# Immigration AI Advisor

Team: Borderless.AI
Sprint 3 Presentation

# Agenda

- Team Member Roles and Responsibilities
- Improvements
- Project Description
- Team Working Agreement
- Personas
- MVP
- Technologies
- Algorithms
- Diagrams
- Sprint 2 Recap
- Product Backlog (Sprint1,Sprint 2)
- Sprint Summary (Sprint 1, Sprint 2)
- Sprint 3 Backlog
- Metrics
- Retrospective
- Project Demo
- API
- Github link
- Live Demo

# 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.*

# Improvements

- User Stories/Tasks
  - Type of user/user role
  - Story points- fibonacci numbers for Sprint 3
  - Table for backlog stories
- Team velocity - history, average, etc
- User story 2.6 in sprint 3
- Api slides
- Explaining Burndown chart in more detai during presentation

# 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.

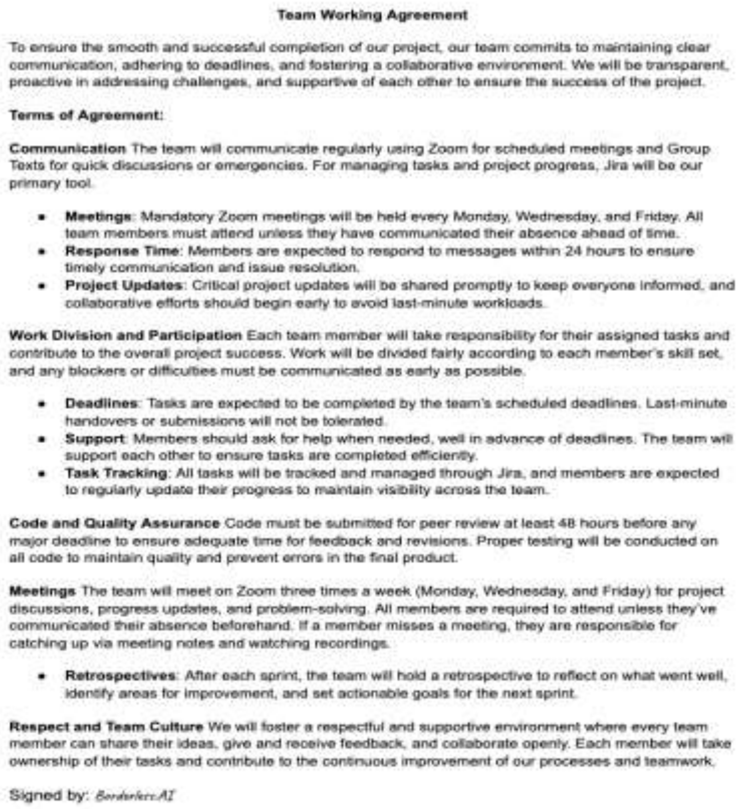
**Benefit Outcomes:**

- 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*

# 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.

# MVP

**Description:**

Building upon this foundation, the current sprint introduces a new feature: the addition of abstract summaries to the retrieved documents. This enhancement provides users with concise and meaningful overviews of their documents, further simplifying the process of understanding and managing immigration-related paperwork. By combining automated key information extraction with high-level summaries, the MVP delivers greater value, reducing time spent on manual reviews and improving overall accessibility

**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.

# 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.

# 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

# Diagrams

Architecture Diagrams

Context Diagram

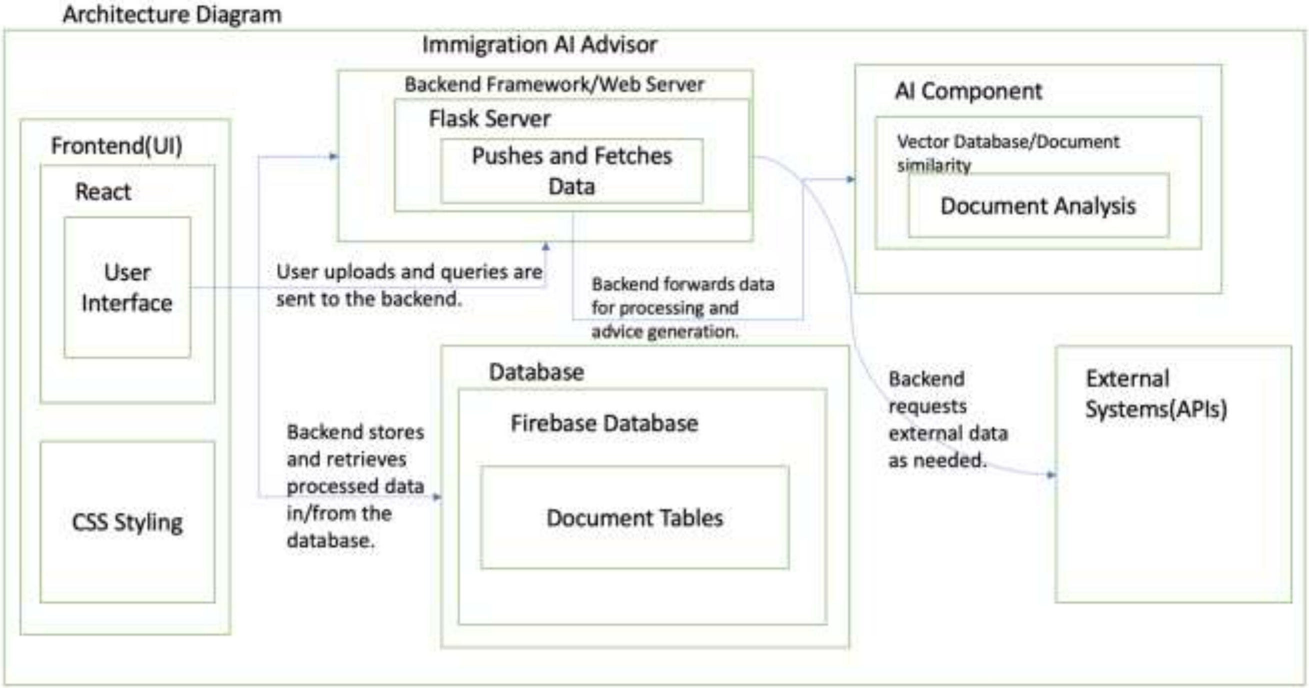
ER Diagram

Sequence Diagram

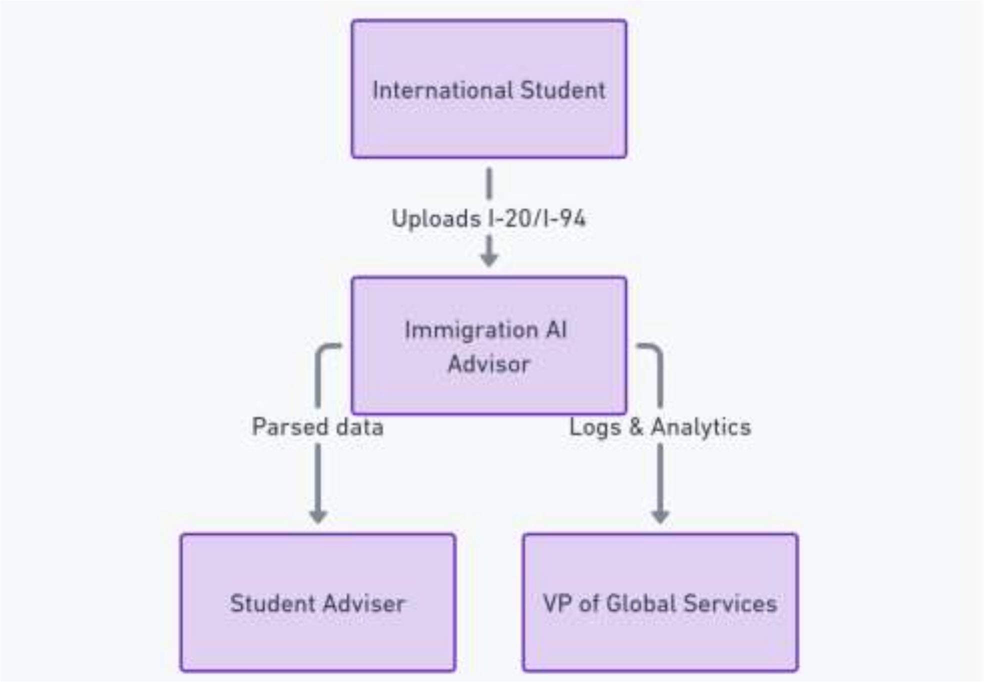
State Diagrams

BORDERLESS.AI

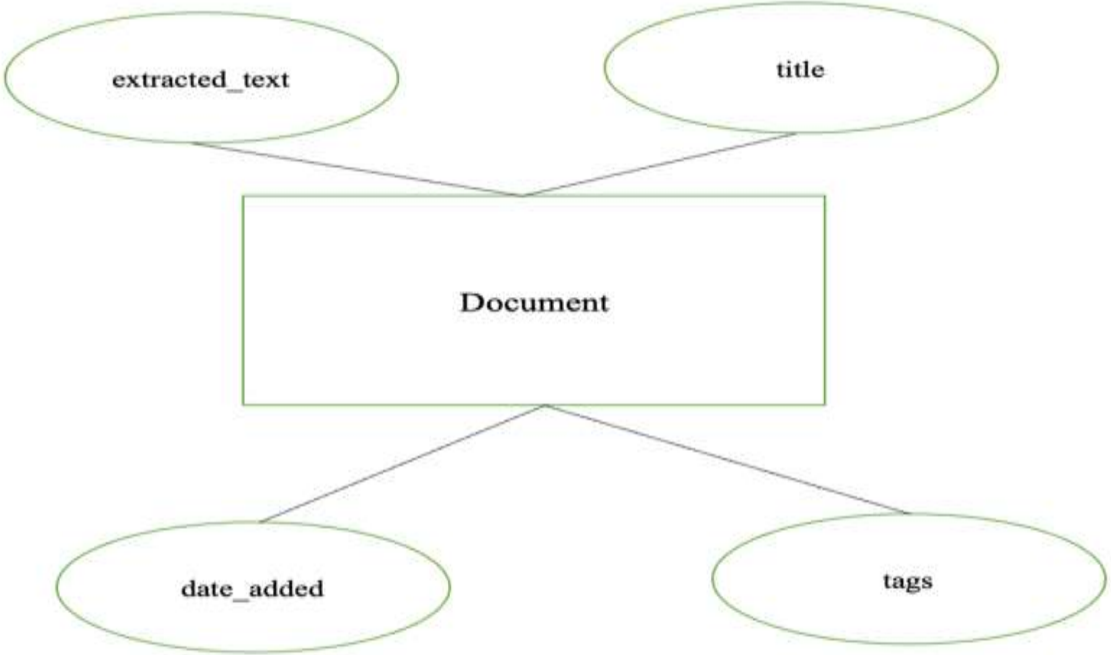
# Architecture Diagram


Architecture Diagram
Immigration AI Advisor
Frontend(UI)
React
User Interface
CSS Styling
Backend Framework/Web Server
Flask Server
Pushes and Fetches Data
User uploads and queries are sent to the backend.
Backend forwards data for processing and advice generation.
Backend stores and retrieves processed data in/from the database.
Database
Firebase Database
Document Tables
AI Component
Vector Database/Document similarity
Document Analysis
Backend requests external data as needed.
External Systems(APIs)

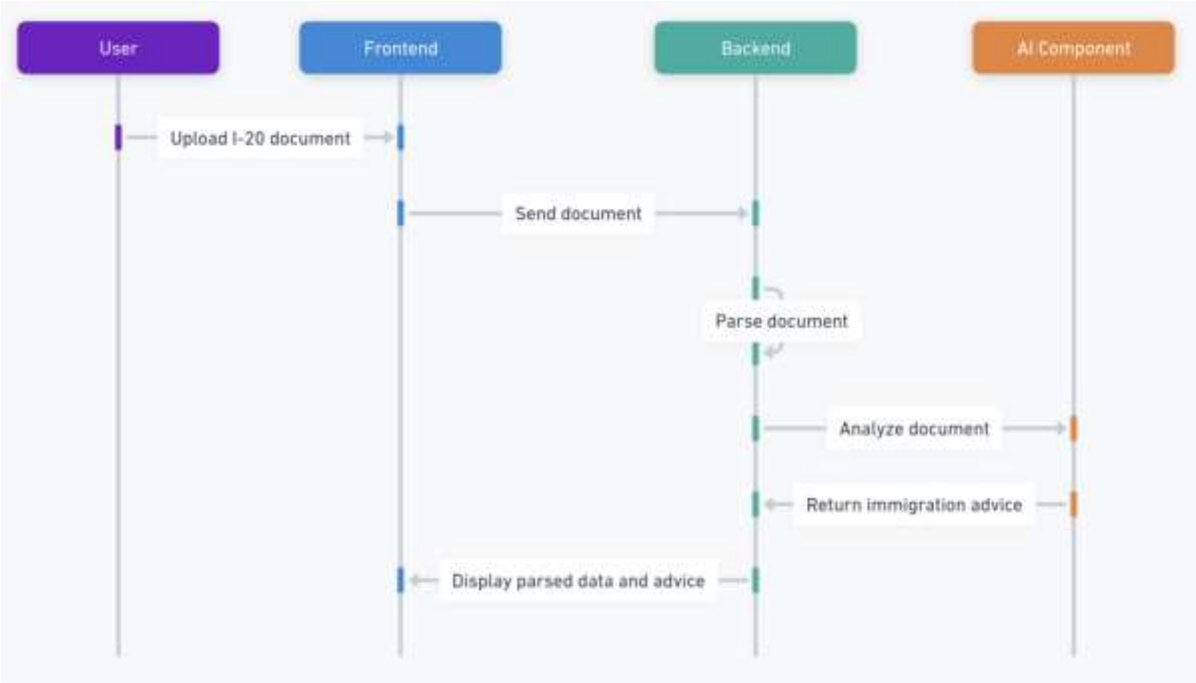
# Context Diagram



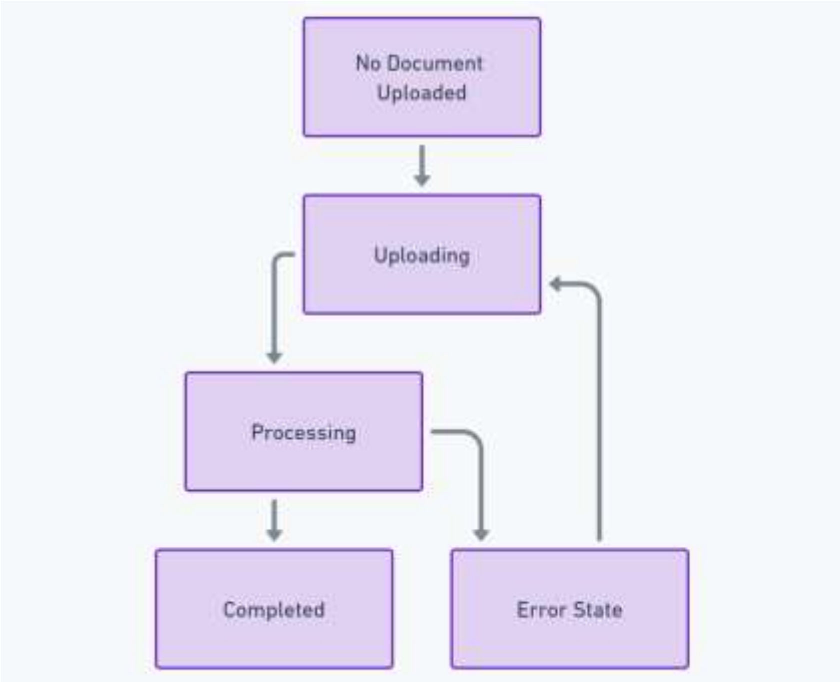
# ER Diagram (Entity-Relationship Diagram)



# Sequence Diagram



# State Diagram



# Sprint 2 Recap

**Sprint Goals:**

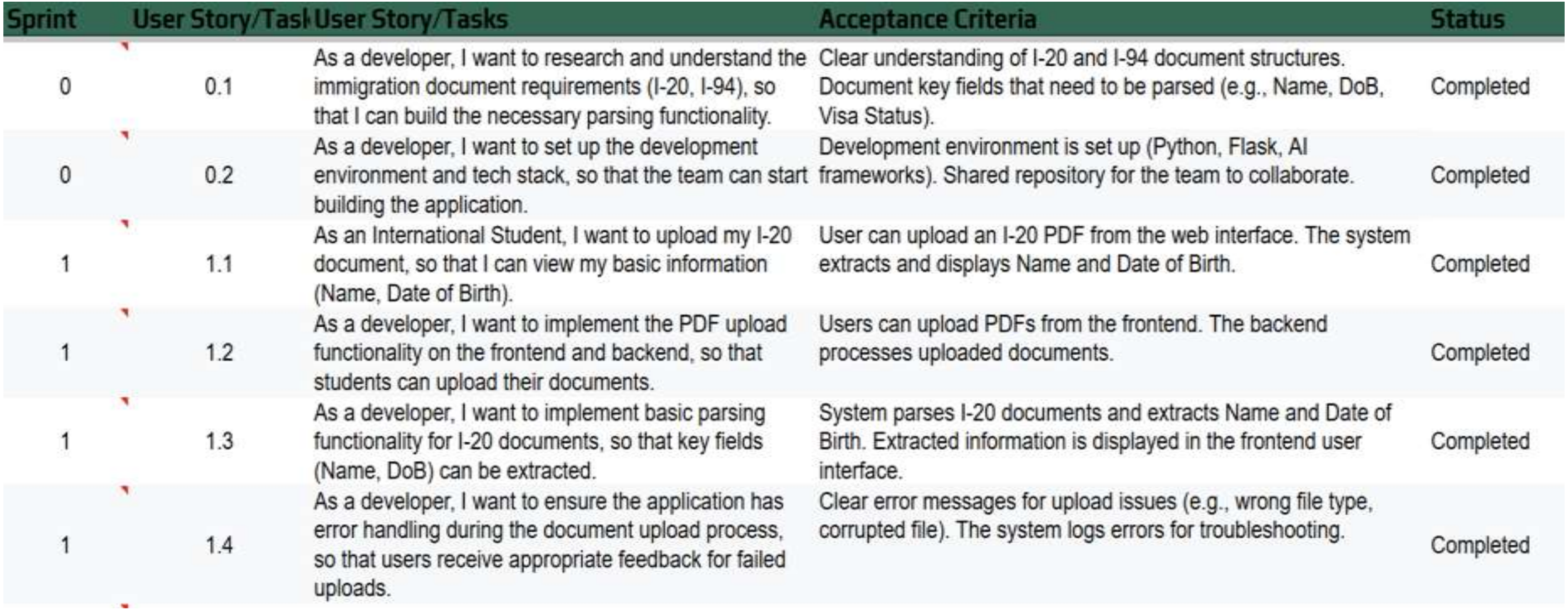
**Objective**: Introduce AI-driven functionality and enhance the user interface for the Immigration AI Advisor.

- Implement a basic AI model to provide personalized advice.
- Ensure flexibility for handling complex parsing in the future.
- Develop a front-end chatbox for user interaction.

**Key Accomplishments:**

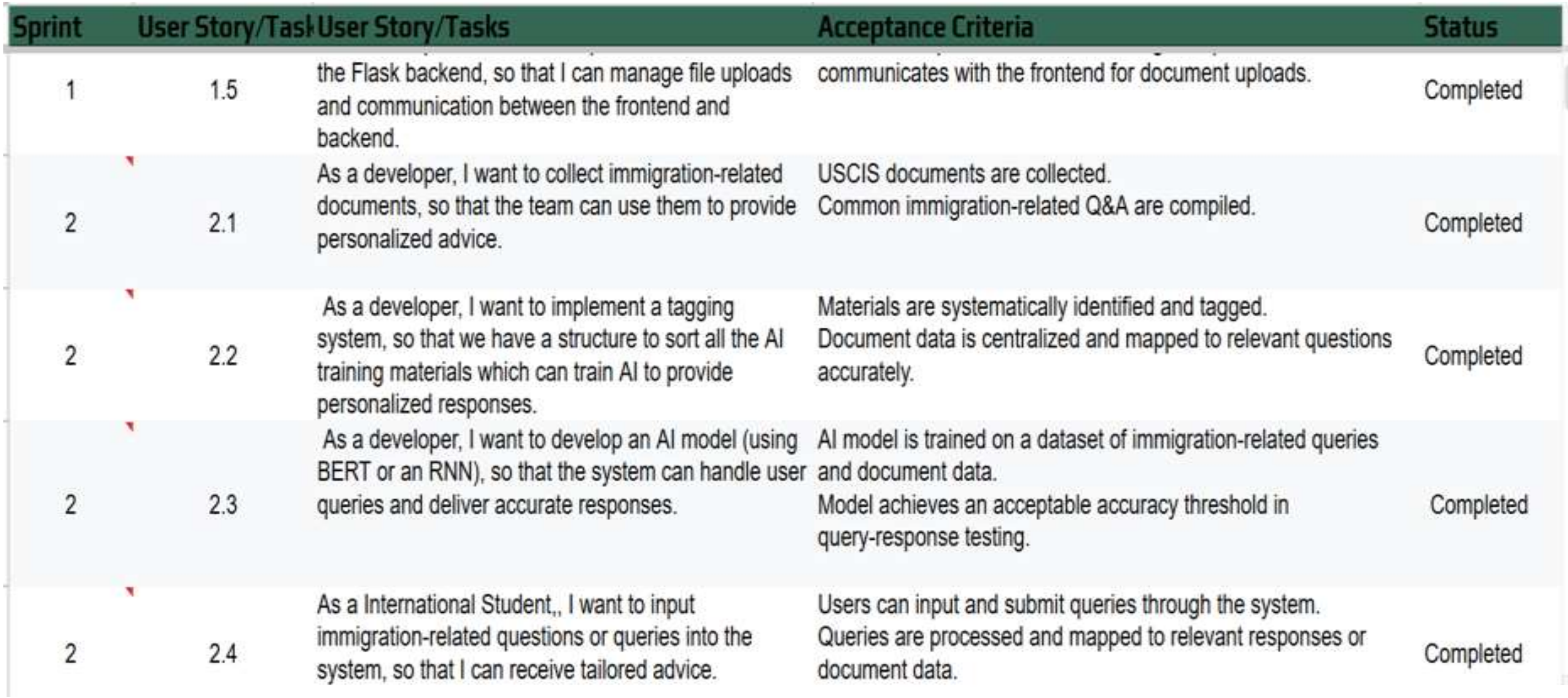
- AI Model Development:Successfully implemented an AI model (using BERT or an RNN) trained on immigration-related queries and document data
- Front-End Chatbox**:**Developed a user-friendly and functional chatbox.
- Tagging System:Created a systematic tagging system for organizing and structuring AI training materials.
- Document Collection:Gathered USCIS documents and compiled common immigration-related Q&A to train the AI model effectively.

# Project Backlog
# Stories or Tasks for the Whole Project

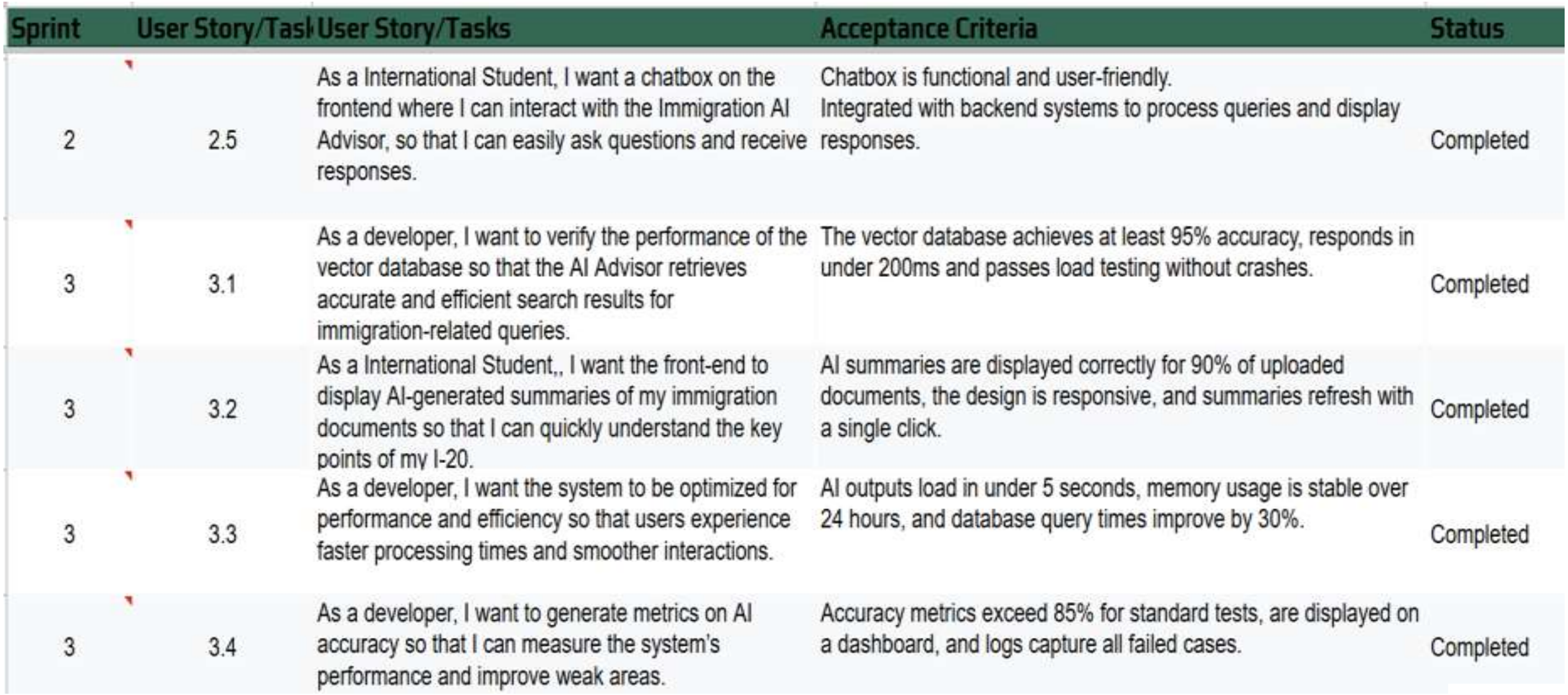


| Sprint | User Story/Task | User Story/Tasks | Acceptance Criteria | Status |
|---|---|---|---|---|
| 0 | 0.1 | As a developer, I want to research and understand the immigration document requirements (I-20, I-94), so that I can build the necessary parsing functionality. | Clear understanding of I-20 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 |

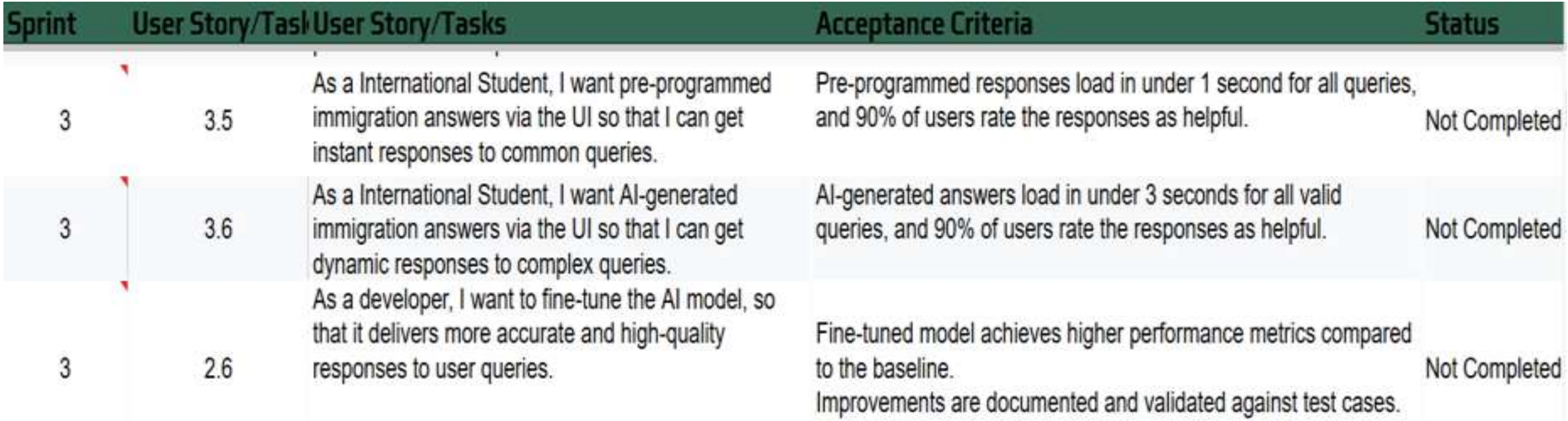
BORDERLESS.AI



| Sprint | User Story/Task | User Story/Tasks | Acceptance Criteria | Status |
|---|---|---|---|---|
| 1 | 1.5 | the Flask backend, so that I can manage file uploads and communication between the frontend and 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.<br>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.<br>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.<br>Model achieves an acceptable accuracy threshold in query-response testing. | Completed |
| 2 | 2.4 | As a International Student,, 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.<br>Queries are processed and mapped to relevant responses or document data. | Completed |



| Sprint | User Story/Task | User Story/Tasks | Acceptance Criteria | Status |
|---|---|---|---|---|
| 2 | 2.5 | As a International Student, 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.<br>Integrated with backend systems to process queries and display responses. | Completed |
| 3 | 3.1 | As a developer, I want to verify the performance of the vector database so that the AI Advisor retrieves accurate and efficient search results for immigration-related queries. | The vector database achieves at least 95% accuracy, responds in under 200ms and passes load testing without crashes. | Completed |
| 3 | 3.2 | As a International Student,, I want the front-end to display AI-generated summaries of my immigration documents so that I can quickly understand the key points of my I-20. | AI summaries are displayed correctly for 90% of uploaded documents, the design is responsive, and summaries refresh with a single click. | Completed |
| 3 | 3.3 | As a developer, I want the system to be optimized for performance and efficiency so that users experience faster processing times and smoother interactions. | AI outputs load in under 5 seconds, memory usage is stable over 24 hours, and database query times improve by 30%. | Completed |
| 3 | 3.4 | As a developer, I want to generate metrics on AI accuracy so that I can measure the system's performance and improve weak areas. | Accuracy metrics exceed 85% for standard tests, are displayed on a dashboard, and logs capture all failed cases. | Completed |



| Sprint | User Story/Tasl | User Story/Tasks | Acceptance Criteria | Status |
|---|---|---|---|---|
| 3 | 3.5 | As a International Student, I want pre-programmed immigration answers via the UI so that I can get instant responses to common queries. | Pre-programmed responses load in under 1 second for all queries, and 90% of users rate the responses as helpful. | Not Completed |
| 3 | 3.6 | As a International Student, I want AI-generated immigration answers via the UI so that I can get dynamic responses to complex queries. | AI-generated answers load in under 3 seconds for all valid queries, and 90% of users rate the responses as helpful. | Not Completed |
| 3 | 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.<br>Improvements are documented and validated against test cases. | Not Completed |

# Sprint Summary (Sprint 0)

- Problem: International students face delays in getting immigration advice due to understaffed offices and complex requirements, creating a pressing need for an automated solution
- Solution: AI-based application that reads immigration documents (I-20, I-94) and provides immediate, accurate advice based on current regulations
- Technology Stack: Discussed combination of modern tools - Java/Python/ReactJS for development, TensorFlow/PyTorch for AI implementation, focusing on document analysis capabilities
- Core Deliverable: Development of an AI advisor that can interpret immigration documents and cross-reference them with latest regulations to provide accurate guidance
- Quality Standards: Mandatory 48-hour code review period before deadlines, with strict testing requirements and no last-minute submissions allowed
- Communication Protocol: 24-hour response requirement for messages, mandatory meeting attendance, and regular Jira updates to maintain project momentum
- Development Process: Agile methodology with clear documentation in GitHub, regular retrospectives, and continuous improvement focus through sprint reviews and adjustments

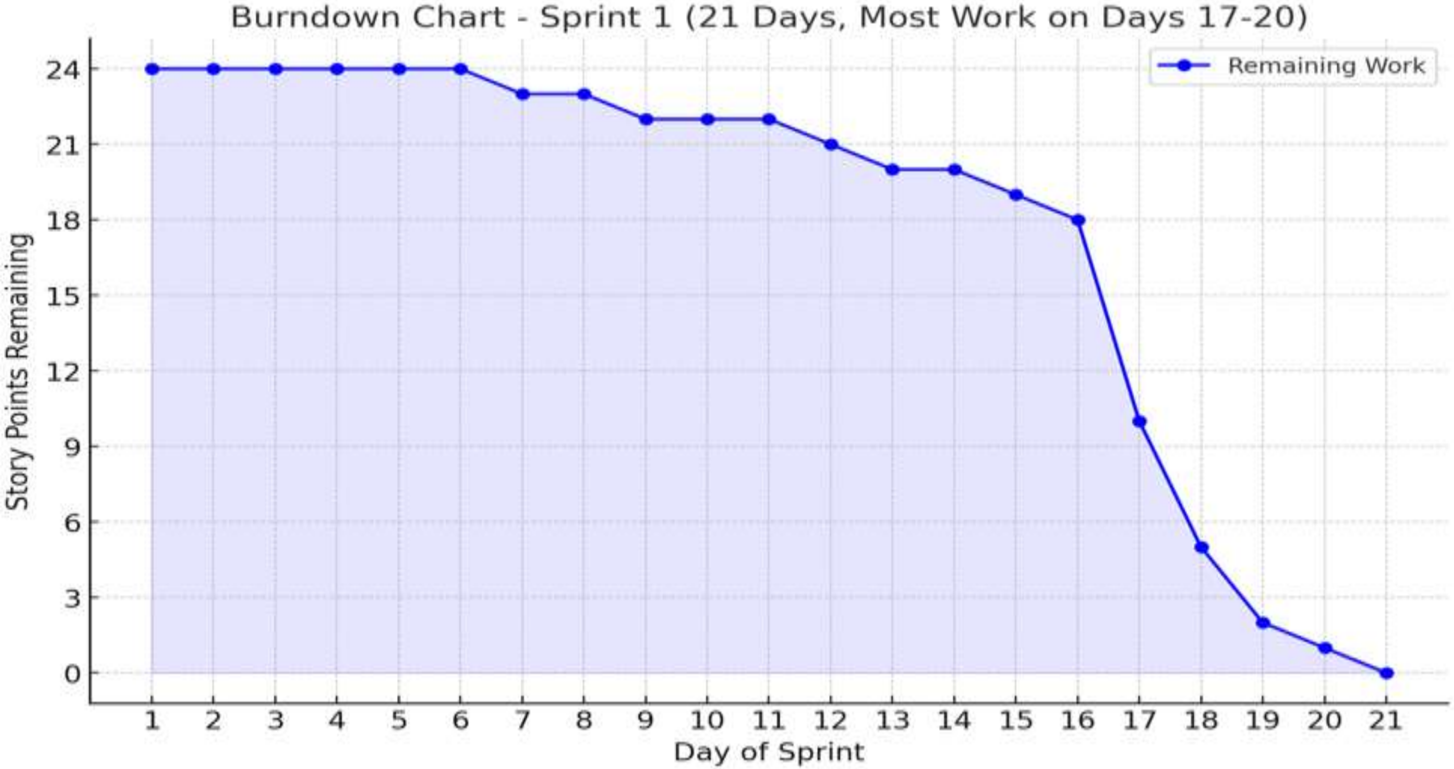
BORDERLESS.AI

# Sprint Summary (Sprint 1)

- Successfully implemented core MVP functionality - PDF upload and basic document parsing for I-20 forms, extracting key fields like Name and Date of Birth
- Achieved 100% completion rate (24/24 story points) for all planned user stories and tasks, demonstrating effective sprint planning and execution
- Established full-stack infrastructure with Python/Flask backend and ReactJS frontend, setting up essential communication between layers
- Implemented comprehensive error handling system for document uploads, including user-friendly messages and system logging
- Set up project architecture following best practices, as shown in architecture diagrams with clear separation of frontend, backend, AI component, and database layers
- Completed all essential test cases across different components (frontend upload, backend processing, document parsing, error handling)
- Created working prototype demonstrated in project demo, showing successful document upload and information extraction capabilities
- Developed robust team workflow with regular meetings (Monday/Wednesday/Friday schedule) and clear communication channels (Zoom, Group Texts, Jira)
- Identified areas for improvement in sprint retrospective, including task estimation and workload distribution, leading to actionable items for Sprint 2
- Prepared foundation for Sprint 2 by planning I-94 document integration and AI component enhancement, with clear acceptance criteria defined

BORDERLESS.AI

# Sprint Summary (Sprint 1,Burndown chart)

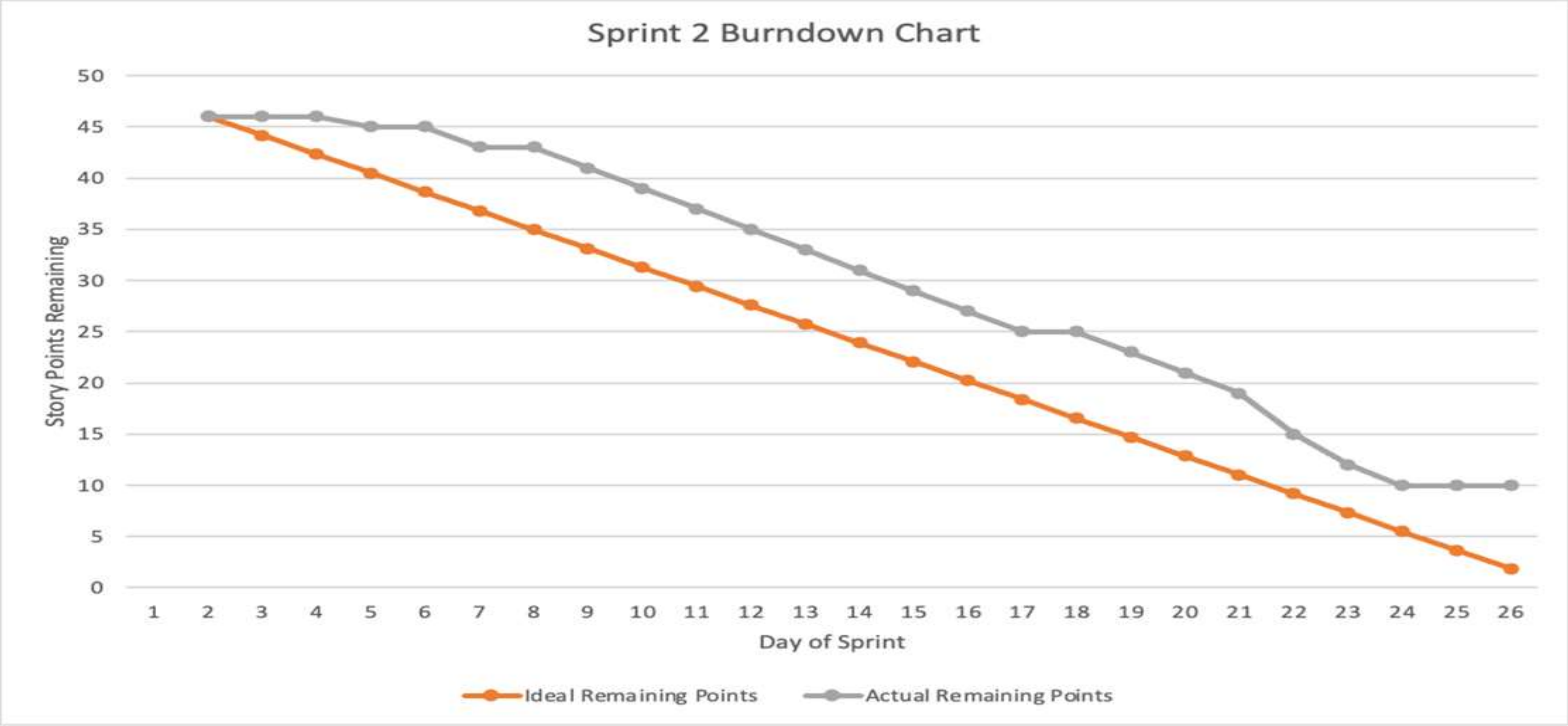


# Sprint Summary (Sprint 2)

- The team successfully completed 5 out of 6 planned user stories, achieving a completion rate of 78.28% (36 out of 46 story points)
- Major accomplishments included implementing the AI model with BERT/RNN for handling immigration queries and document analysis
- The team developed a functional chatbox interface integrated with backend systems for user interaction
- A tagging system was implemented to organize AI training materials and map documents to relevant queries
- The team successfully collected and compiled immigration-related documents and Q&A content from USCIS
- One task remained incomplete: fine-tuning the AI model for improved accuracy (10 story points)
- The frontend design was improved and integrated with the backend for document processing and query handling
- Basic AI functionality was implemented, allowing the system to provide personalized immigration advice
- The team identified communication as an area for improvement, particularly regarding WhatsApp responsiveness and meeting attendance
- For Sprint 3, the team committed to stricter deadlines, including completing components 72 hours before due dates and conducting daily standups

BORDERLESS.AI

# Sprint Summary (Sprint 2,Burndown chart)



# Sprint 3 Backlog
# Stories or Tasks committed for this Sprint

**User Story/Task ID: 3.1**

- **Description:** As a developer, I want to verify the performance of the vector database so that the AI Advisor retrieves accurate and efficient search results for immigration-related queries.
- **Acceptance Criteria:** The vector database achieves at least 95% accuracy, responds in under 200ms, and passes load testing without crashes.
- **Story Points:** 3

**User Story/Task ID: 3.2**

- **Description:** As a student, I want the front-end to display AI-generated summaries of my immigration documents so that I can quickly understand the key points of my I-20.
- **Acceptance Criteria:** AI summaries are displayed correctly for 90% of uploaded documents, the design is responsive, and summaries refresh with a single click.
- **Story Points:** 3

**User Story/Task ID: 3.3**

- **Description:** As a developer, I want the system to be optimized for performance and efficiency so that users experience faster processing times and smoother interactions.
- **Acceptance Criteria:** AI outputs load in under 5 seconds, memory usage is stable over 24 hours, and database query times improve by 30%.
- **Story Points:** 5

BORDERLESS.AI

# Sprint 3 Backlog
## Stories or Tasks committed for this Sprint

**User Story/Task ID: 3.4**

- **Description:** As a developer, I want to generate metrics on AI accuracy so that I can measure the system's performance and improve weak areas.
- **Acceptance Criteria:** Accuracy metrics exceed 85% for standard tests, are displayed on a dashboard, and logs capture all failed cases.
- **Story Points:** 8

**User Story/Task ID: 3.5**

- **Description:** As a student, I want pre-programmed immigration answers via the UI so that I can get instant responses to common queries.
- **Acceptance Criteria:** Pre-programmed responses load in under 1 second for all queries, and 90% of users rate the responses as helpful.
- **Story Points:** 8

# Sprint 3 Backlog
# Stories or Tasks committed for this Sprint

**User Story/Task ID: 3.6**

- **Description:** As a student, I want AI-generated immigration answers via the UI so that I can get dynamic responses to complex queries.
- **Acceptance Criteria:** AI-generated answers load in under 3 seconds for all valid queries, and 90% of users rate the responses as helpful.
- **Story Points:** 13

**User Story/Task ID: 2.6**

- **Description:** 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, and improvements are documented and validated against test cases.
- **Story Points:** 5

# Sprint 3 Backlog
# Test Cases

**Test Case 1: Verify End-to-End System Workflow**

- **Description:** Ensure the system processes I-20 uploads and generates summaries and responses seamlessly.
- **Steps:** Upload an I-20 document, view the summary, and submit a question.
- **Expected Result:** The system completes all actions smoothly, with summaries and responses delivered within 5 seconds total.

**Test Case 2: Verify Vector Database Accuracy**

- **Description:** Ensure the vector database retrieves accurate results for test queries.
- **Steps:** Upload sample documents, execute test queries, and compare results to expected outputs.
- **Expected Result:** At least 95% of queries return accurate results.

**Test Case 3: Verify Front-End Display of AI Summaries**

- **Description:** Confirm that AI-generated summaries display correctly in the UI.
- **Steps:** Upload an I-20 document and check the "Summary" section of the UI.
- **Expected Result:** Summaries are displayed accurately and match backend outputs.

# Sprint 3 Backlog
# Test Cases

**Test Case 4: Test Responsiveness of the Summary Section**

- **Description:** Ensure the AI summaries section is responsive on different devices.
- **Steps:** Open the web application on desktop, tablet, and mobile, and check the layout.
- **Expected Result:** The summary section is fully responsive on all screen sizes.

**Test Case 5: Verify System Performance Under Load**

- **Description:** Test system stability and response times with multiple concurrent users.
- **Steps:** Simulate 50 users submitting queries simultaneously and monitor system behavior.
- **Expected Result:** The system remains stable, with no crashes and response times under 5 seconds.

# Sprint 3 Backlog
# Test Cases

**Test Case 6: Validate Accuracy of Pre-Programmed Answers**

- **Description:** Verify pre-programmed answers match predefined FAQs accurately.
- **Steps:** Submit 10 predefined FAQ queries and compare responses with expected answers.
- **Expected Result:** All pre-programmed answers are correct and match expected outputs.

**Test Case 7: Test AI-Generated Answer Response Time**

- **Description:** Ensure AI-generated answers are delivered within acceptable time limits.
- **Steps:** Submit a complex immigration-related question and measure response time.
- **Expected Result:** AI-generated answers load in under 3 seconds.

# Sprint 3  Backlog Stories Completed

- **User Story/Task ID:** 3.1
  - As a developer, I want to verify the performance of the vector database so that the AI Advisor retrieves accurate and efficient search results for immigration-related queries.
- **User Story/Task ID:** 3.2
  - As a student, I want the front-end to display AI-generated summaries of my immigration documents so that I can quickly understand the key points of my I-20.
- **User Story/Task ID:** 3.3
  - As a developer, I want the system to be optimized for performance and efficiency so that users experience faster processing times and smoother interactions.
- **User Story/Task ID:** 3.4
  - As a developer, I want to generate metrics on AI accuracy so that I can measure the system’s performance and improve weak areas.
- **User Story/Task ID:** 3.5
  - As a student, I want pre-programmed immigration answers via the UI so that I can get instant responses to common queries.

# Sprint 3  Backlog Stories Not Completed

- **User Story/Task ID:** 3.6
    - As a student I want AI-generated immigration answers via the UI so that I can get dynamic responses to complex queries.

- **User Story/Task ID:** 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.

# Metrics: Sprint 3 Velocity

- Total Story Points: 45
- Completed Story Points: 27
- Not Completed Story Points: 18
- **Velocity: 27**

# Metrics: Team's Historical Velocity and Average

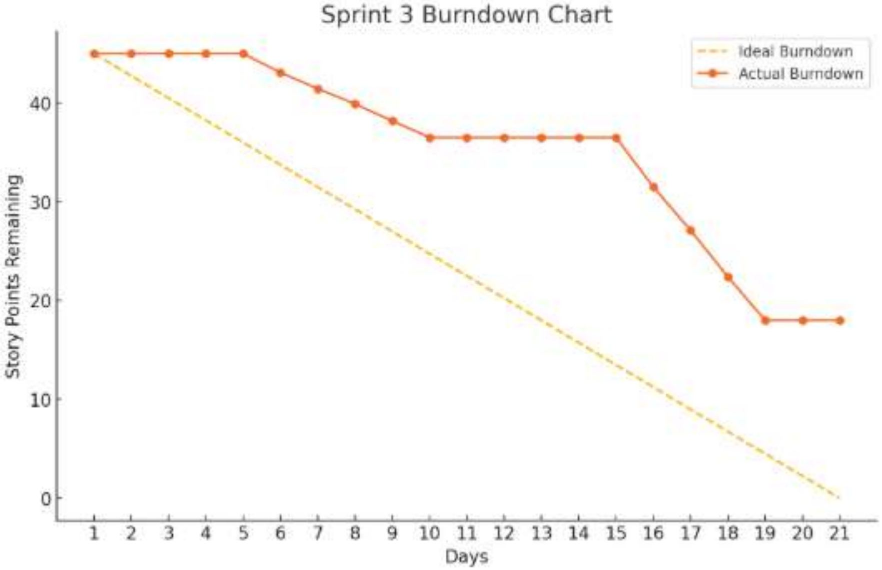
Sprint 1 **Velocity: 24**

Sprint 2 **Velocity: 36**

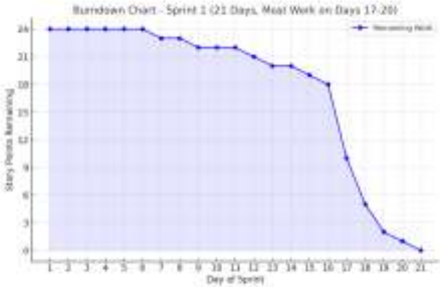
Sprint 3 **Velocity: 27**

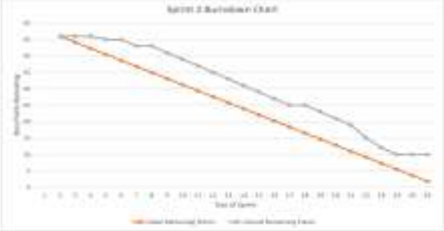
**Average: 29**

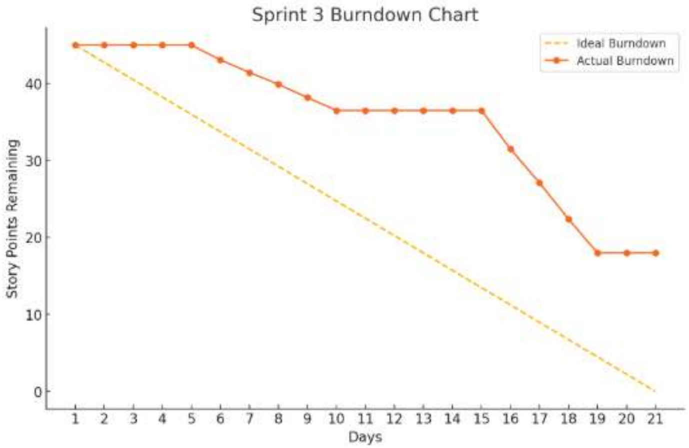
# Metrics Sprint 3 Burndown Chart



# Metrics: Historical Burndown Chart







# Metrics: Completed/Committed Ratio

**Sprint 3:**

Completed: 27
Not Completed: 18
Total Story Points: 45

**Completed/Committed Ratio: 27/45 = 60%**

# Team's historical Completed/Committed Ratio

Sprint 1:

24/24 = 100%

Sprint 2:

36/46 = 78.26%

Sprint 3:

27/45 = 60%

Average 87/ 115 = **75.7%**

# Retrospective

**What Went Well?**

Completed many user stories despite busy schedules.

Positive feedback and strong teamwork; pulled everything together successfully.

Improved coordination with tools like Jira and better communication (e.g., text responses).

**What Can Be Improved?**

Create more sub-tasks to ensure stories don't fall behind.

Better monitor and manage items to reduce stress.

Respect deadlines while balancing personal responsibilities.

**Actionable Items**:

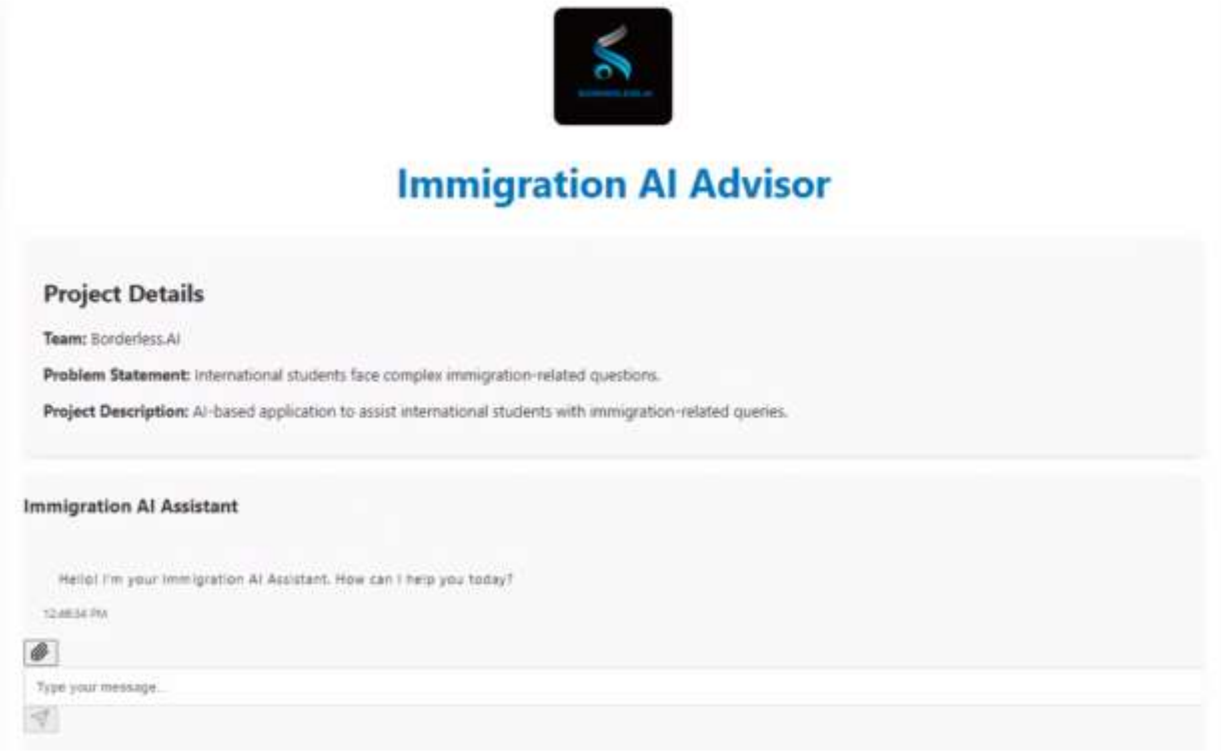
Divide tasks into smaller, time-bound sub-tasks.

Make daily personal detailed to-do lists for better personal and work task tracking.

Practice active listening , respect and follow through on shared information.

BORDERLESS.AI

# Project Demo



## Immigration AI Advisor

### Project Details

**Team:** Borderless.AI

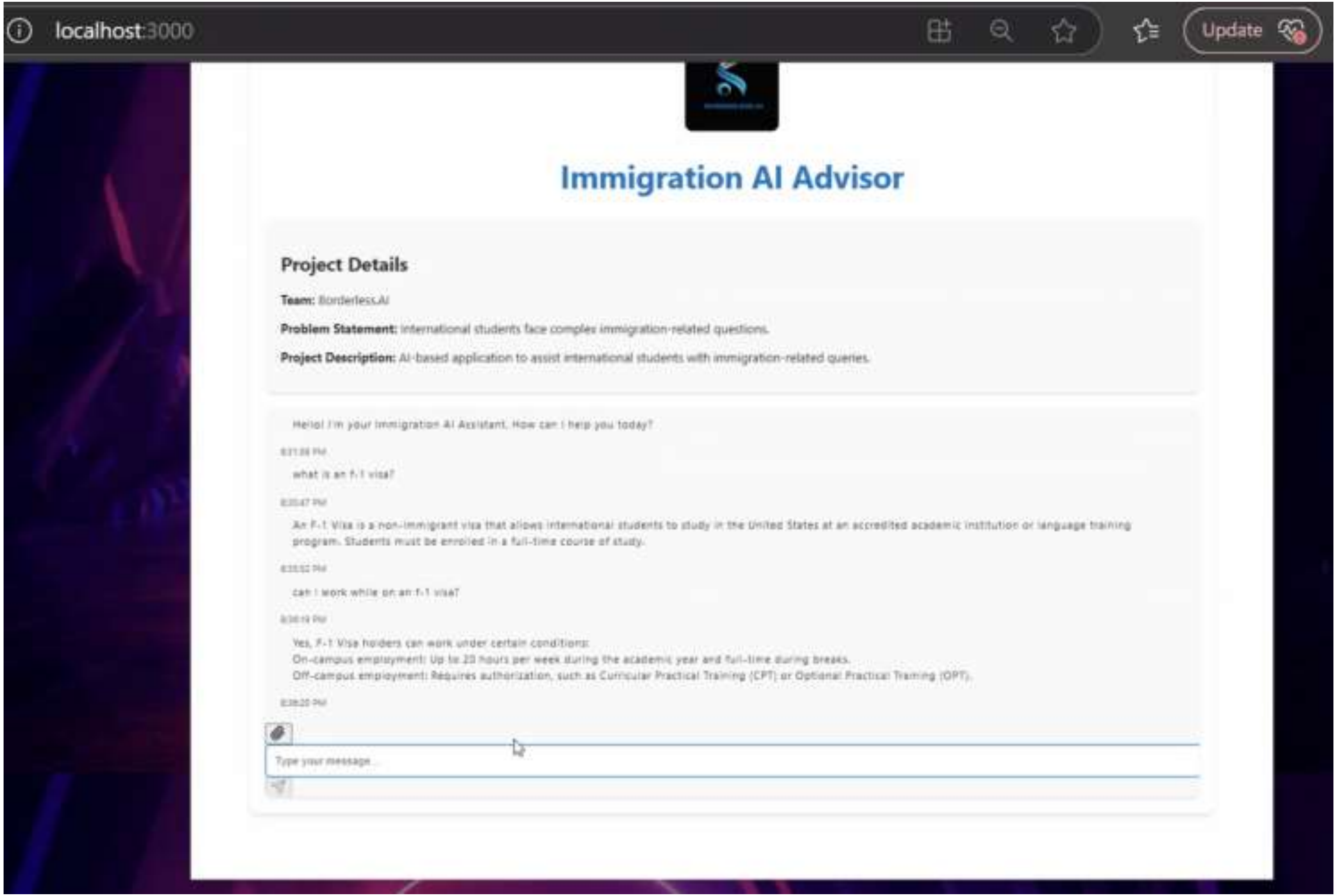
**Problem Statement:** International students face complex immigration-related questions.

**Project Description:** AI-based application to assist international students with immigration-related queries.

### Immigration AI Assistant

Hello! I'm your Immigration AI Assistant. How can I help you today?

12:48:34 PM



# Immigration AI Advisor

## Project Details

**Team:** Borderless.AI

**Problem Statement:** International students face complex immigration-related questions.

**Project Description:** AI-based application to assist international students with immigration-related queries.

Hello! I'm your Immigration AI Assistant. How can I help you today?

what is an f-1 visa?

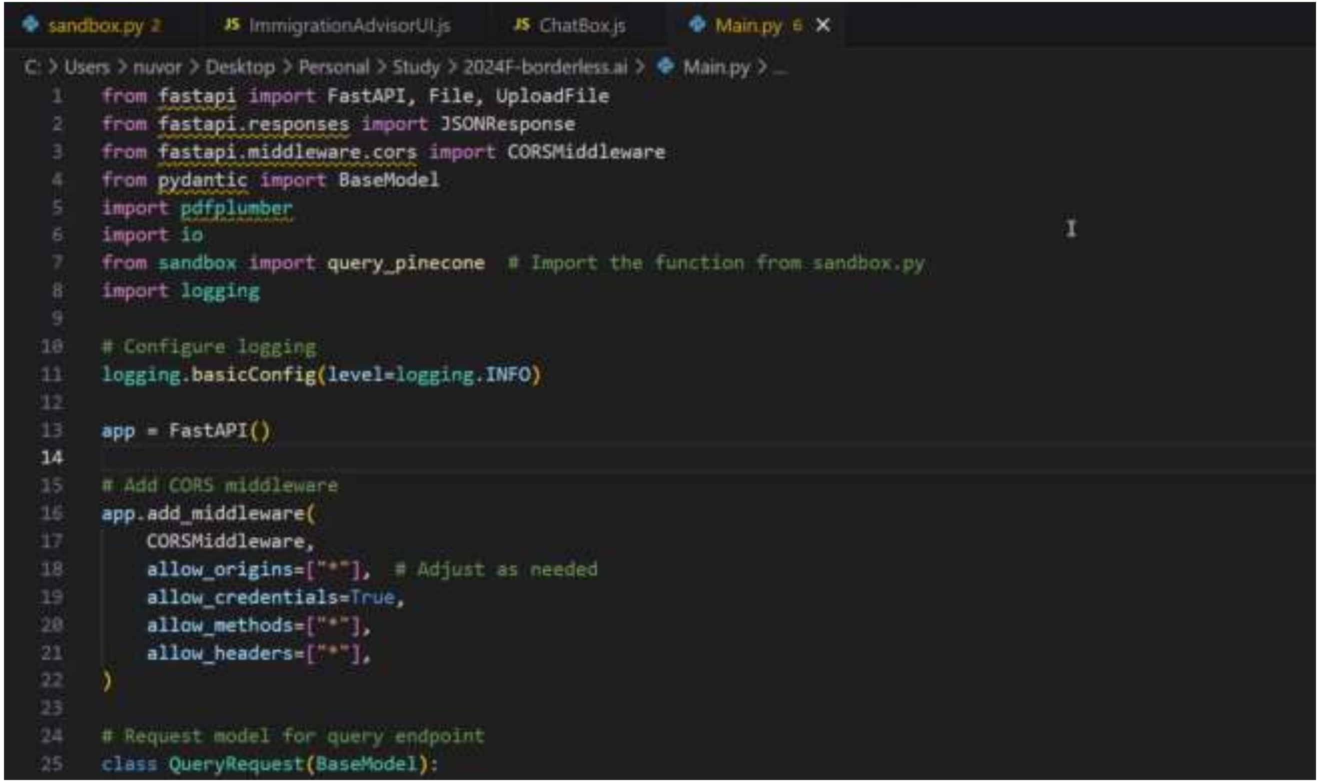
An F-1 Visa is a non-immigrant visa that allows international students to study in the United States at an accredited academic institution or language training program. Students must be enrolled in a full-time course of study.

can i work while on an f-1 visa?

Yes, F-1 Visa holders can work under certain conditions:
On-campus employment: Up to 20 hours per week during the academic year and full-time during breaks.
Off-campus employment: Requires authorization, such as Curricular Practical Training (CPT) or Optional Practical Training (OPT).

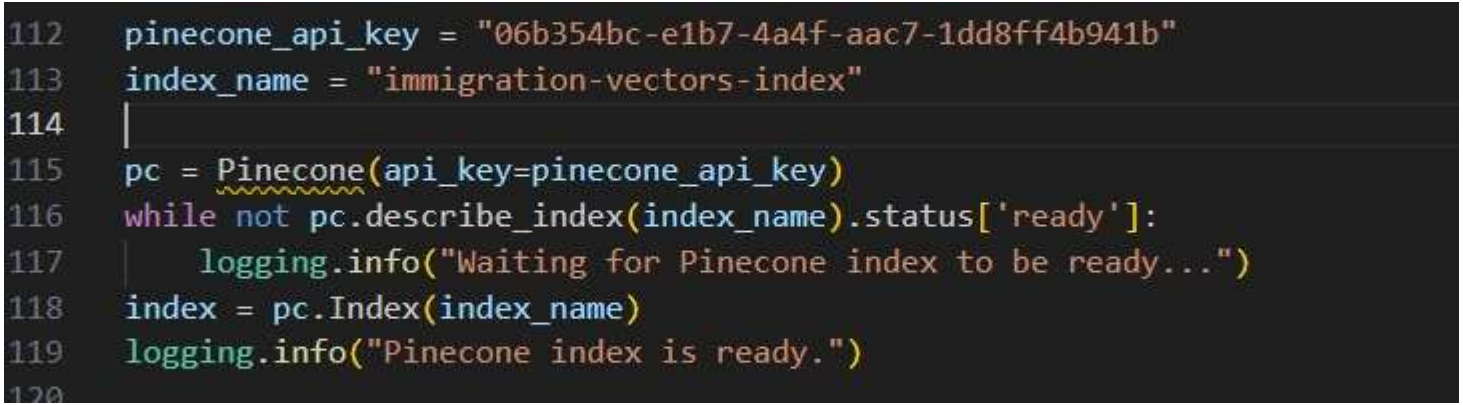
BORDERLESS.AI

# Project Demo


sandbox.py 2
ImmigrationAdvisorUI.js
ChatBox.js
Main.py 6
C: > Users > nuvor > Desktop > Personal > Study > 2024F-borderless.ai > Main.py > ...

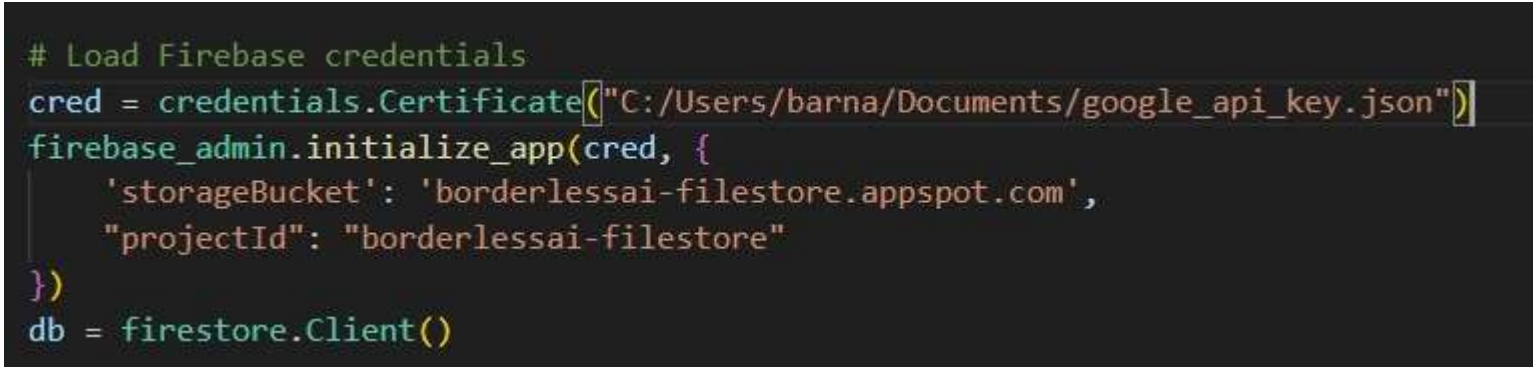
```python
from fastapi import FastAPI, File, UploadFile
from fastapi.responses import JSONResponse
from fastapi.middleware.cors import CORSMiddleware
from pydantic import BaseModel
import pdfplumber
import io
from sandbox import query_pinecone  # Import the function from sandbox.py
import logging

# Configure logging
logging.basicConfig(level=logging.INFO)

app = FastAPI()

# Add CORS middleware
app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],  # Adjust as needed
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)

# Request model for query endpoint
class QueryRequest(BaseModel):
```

# PINECONE API



```python
pinecone_api_key = "06b354bc-e1b7-4a4f-aac7-1dd8ff4b941b"
index_name = "immigration-vectors-index"

pc = Pinecone(api_key=pinecone_api_key)
while not pc.describe_index(index_name).status['ready']:
    logging.info("Waiting for Pinecone index to be ready...")
index = pc.Index(index_name)
logging.info("Pinecone index is ready.")
```

# FIREBASE API



```
# Load Firebase credentials
cred = credentials.Certificate("C:/Users/barna/Documents/google_api_key.json")
firebase_admin.initialize_app(cred, {
    'storageBucket': 'borderlessai-filestore.appspot.com',
    "projectId": "borderlessai-filestore"
})
db = firestore.Client()
```

# Github Link

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

# Live Application Demo

# THANK YOU!