

Followed Agile programming principles.

Completed most user stories.

What Can Be Improved?

Improve communication via WhatsApp.

Adhere to deadlines more strictly.

Update the team regularly on individual tasks.

Provide specific details on issues and challenges.

Increase transparency on work progress.

Avoid last-minute communication; encourage collaboration.

Actionable Items:

Respond promptly to WhatsApp messages ASAP when seen.

Collaborate more actively in the next sprint by forming subtext groups.

Complete all components 72 hours before deadlines.

Ensure three meetings per week, with one in-person meeting.

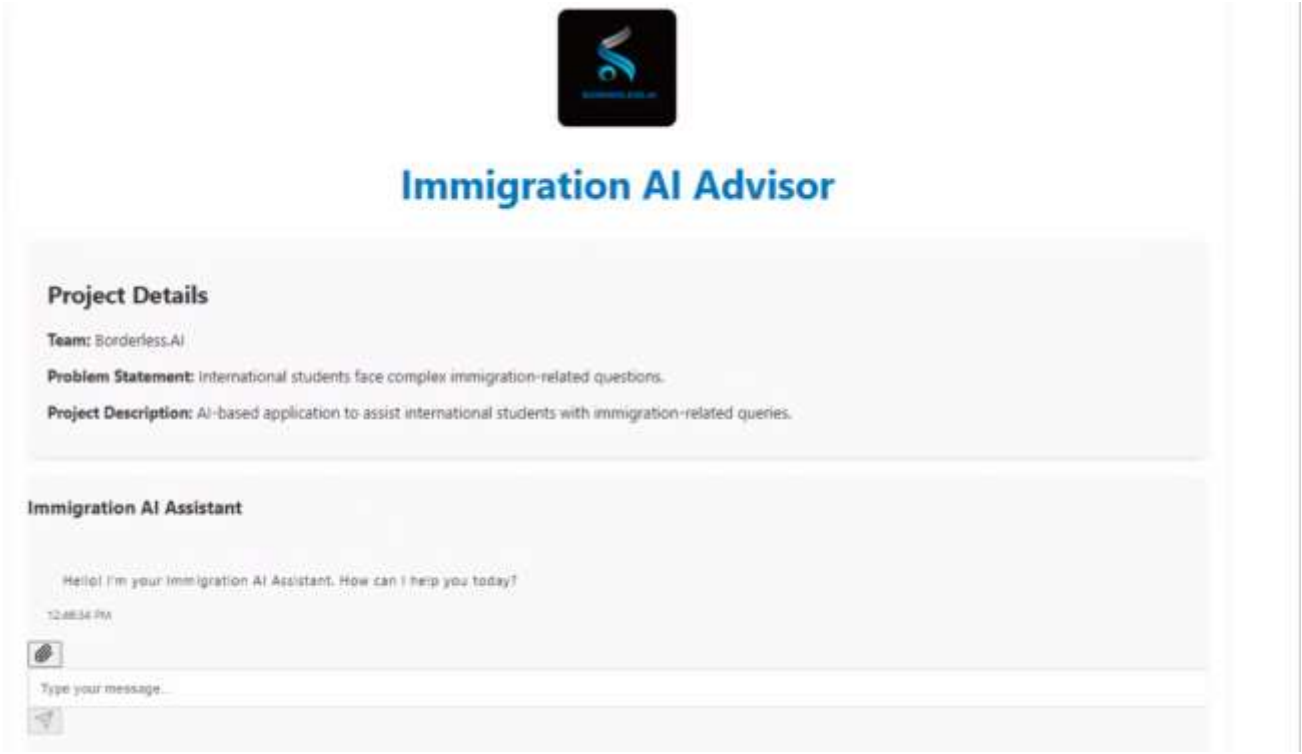
Attempt to conduct daily standups for progress tracking.

Sprint 3 Planning and Commitment

# Sprint	User Story/Tasks ID	User Story/Tasks	Acceptance Criteria	Status
3	3.1	Generate text responses based on query and knowledge base	The system generates accurate, coherent, and context-aware text responses. Responses are relevant to the query and align with the knowledge base. Handles queries outside the knowledge base gracefully by providing appropriate feedback or a fallback response.	Not Completed
3	3.2	Improve semantic search	Semantic search returns the most contextually relevant results for user queries. Latency for search results is minimized and consistent. Improved precision and recall metrics for search functionality are documented.	Not Completed
3	3.3	Clean up database (look at "extracted_text" field/column in Firebase, clean up)	The extracted_text field/column is normalized and free from redundant or irrelevant data. All unnecessary or malformed entries in Firebase are removed. The cleaned database supports efficient queries and ensures consistency across records.	Not Completed
3	3.4	Improve tokenization and segmentation methods (use something more state-of-the-art)	Tokenization and segmentation methods reduce errors in text parsing and analysis. Benchmarked against baseline methods with measurable improvements in performance. Integrated seamlessly into the existing AI pipeline without causing regressions.	Not Completed
3	3.5	Support users (login/auth)	Users can register and log in securely. Authentication is implemented with industry-standard practices (e.g., OAuth, JWT). User roles (e.g., admin, student) are defined and enforced within the application.	Not Completed
3	3.6	Store user documents/contents	Uploaded user documents are securely stored in a database or cloud storage. Documents are tagged and indexed for efficient retrieval. Document storage complies with relevant privacy and data protection standards (e.g., GDPR, CCPA).	Not Completed
3	3.7	Response to queries based on existing functionality, but incorporating user-uploaded documents	The system uses user-uploaded documents to generate tailored responses. Queries are processed to include relevant context from the uploaded documents. Provides accurate and personalized advice based on document data.	Not Completed
3	3.8	Deploy a free, private demo version to web	A demo version of the application is deployed to a web server. Users can interact with core functionalities (e.g., upload documents, ask queries). Demo is secured to prevent unauthorized access and data leaks.	Not Completed

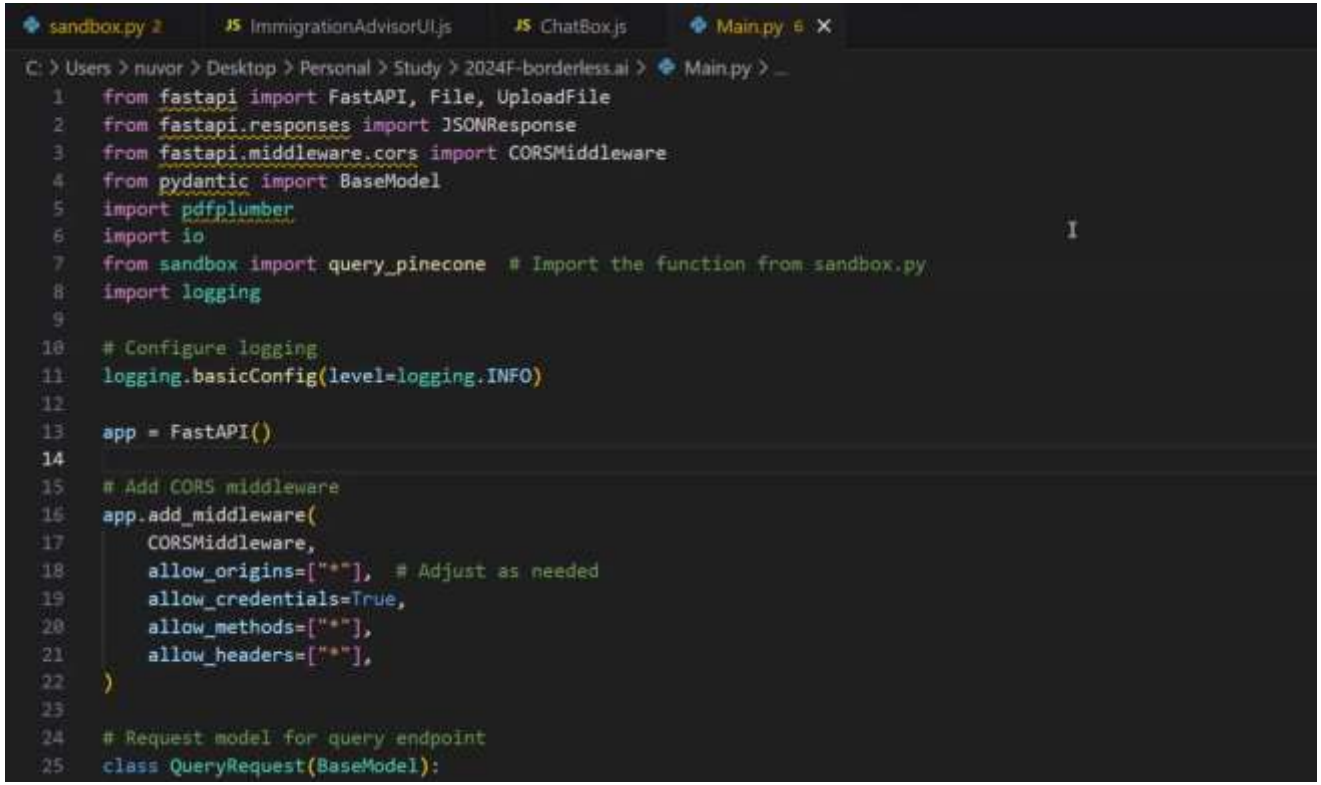
BORDERLESS.AI

Project Demo



BORDERLESS.AI

Project Demo



BORDERLESS.AI

Github Link

https://github.com/htmw/2024F-borderless.ai/wiki

BORDERLESS.AI

Live Application Demo

BORDERLESS.AI

THANK YOU!

BORDERLESS.AI