

Team Johto

Alex Corona | Brandon Medina | Haoran Jin
| Prajwal Vinod Naik | Nihaad Saleem

Milestone 1.0 Demo + Agile Documentation

Table of Contents

- What Our Application Does
- Key Features
- Fulfillment of User Stories
- Agile Methods in Action
- Database Design
- Dashboard - code
- Flask Application Integration with Database (main.py)
- Database Initialization, Reset, and Seeding Process
- Repo & Documentation
- Team Contributions
- What's Next for Milestone 2.0
- Summary & Reflections

Student-Alumni Career Mentorship Dashboard

Purpose:

Bridge career paths between students and alumni via a Flask-based dashboard

The screenshot displays the Student-Alumni Career Mentorship Dashboard interface. At the top, there is a navigation bar with the title "Student-Alumni Career Mentorship Dashboard" and links for "Dashboard", "Career Path", "Mentors", and "Logout".

The main content area includes the following sections:

- Welcome, Alex!**: A greeting message with the text "Connect with alumni and explore your career options."
- Alex Corona**: User profile information for a Computer Science Student located in Los Angeles, CA, with icons for messaging and connecting.
- Academic Progress**: A donut chart showing 36 Credits, divided into three categories: Completed (blue), In Progress (purple), and Not Started (green). The chart indicates that most credits are completed.
- Career Path**: A section for the Software Engineer career path, listing roles such as Intern, Junior Developer, and Frontend Developer.
- Find a Mentor**: A search form allowing users to filter mentors by degree (Any) and industry (Any), with a "Search" button.

Key Features

Login / Register page

Dashboard with profile card, donut chart
(academic progress)

Career Path section

Mentor search interface

Networking integrations (LinkedIn, GitHub,
Email)



Fulfillment of User Stories

Examples:

"As a student, I want to track academic progress."

"As a mentee, I want to find relevant mentors."

Stories mapped to Jira tasks and subtasks

Estimated durations and responsibilities assigned

Milestone 1.0 tasks aligned to SDF tickets

The screenshot shows the Jira web interface. In the top navigation bar, there are links for 'For you', 'Recent', 'Starred', 'Apps', 'Plans', and 'Projects'. The 'Software Development' project is selected, indicated by a blue border around its name and icon. Below the navigation, there are tabs for 'Summary', 'Timeline', 'Board', 'Calendar', 'List' (which is currently selected), 'Forms', 'Goals', 'All work', 'Development', 'Code', and 'More'. A search bar at the top right contains the text 'Search'. On the left, a sidebar lists 'Recent' projects like 'Yantra Ethica' and 'Software Development', along with 'Filters', 'Dashboards', and 'Teams'. At the bottom of the sidebar, there's a link to 'Customize sidebar'. The main content area displays a table titled 'Search list' with columns for 'Type', 'Key', 'Summary', 'Status', and 'Comments'. The table contains 12 rows, each representing a task or subtask under the 'Software Development' project. The tasks are: SDF-1 (Alumni Profile & Matching System, Status: TO DO, Comment: Add comment), SDF-2 (Student Mentor Suggestion & Connection System, Status: TO DO, Comment: Add comment), SDF-3 (Admin Analytics Dashboard, Status: TO DO, Comment: Add comment), SDF-4 (Project Management & Documentation, Status: TO DO, Comment: Add comment), SDF-5 (Developer RAG Integration (Optional), Status: TO DO, Comment: Add comment), SDF-15 (Student Mentor Suggestion & Connection System, Status: TO DO, Comment: Add comment), SDF-28 (Project Management & Documentation, Status: IN PROGRESS, Comment: Add comment), SDF-22 (Admin Analytics Dashboard, Status: IN PROGRESS, Comment: Add comment), and SDF-33 (Developer RAG Integration – Optional, Status: TO DO, Comment: Add comment). A 'Create' button is located at the bottom of the table. A purple banner at the bottom right says 'Quickstart'.

Type	Key	Summary	Status	Comments
> ⚡	SDF-1	Alumni Profile & Matching System	TO DO	Add comment
⚡	SDF-2	Student Mentor Suggestion & Connection System	TO DO	Add comment
⚡	SDF-3	Admin Analytics Dashboard	TO DO	Add comment
⚡	SDF-4	Project Management & Documentation	TO DO	Add comment
⚡	SDF-5	Developer RAG Integration (Optional)	TO DO	Add comment
> 📄	SDF-15	Student Mentor Suggestion & Connection System	TO DO	Add comment
> 📄	SDF-28	Project Management & Documentation	IN PROGRESS	Add comment
> 📄	SDF-22	Admin Analytics Dashboard	IN PROGRESS	Add comment
> 📄	SDF-33	Developer RAG Integration – Optional	TO DO	Add comment

Agile Methods in Action

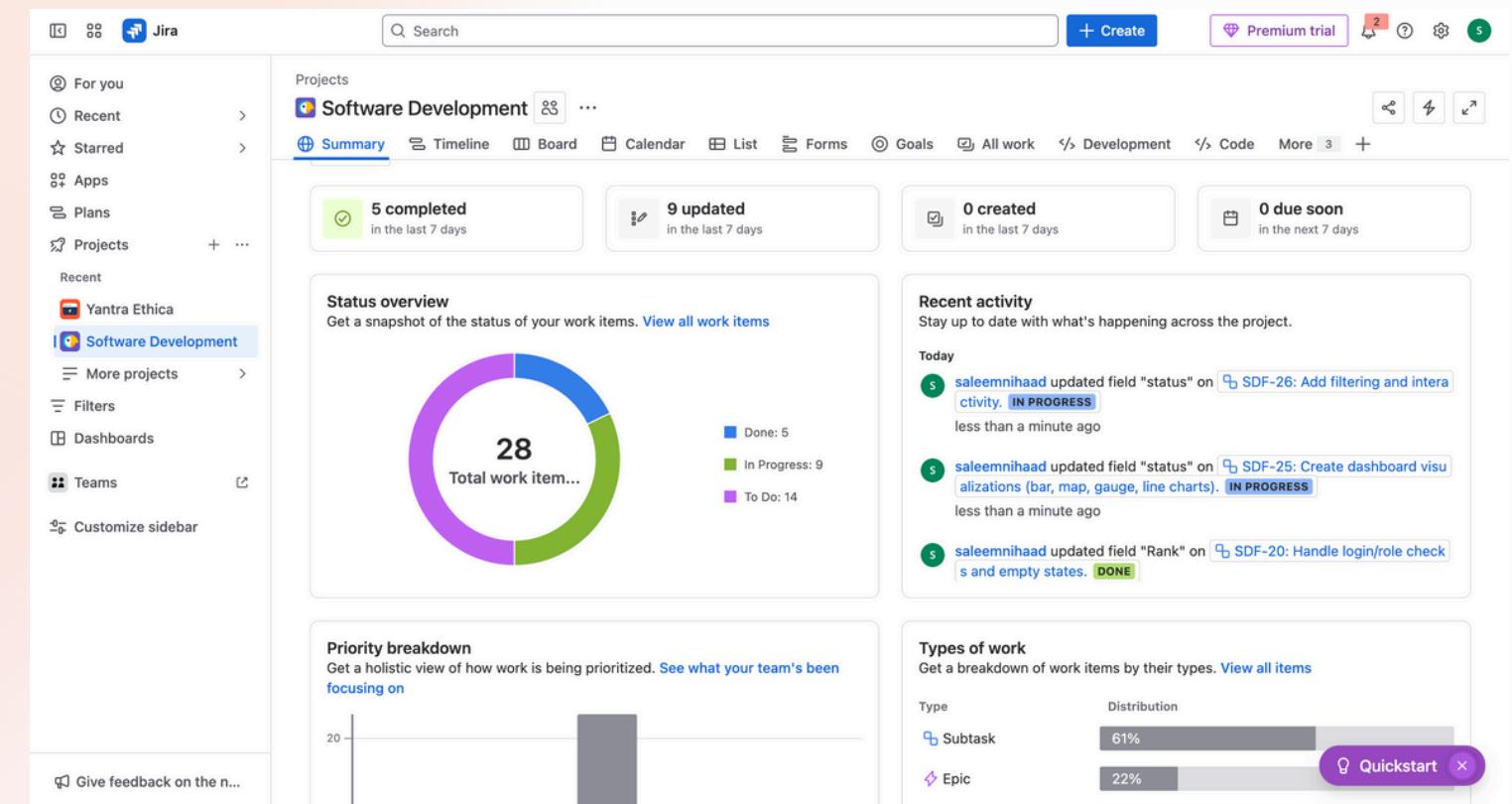
-Tool: Jira (Epics, Sprint Board, Subtasks)

-Burn down tracking: 28 work items total

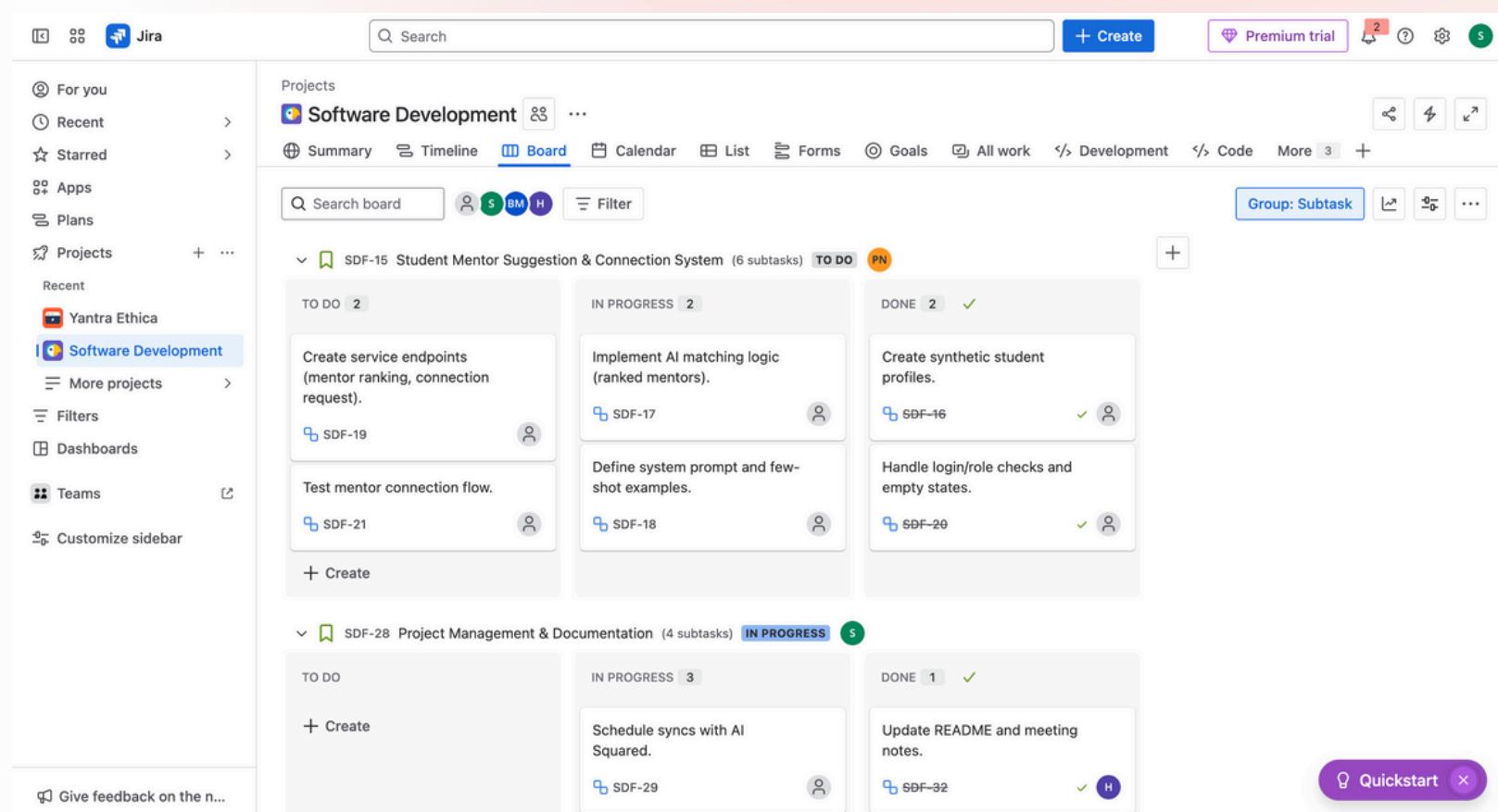
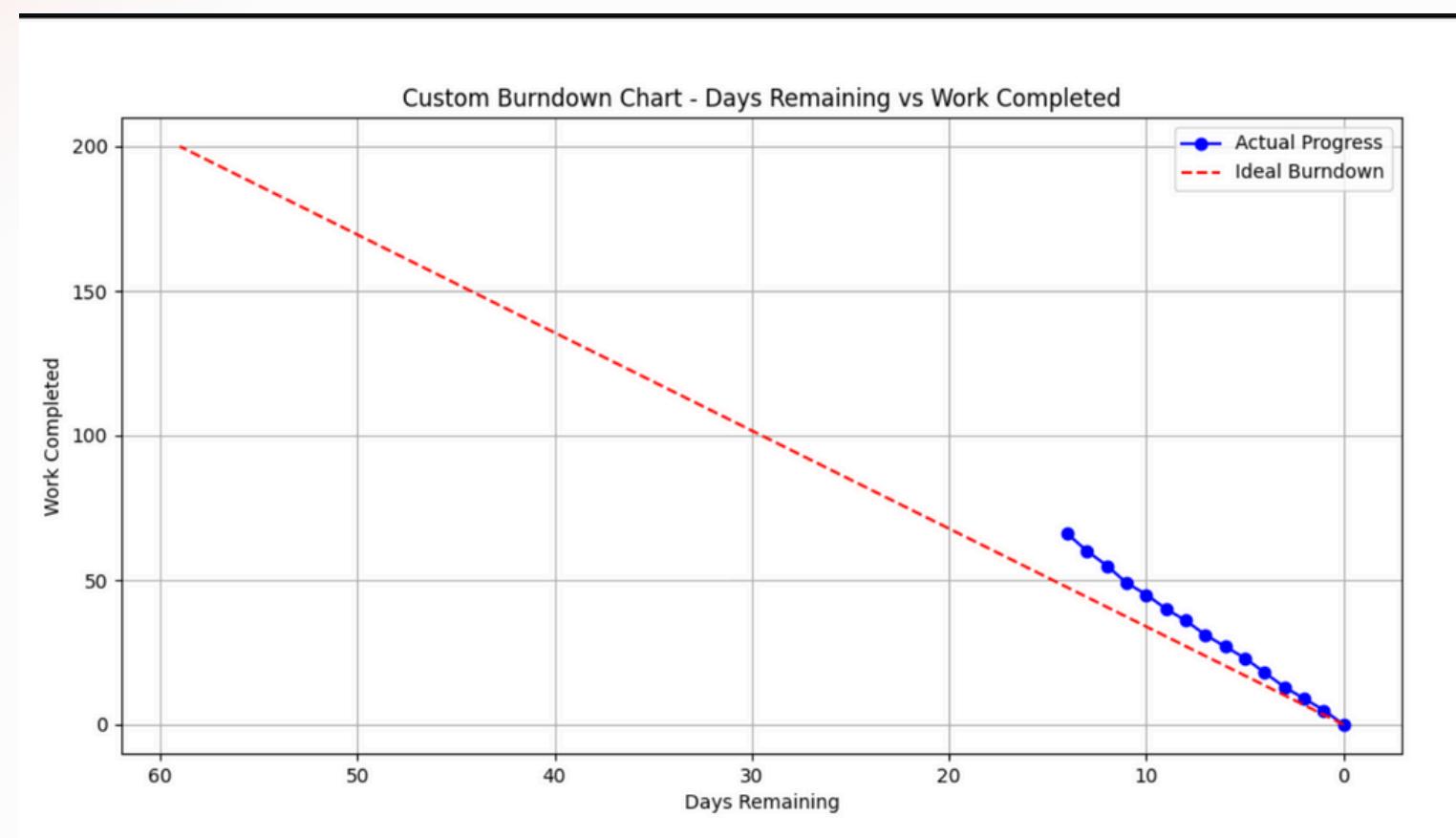
- o 14 To Do, 9 In Progress, 5 Done
- o

-Burndown Chart: Ideal vs Actual plotted clearly

-Standups held 2x/week



The screenshot shows the Jira Project Summary dashboard for the 'Software Development' project. The dashboard includes a sidebar with navigation links like 'For you', 'Recent', 'Starred', 'Plans', 'Projects', 'Recent', 'More projects', 'Filters', 'Dashboards', 'Teams', and 'Customize sidebar'. The main area features a 'Status overview' donut chart showing 28 total work items: 5 completed (green), 9 updated (blue), 0 created (grey), and 0 due soon (yellow). Below the chart are sections for 'Recent activity' (with a list of recent updates from 'saleemnihaad') and 'Types of work' (a distribution bar chart showing 61% Subtask and 22% Epic).



The screenshot shows the Jira Sprint Board view for the 'Software Development' project. The board has two columns: 'TO DO' and 'IN PROGRESS'. Under 'TO DO', there are three sub-tasks: 'Create service endpoints (mentor ranking, connection request)' (SDF-19), 'Test mentor connection flow.' (SDF-21), and 'Define system prompt and few-shot examples.' (SDF-18). Under 'IN PROGRESS', there are three sub-tasks: 'Implement AI matching logic (ranked mentors)' (SDF-17), 'Handle login/role checks and empty states.' (SDF-20), and 'Create synthetic student profiles.' (SDF-16). The sidebar on the left shows the same navigation links as the summary dashboard.

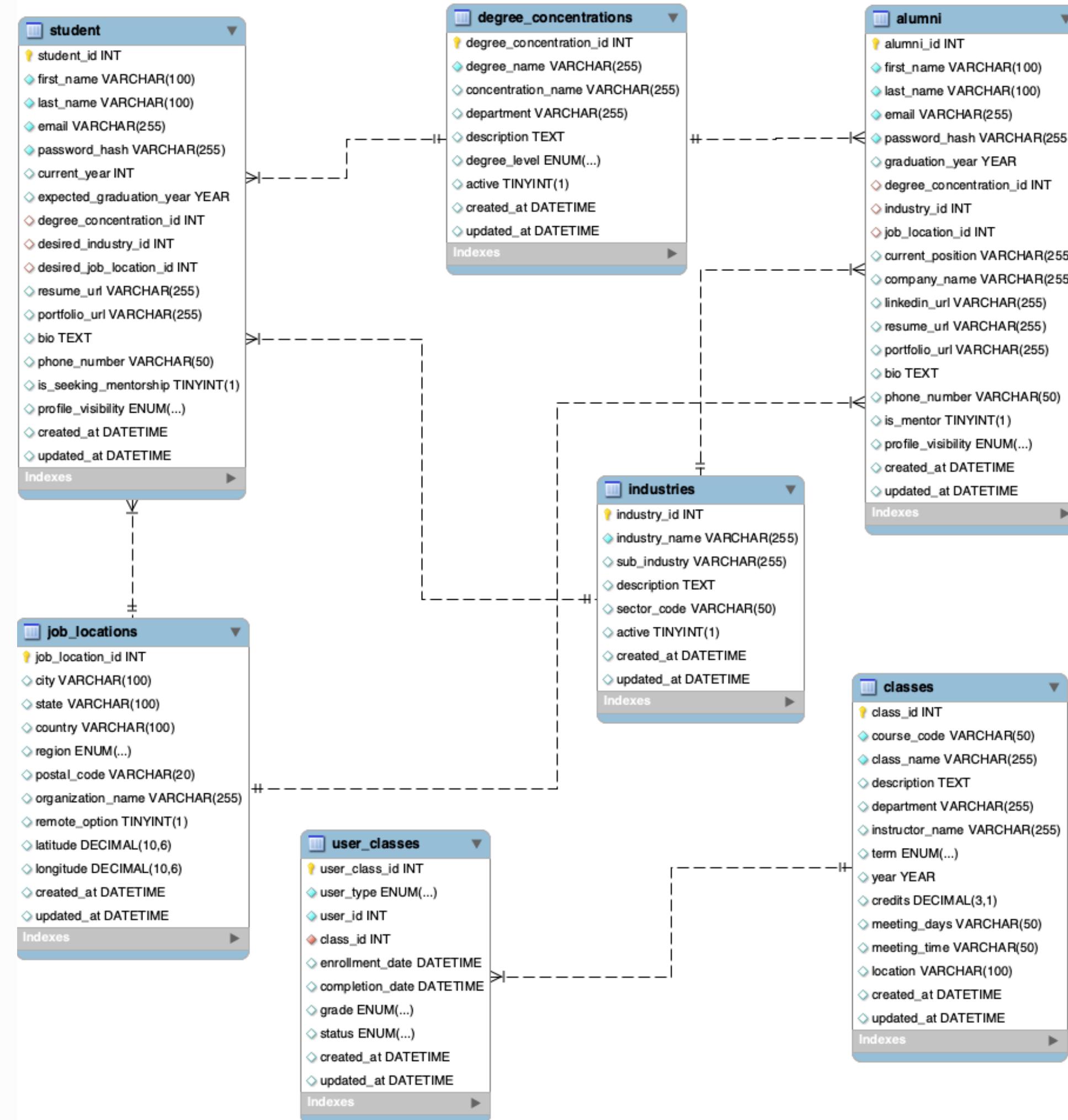
Database Design

Designed a normalized relational schema to manage students, alumni, courses, and mentorship connections.

Implemented referential integrity using foreign key constraints.

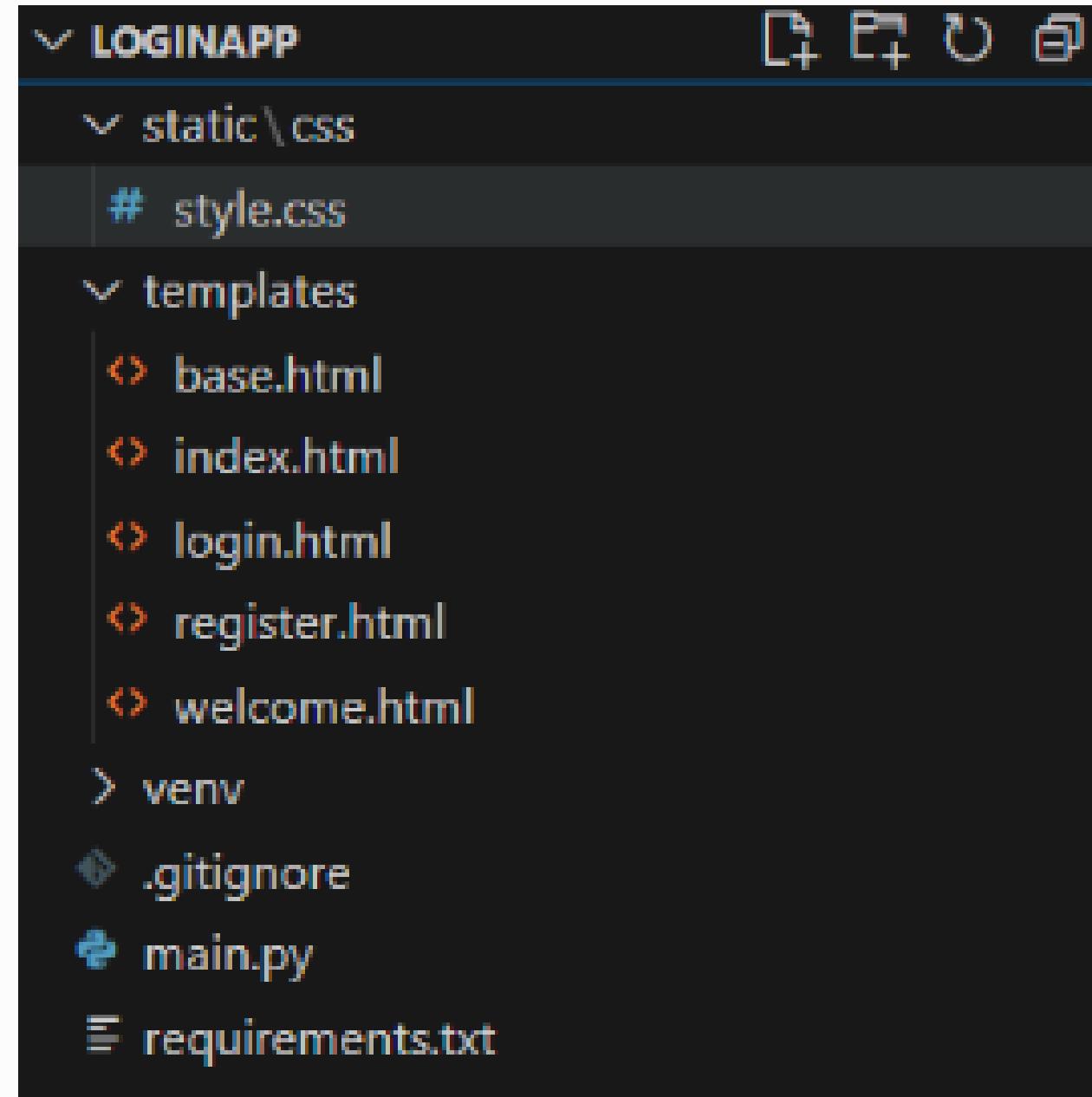
Core design goals: scalability, maintainability, and clear entity relationships.

Includes lookup tables (degree_concentrations, industries, job_locations) for structured metadata.



Login Page Design

- Register: As a visitor, I want to create an account with a username and password, so that I can log in to access the team dashboard.
- Login: As a registered user, I want to log in with my username and password, so that I can access the dashboard.
- To the Dashboard: As a logged-in user, I want to land on a clear welcome page with a button to the Dashboard, so that I know where to go next.
- Logout: As an authenticated user, I want to log out, so that others using the same device cannot access my session.



Login Code

- main.py--routing, database, session, server core
- style.css--dynamic gradient, glowing button
- base.html--skeleton, navigation
- login.html--login card, password eye
- register.html--set own password to login
- welcome.html--display username after successful login
- index.html--no forced login

The screenshot shows a GitHub repository named 'LoginApp' which is public. The repository has 1 branch and 0 tags. The commit history shows four commits from 'haoranjin' made 18 hours ago. The commits are:

- Update Flask run command in README
- Initial commit: Flask fancy login portal
- Initial commit: Flask fancy login portal
- Initial commit: Flask fancy login portal
- Update Flask run command in README
- Initial commit: Flask fancy login portal
- Initial commit: Flask fancy login portal

The README file contains the following text:

```
.\venv\Scripts\Activate flask --app main.py run
```

<http://127.0.0.1:5000>



JOHTO

Login Register

Create Account

Set a username & password

Username

Pick a username

Password

Create a password



Register

Already have an account? [Log in](#)



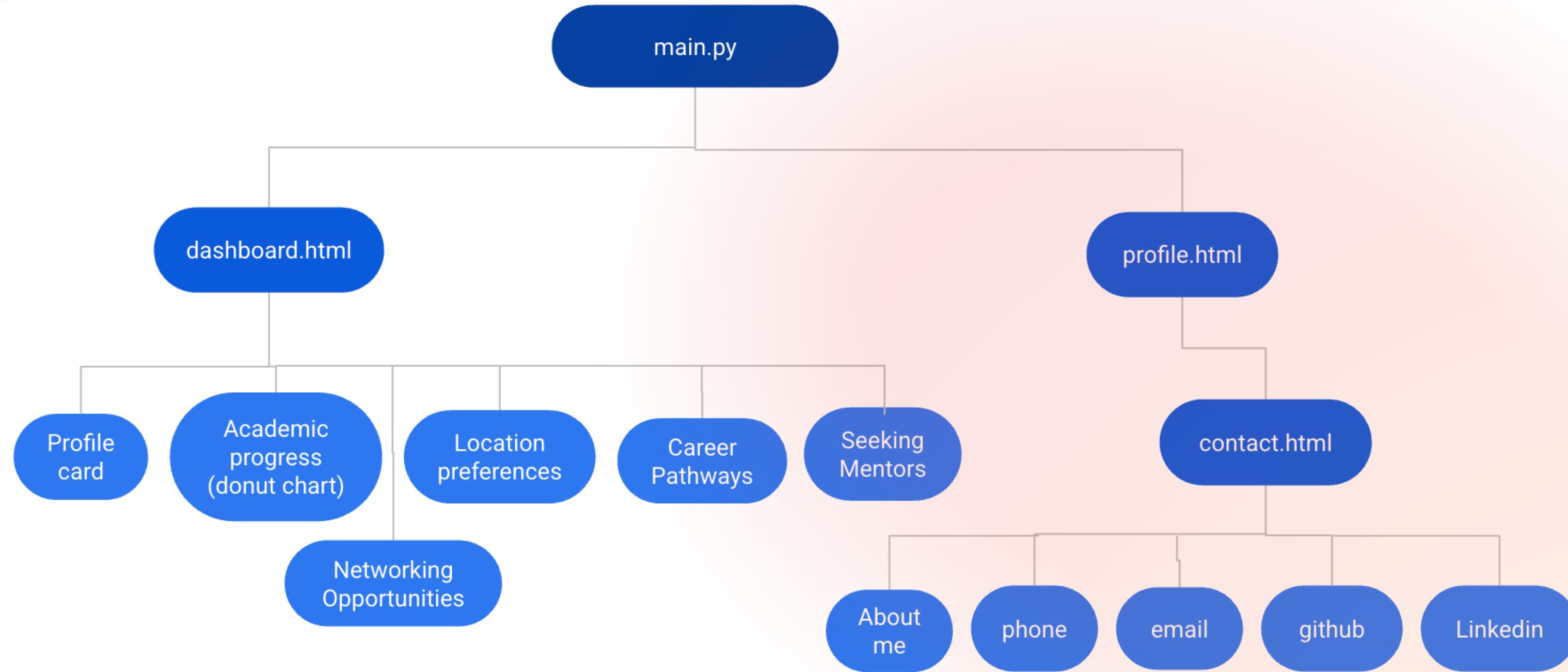
Login successful.

Welcome, haoranjin!

Login successful. This is your welcome page.

[Connect to the Dashboard](#)

Dashboard – code



Flask Application Integration with Database (main.py)

Connected Flask app to SQLite via a helper function:

get_db_connection() → connects to /instance/database.db.

Implemented a login route to validate users by email and password.

Secure session management with Flask's session and before_request for access control.

Supports both students and alumni via a combined SQL query (using UNION ALL).

```
# Login page (simple admin:admin)
@app.route('/login', methods=['GET', 'POST'])
def login():
    error = None
    if request.method == 'POST':
        email = request.form['username'].strip()
        password = request.form['password'].strip()

        conn = get_db_connection()
        cur = conn.cursor()

        # Find user in either Student or Alumni table
        cur.execute("""
            SELECT 'student' AS user_type, student_id AS id, first_name, email, password_hash
            FROM student WHERE email = ?
            UNION ALL
            SELECT 'alumni' AS user_type, alumni_id AS id, first_name, email, password_hash
            FROM alumni WHERE email = ?
        """, (email, email))
        user = cur.fetchone()
        conn.close()
```

Database Initialization, Reset, and Seeding Process

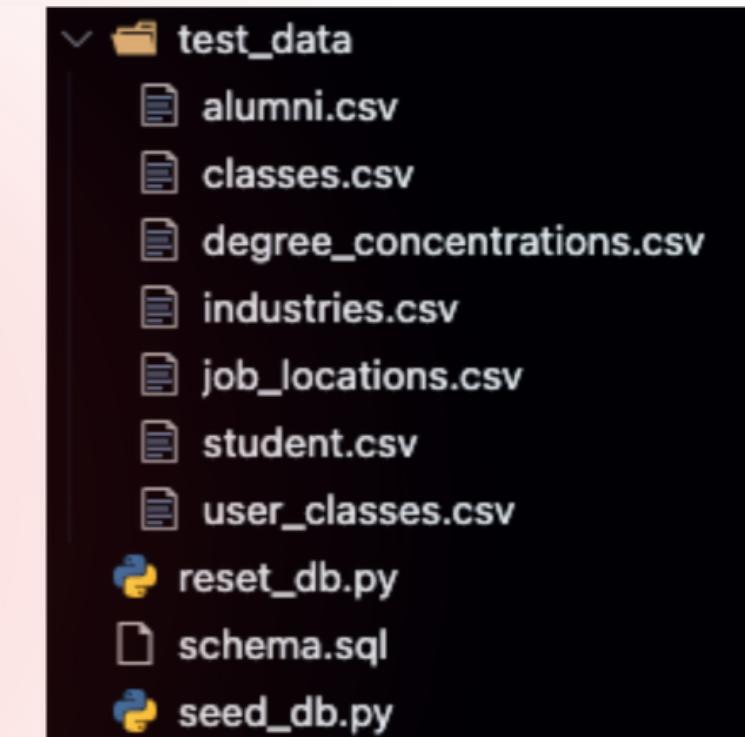
Created modular scripts:

- `init_db.py` / `reset_db.py` → drops and recreates all tables from `schema.sql`.
- `seed_db.py` → populates all tables with structured CSV test data.

Automated database creation for each development phase:

- Drops old database if exists.
- Executes `schema.sql` to build blank tables.
- Reads CSVs from `/db/test_data` to insert sample data.

Ensures reproducibility across development environments.



SQL										
< 1 / 1 > 1 - 10 of 10										
user_class_id	user_type	user_id	class_id	enrollment_date	completion_date	grade	status	created_at	updated_at	
1	student	1	1	9/1/25			In Progress	enrolled	10/22/25 0:00	10/22/25 0:00
2	student	1	4	9/1/25			In Progress	enrolled	10/22/25 0:00	10/22/25 0:00
3	student	2	5	1/15/25	5/15/25	A	completed	10/22/25 0:00	10/22/25 0:00	
4	student	3	7	8/30/25			In Progress	enrolled	10/22/25 0:00	10/22/25 0:00
5	student	4	8	9/2/25			In Progress	enrolled	10/22/25 0:00	10/22/25 0:00
6	student	5	9	8/25/24	12/10/24	A-	completed	10/22/25 0:00	10/22/25 0:00	
7	alumni	1	1	9/1/17	12/15/17	A	completed	10/22/25 0:00	10/22/25 0:00	
8	alumni	2	5	9/1/16	1/15/17	A-	completed	10/22/25 0:00	10/22/25 0:00	
9	alumni	3	8	9/1/18	1/15/19	B+	completed	10/22/25 0:00	10/22/25 0:00	
10	alumni	4	9	9/1/19	1/15/20	A	completed	10/22/25 0:00	10/22/25 0:00	

Repo & Documentation

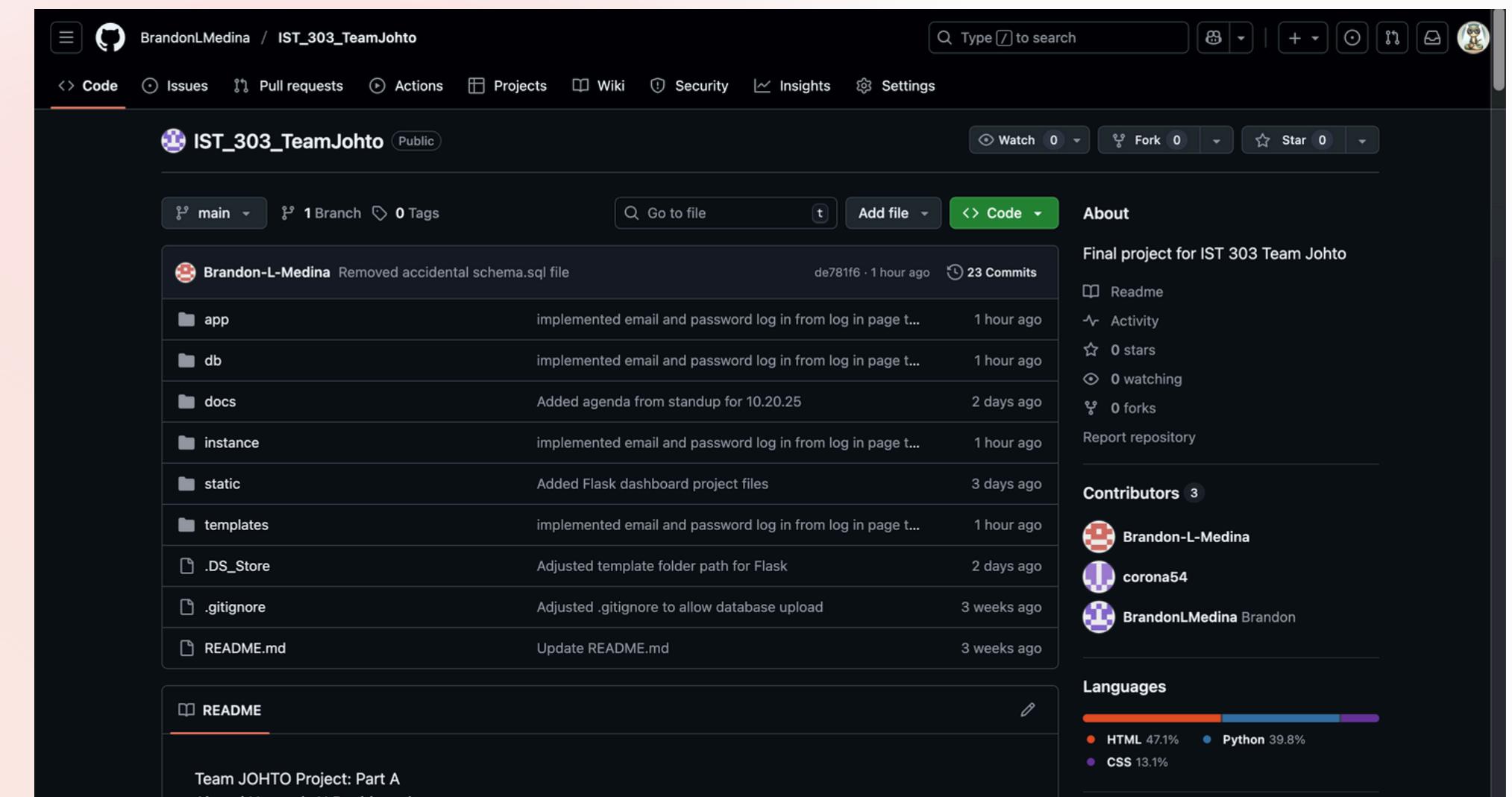
Folder Part C uploaded to GitHub

README updated with:

- Running Instructions
- Testing Notes
- User Story Mapping

GitHub Repo:

[https://github.com/BrandonLMedina/IST_303_TeamJohto]



Team Contributions

Member	Responsibility
Alex	Flask routing, dashboard features
Brandon	Database design, backend logic
Haoran	Login Design, Testing, debugging, documentation
Prajwal	Mentor matching logic, algorithms
Nihaad	UI/UX design, Agile (Jira setup)

What's Next for Milestone 2.0 ?



- Admin Analytics Dashboard (SDF-3)
 - AI Agent Integration (SDF-5, SDF-33)
 - Enhanced validation & error handling
 - Auto-generated outreach emails
- Jira Items: SDF-3, SDF-4, SDF-5, SDF-33

Mentor Matching ⓘ

Great! Based on your degree concentration in Data Science and industry in Tech, I have the following alumni mentors available for you:

1 Maria Lopez

- Match Score: 0.95
- Reason: Maria completed the same Data Science program and works in Tech Consulting in Los Angeles.

2 Priya Patel

- Match Score: 0.95
- Reason: Priya also holds a Data Science degree and is engaged in Tech Consulting in San Francisco.

These mentors have relevant degrees and work in the Tech industry. If you need more details or if you'd like to connect with them, let me know!

Summary & Reflections

Milestone 1.0 successfully deployed

Functional code + tested features

Agile approach well-integrated

Jira used for tracking, sprinting, and logging

Next: focus on analytics & scalability

"This project gave us hands-on experience with full-stack dev, testing, and teamwork."



Thank You