

Team Johto

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

Milestone 2.0 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 CGU Alumni Network Dashboard. At the top, a red header bar shows the text "CGU100" on the left and "About Dashboard Help" on the right. Below the header, a welcome message "Welcome back, Sofia!" is displayed. The main title "CGU Alumni Network Dashboard" is centered above a user profile section. The profile for "Sofia Nguyen" includes a placeholder profile picture, her name, email (email5@cgu.edu.invalid), phone number (909) 555-1005, degree (Masters — M.S. in Information Systems & Technology), years in program (3), and expected graduation year (2025). A "View Profile" button is located below the profile details. To the right of the profile is a "Preferences" section listing organization (Microsoft Corporation), city (Redmond), state (Washington), country (United States), and region (West Coast). Below the profile and preferences are three main dashboard modules: "Academic Progress" (a donut chart showing completed vs. planned classes), "Career Pathways" (listing industry, sub-industry, sector code, and description), and "Mentorship" (listing seeking mentor status and profile visibility). A "Job Opportunities" module with a "Find Job Opportunities" button is also present.

CGU100

About Dashboard Help

Welcome back, Sofia!

CGU Alumni Network Dashboard

Sofia Nguyen
Email: email5@cgu.edu.invalid
Phone: (909) 555-1005
Degree: Masters — M.S. in Information Systems & Technology
Years in Program: 3
Expected Graduation: 2025

View Profile

Academic Progress
Classes Completed vs Remaining

Preferences
Organization: Microsoft Corporation
City: Redmond
State: Washington
Country: United States
Region: West Coast

Career Pathways
Industry: Management Consulting
Sub-Industry: Technology Strategy
Sector Code: MC-007
Description: Advises organizations on IT strategy, innovation, and digital transformation.

Mentorship
Seeking Mentor: No
Profile Visibility: public

Job Opportunities
Find Job Opportunities

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 placeholder '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

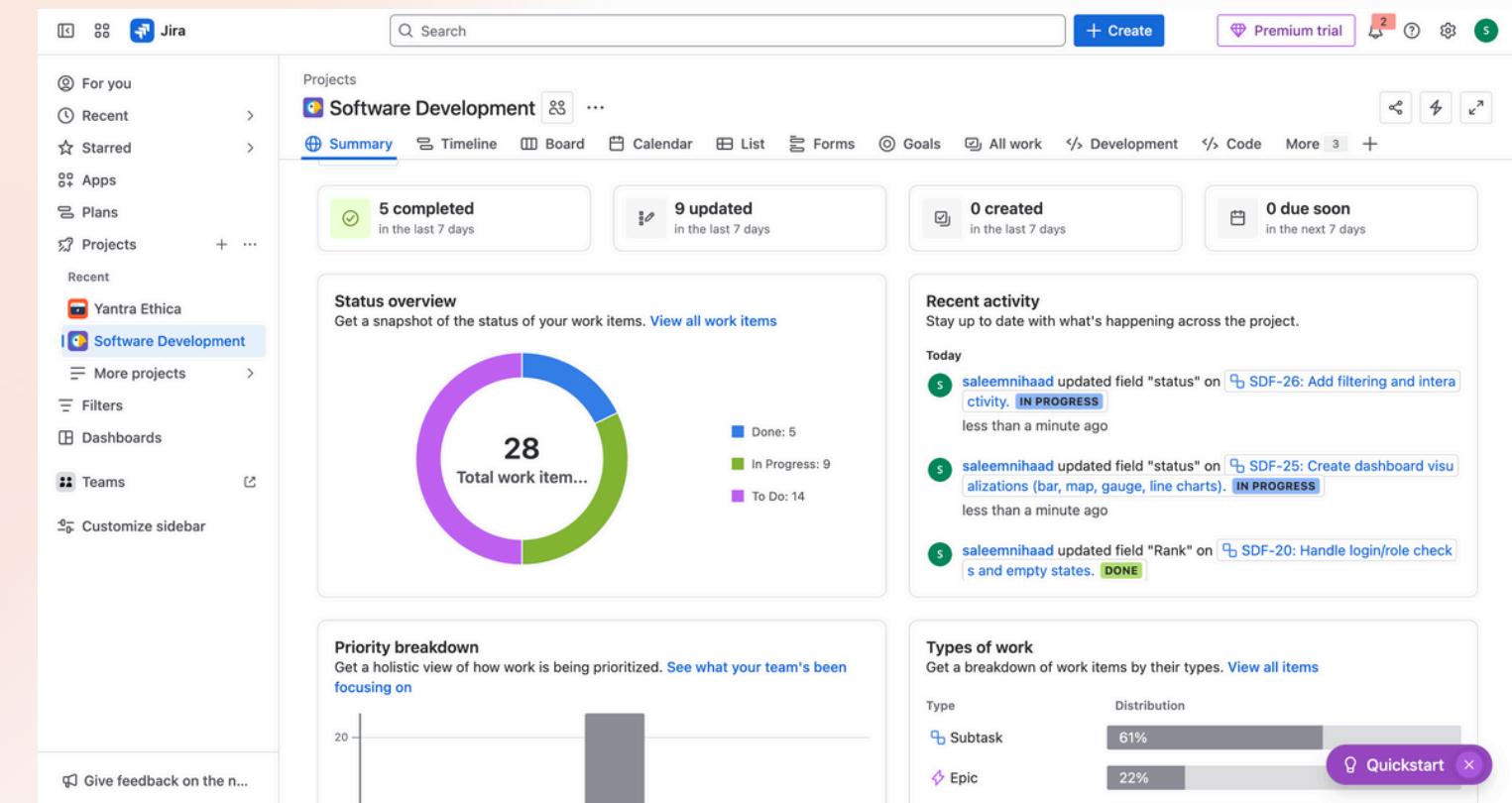
-Tool: Jira (Epics, Sprint Board, Subtasks)

-Burn down tracking: 28 work items total

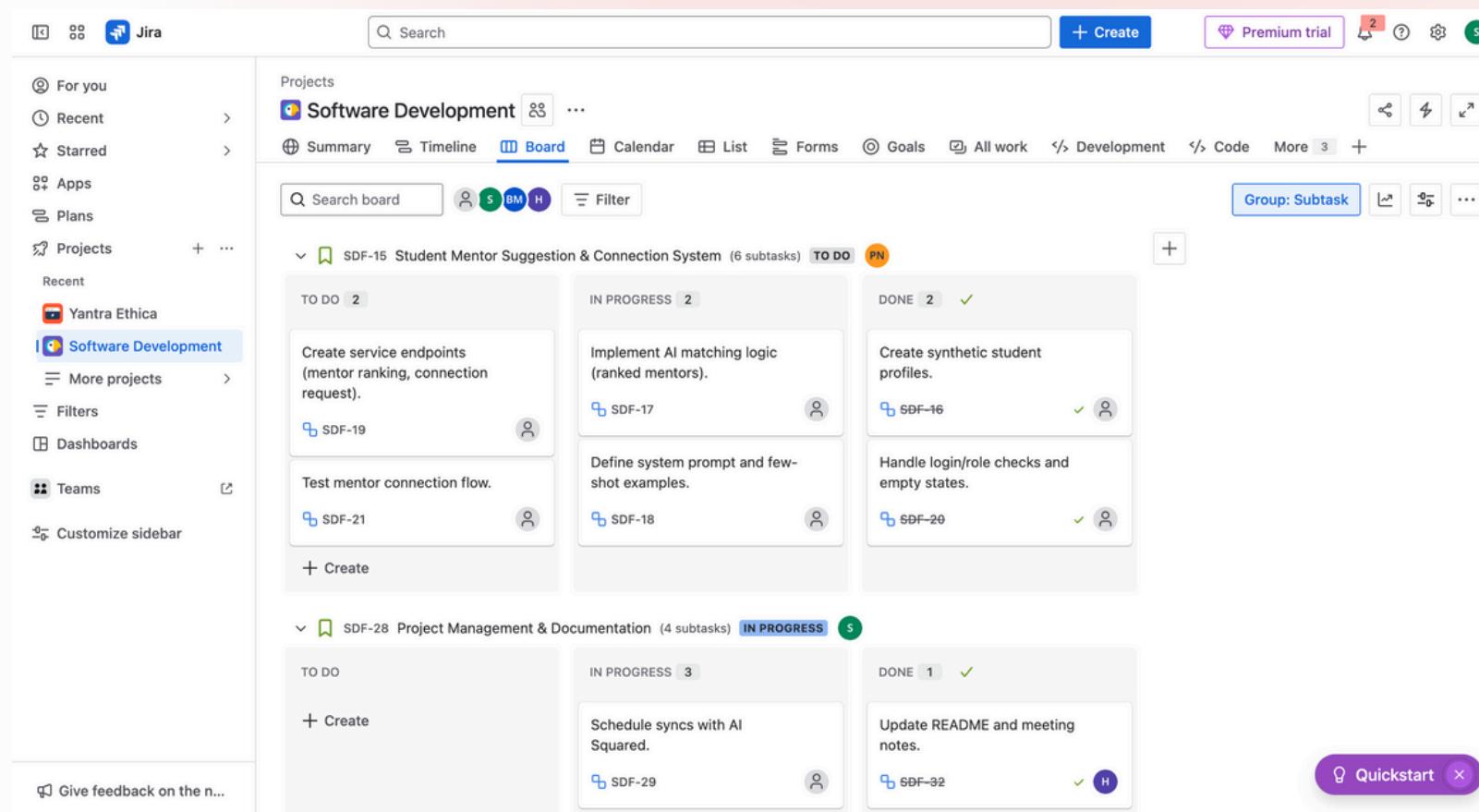
- o 14 To Do, 9 In Progress, 5 Done
- o

-Burndown Chart: Ideal vs Actual plotted clearly

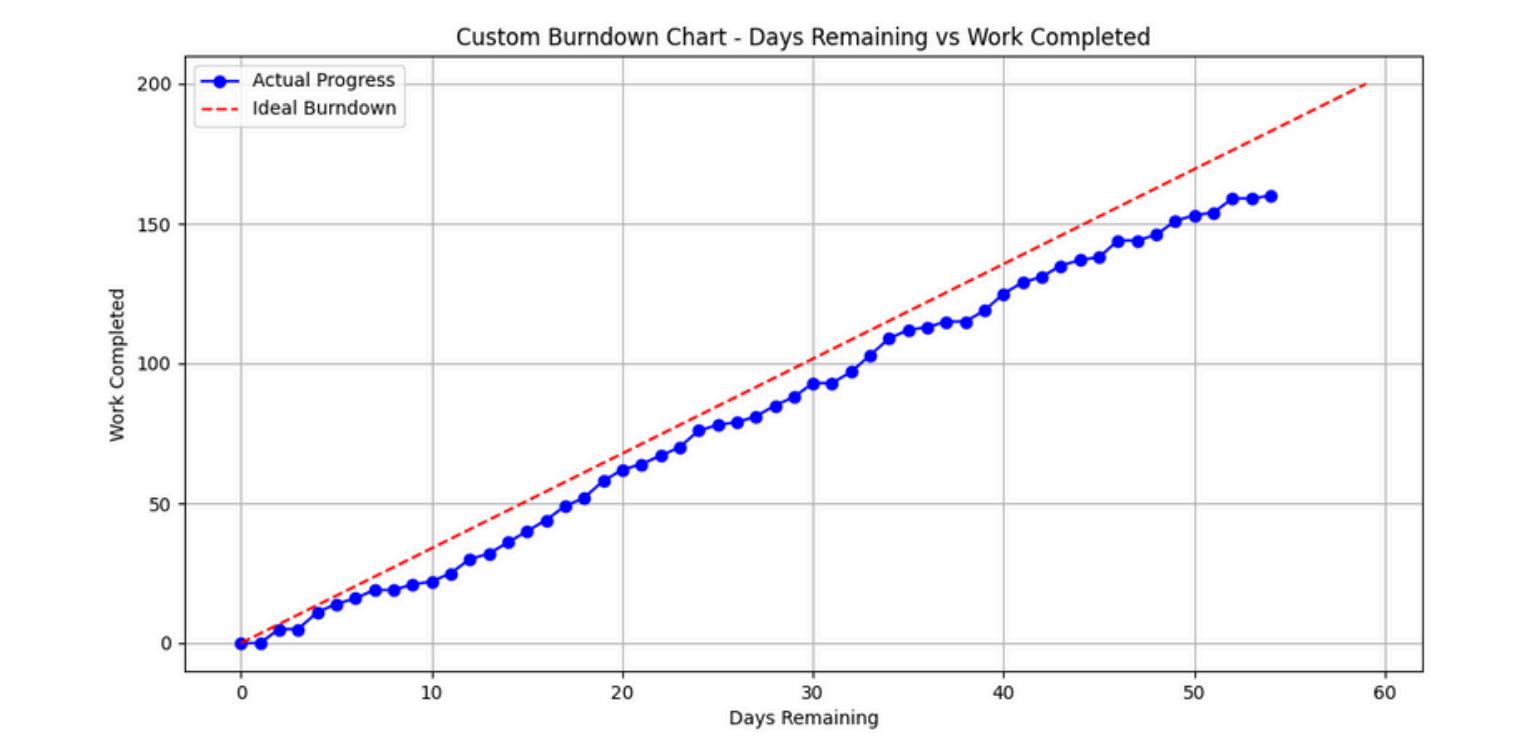
-Standups held 2x/week



The screenshot shows the Jira project summary dashboard for 'Software Development'. It features a central circular status overview showing 28 total work items, divided into 5 completed (green), 9 updated (blue), 0 created (grey), and 0 due soon (yellow). Below this are sections for recent activity, priority breakdown, and types of work. The sidebar on the left provides navigation links for various Jira features like epics, sprint boards, and subtasks.



The screenshot shows the Jira sprint board for the 'Software Development' project. It displays two sprints: 'SDF-15 Student Mentor Suggestion & Connection System' and 'SDF-28 Project Management & Documentation'. Each sprint has columns for 'TO DO', 'IN PROGRESS', and 'DONE'. Subtasks are listed under each column, with their current status (e.g., 'SDF-19 IN PROGRESS') indicated. The sidebar on the left shows the project structure and recent activity.



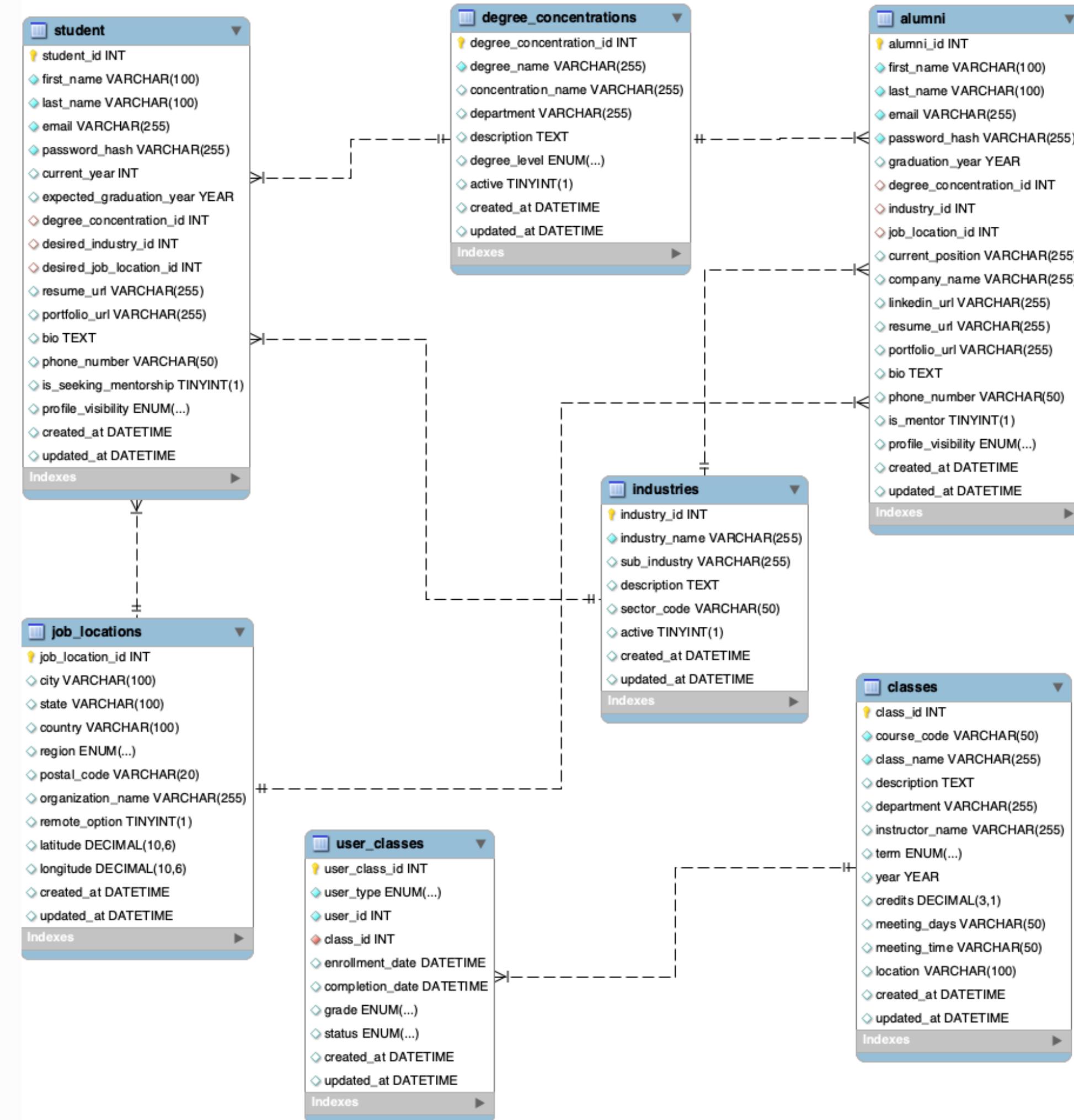
Database Design

Designed a normalized relational schema to manage users (now uses a single table), 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.



Welcome to CGU Alumni Network

Log in to continue

Email

Password



Log In

Register

Don't have an account? You can register above.

Login Page

login → DB verify → save to session
→ redirect to dashboard → query by user_id
→ personalized page

```
194 # -----
195 # LOGIN PROTECTION
196 #
197 @app.before_request
198 def require_login():
199     allowed_routes = ['index', 'register', 'login', 'static']
200     if request.endpoint not in allowed_routes and not session.get('logged_in'):
201         return redirect(url_for('login'))
202
203 #
204 # INDEX (HOMEPAGE)
205 #
206 @app.route('/')
207 def index():
208     return render_template('index.html')
209
210 #
211 # LOGIN - unified users table
212 #
213 @app.route('/login', methods=['GET', 'POST'])
214 def login():
215     error = None
216
217     if request.method == 'POST':
218         email = request.form['username'].strip()
219         password = request.form['password'].strip()
220
221         conn = get_db_connection()
222         cur = conn.cursor()
223
224         cur.execute("""
225             SELECT user_id, first_name, email, password_hash, user_type
226             FROM users
227             WHERE email = ?
228             """, (email,))
229         user = cur.fetchone()
230         conn.close()
231
232
233         if not user:
234             error = "Email not found."
235         elif password != user['password_hash']: # (Later upgrade to hashing)
236             error = "Invalid password."
237         else:
238             session['logged_in'] = True
239             session['user_id'] = user['user_id']
240             session['user_type'] = user['user_type']
241             session['email'] = user['email']
242             flash(f"Welcome back, {user['first_name']}!")
243             return redirect(url_for('dashboard'))
244
245     return render_template('login.html', error=error)
246
```

Welcome/Login → write session → redirect to /dashboard → /dashboard uses the session's user_id to join tables and query the database → render the returned fields into dashboard.html.

```
@app.route('/dashboard')
def dashboard():
    if not session.get('logged_in'):
        return redirect(url_for('login'))

    user_id = session['user_id']
    user_type = session['user_type']

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

    # IMPORTANT:
    # Students get desired_* lookups
    # Mentors get actual job_location_id & industry_id lookups
    cur.execute("""
        SELECT
            u.user_id AS id,
            u.user_type,
            u.first_name,
            u.last_name,
            u.email,
            u.phone_number,
            u.current_year,
            u.expected_graduation_year,
            u.graduation_year,
            u.current_position,
            u.company_name,
            u.profile_visibility,
            u.is_seeking_mentorship,
            u.is_mentor,
            dc.degree_level,
            dc.degree_name,
            dc.concentration_name
        """)
```

Welcome to CGU Alumni Network!

Logged in as: `email6@cgu.edu.invalid`

[Go to Dashboard →](#)

About the Platform

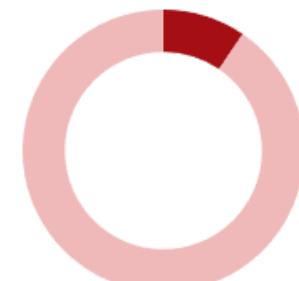
The CGU Alumni Network helps students and alumni connect, share career insights, and grow professionally. Explore alumni profiles, update your information, and interact with the community.

Welcome back, Alex!

CGU Alumni Network Dashboard

**Alex Morgan****Email:** email6@cgu.edu.invalid**Phone:** (909) 555-1006**Degree:** Masters — M.S. in Information Systems & Technology**Current Position:** Senior Data Scientist**Company:** Alphabet Inc.[View Profile](#)

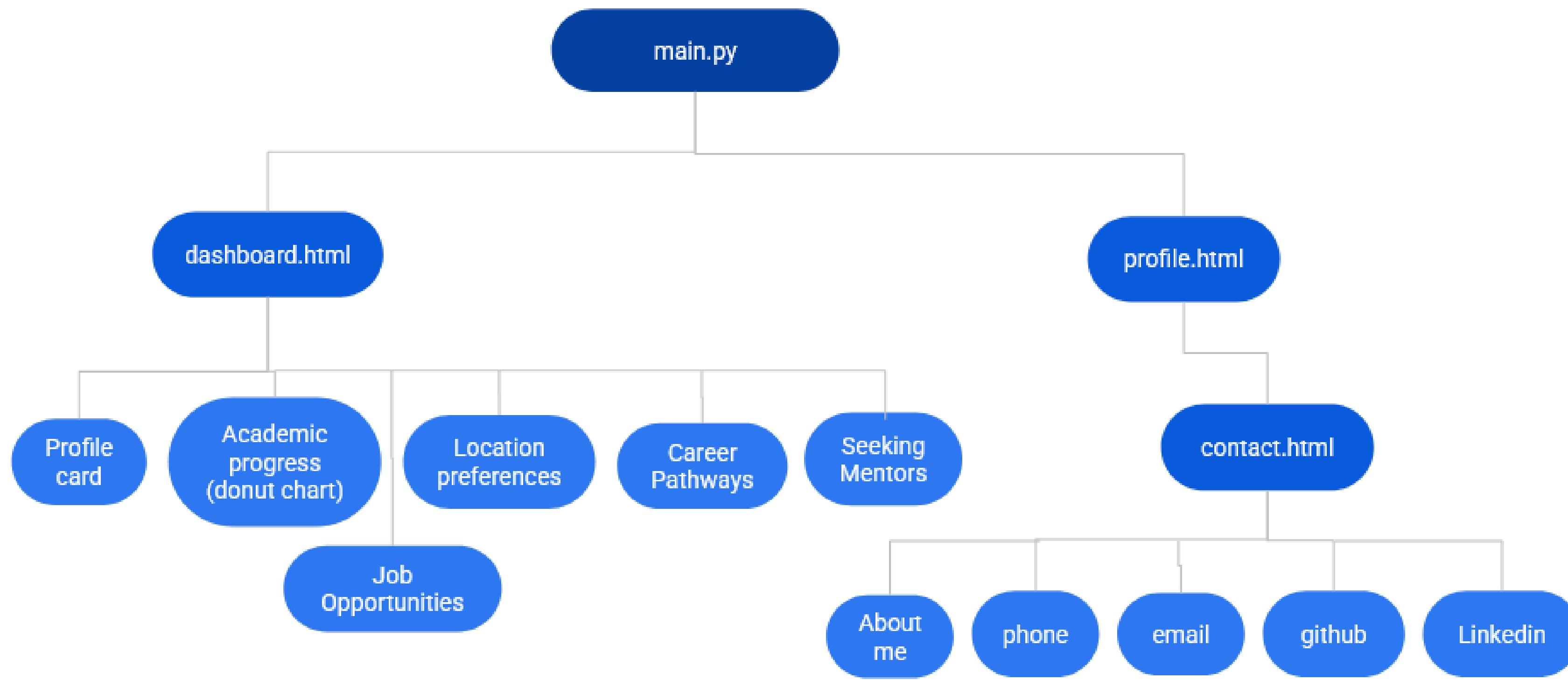
Academic Progress



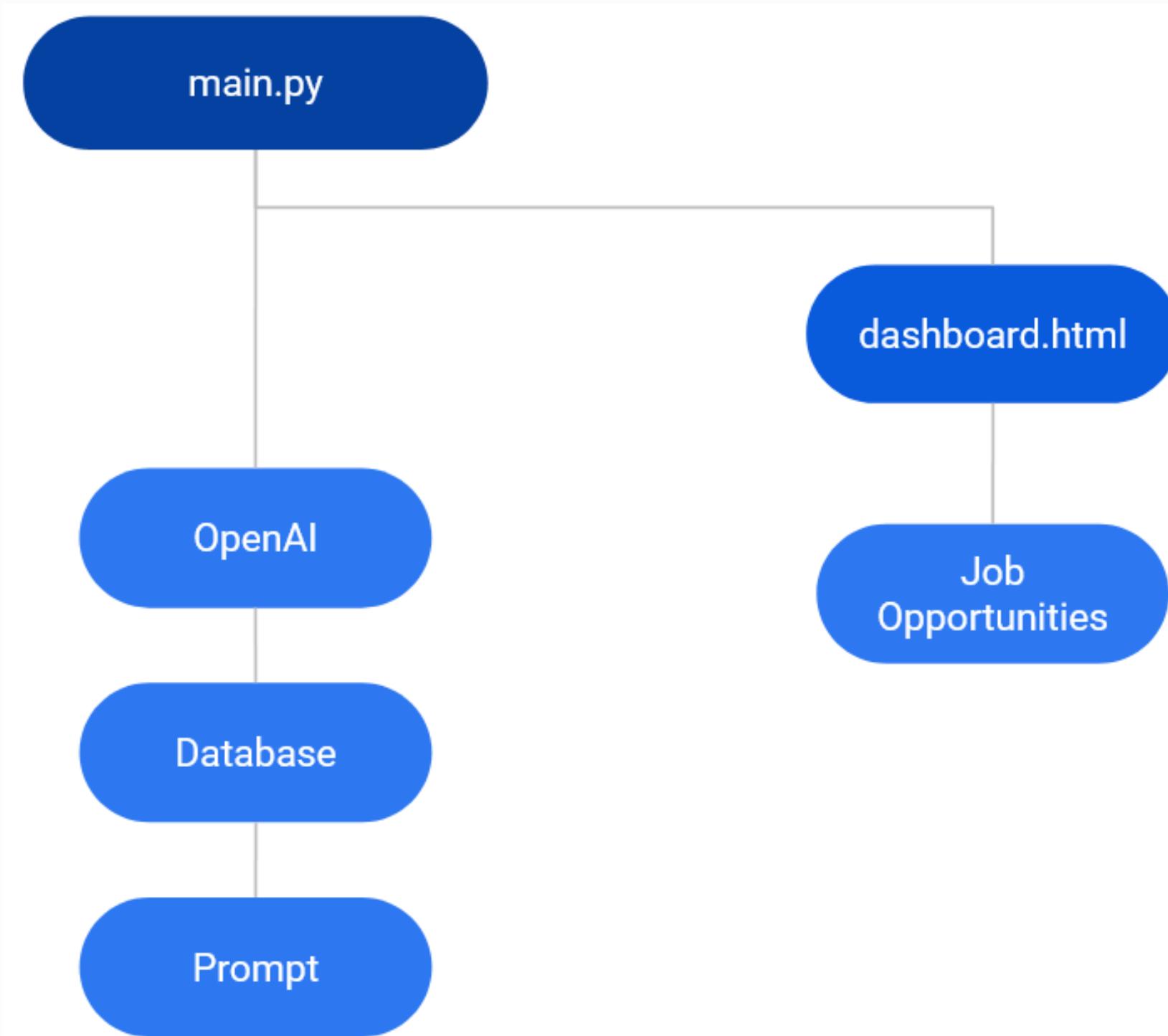
Preferences

Organization: Alphabet Inc. (Google)**City:** Mountain View**State:** California**Country:** United States**Region:** West Coast

Dashboard - code



Job Opportunities – code



Mentor Matching agent

Assistant ◇

✖

File Searched files

"Information Systems major IST industry"

"IST industry"

"Information Systems major"

Matched Mentors:

- Name: Alex Morgan Major: M.S. in Information Systems & Technology (Data Science & Analytics) Industry: Data Science & Analytics Job Title: Senior Data Scientist Location: Alphabet Inc., Mountain View, California Bio: Leads product ML teams building scalable recommendation systems; focuses on fairness and interpretability.
- Name: Priya Shah Major: M.S. in Information Systems & Technology (Cybersecurity & Networking) Industry: Cybersecurity Job Title: Director, Enterprise Security Location: Microsoft Corporation, Redmond, Washington Bio: Heads global security operations and incident response; advisor on cloud security architecture.

▼

⌚ 8.8s ↑ 14.9kt ↓ 302t

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.

```
# 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()
```

Dynamic Dashboard Data

Dashboard Flow

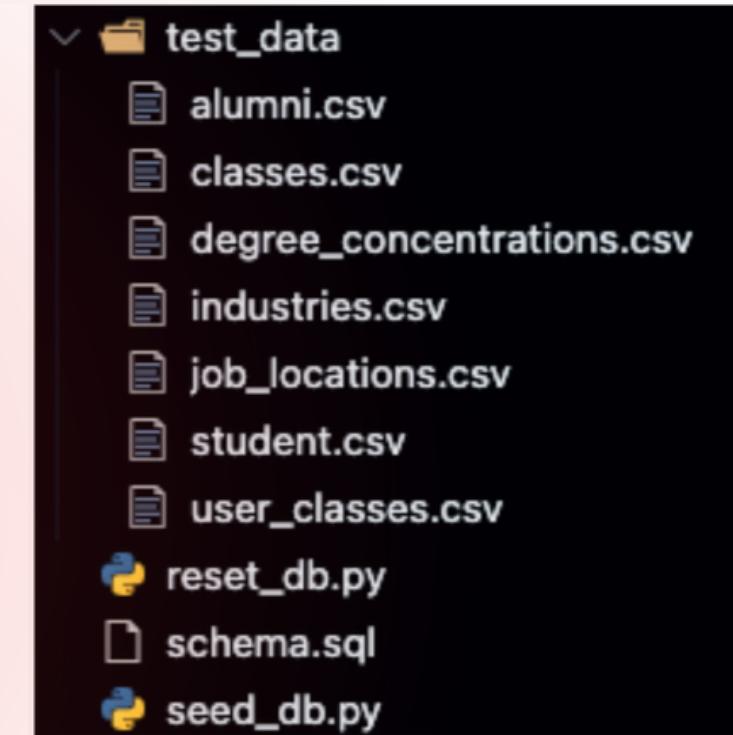
- Once the session is validated with username/password the helper function stores user type (Mentor or Student) and UserID

Database Query

- The main.py runs a SQL query that LEFT JOINs the user record with Degree Concentrations, Industries, and Job Location tables

Display Logic

- After fetching the data the main.py stores the data in a dict and passes the data to the dashboard for display and dynamic updating



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	

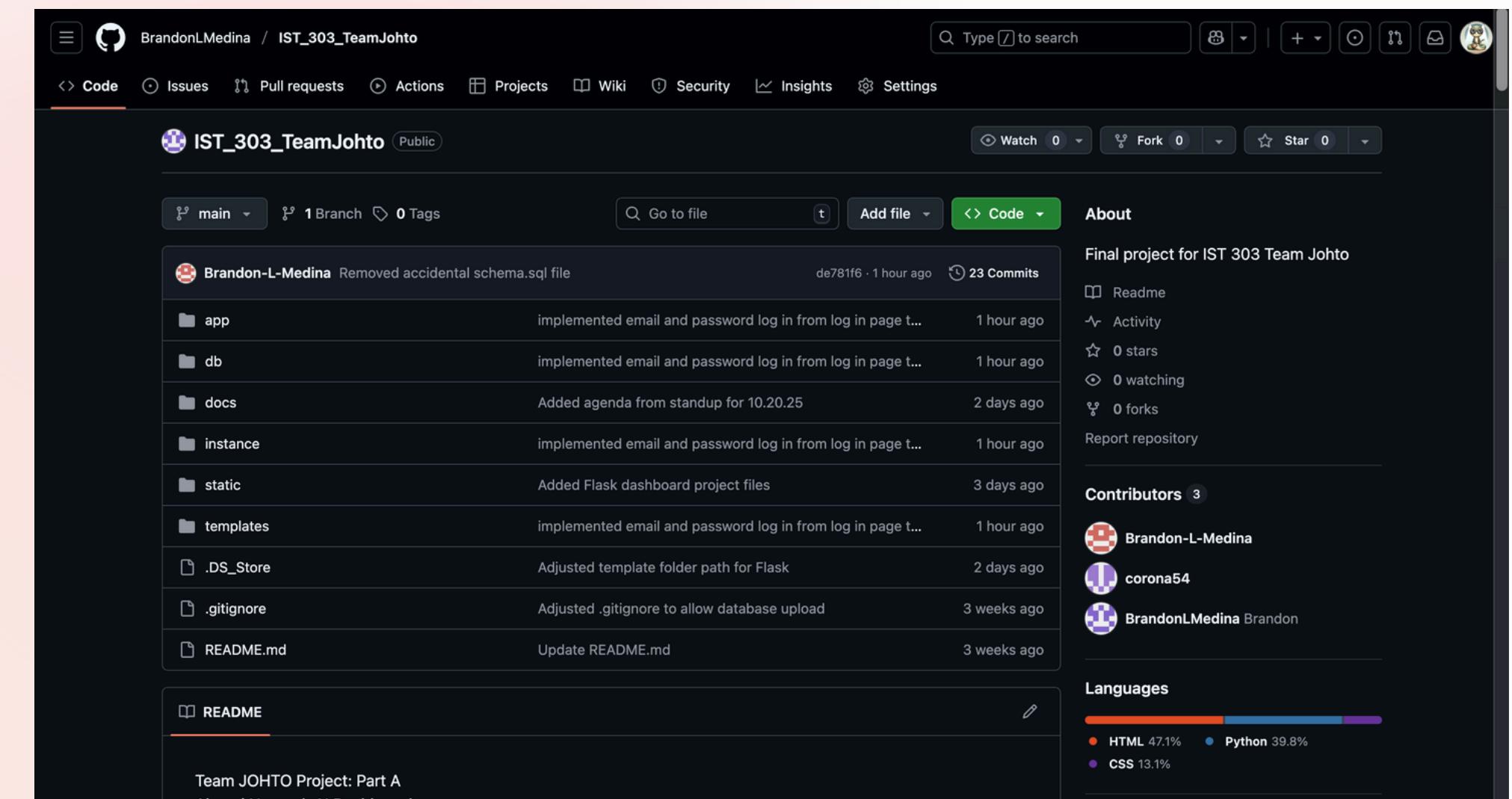
Repo & Documentation

Folder Part D uploaded to GitHub

README updated with:

- Running Instructions
- Testing Notes
- User Story Mapping

GitHub Repo: [[https://github.com/
BrandonLMedina/
IST_303_TeamJohto](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)

Summary & Reflections

Milestone 2.0 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