

1. Project Overview & Goals

Develop a web-based system to manage the lifecycle of capstone/senior design projects, including:

- Sponsor project submission & profile management
- Student preference ranking
- Instructor/admin project assignment
- File/image/video attachments on project profiles
- Basic viewing/listing interfaces for different roles

Core User Roles & Features

- **Sponsors:** Submit/edit projects, upload attachments (text, images, videos, files)
- **Students:** View available projects, submit ranked preferences
- **Instructor/Admin (Dr. Anderson):** View student rankings, manually assign students to projects, manage overall records
- **System:** Email notifications, secure file handling

2. Technology Stack (Decided)

- **Backend:** Django (REST API)
- **Frontend:** Vue.js (plain/composition API preferred over Nuxt.js for simpler learning curve)
- **Database:** PostgreSQL (recommended for binary/large file storage via TOAST compression)
- **Deployment/Testing Environment:** Docker containers (primary approach)
 - Start with local/containerized setup → VM/container snapshots for testing & backups
 - Cloud hosting (Google Cloud or similar) considered for later production
- **Authentication:** Plan to implement OAuth (investigate UA/Bama migration path)

3. Repository & Hosting

- **GitHub:** New project-specific repository to be created by **Jackson**

- Enables clean separation and potential use of GitHub Pages (if needed for static content)
- Avoid using personal accounts for the main project

4. Database Schema Outline (High-Level)

- **Project** (1 record per project)
 - ID (PK)
 - Project_Name
 - Project_Description
 - Sponsor_ID (FK)
 - Project_Website (nullable)
 - Year_Began (int)
 - Status (enum)
- **Sponsor**
 - ID (PK)
 - Name
 - Email
 - Phone_Number (nullable)
 - Organization (nullable)
- **Student**
 - ID (PK)
 - Name
 - Email (future: integrate/match with UA database)
- **Assignment** (junction table – person ↔ project)
 - ID (PK)
 - Project_ID (FK)
 - Person_ID (FK)
 - Person_Type (enum: sponsor/student)
 - Semester (enum)
 - Year (int)
- **Preference** (student rankings)
 - ID (PK)
 - Student_ID (FK)
 - Project_ID (FK)
 - Rank_Value (int)
- **Attachments**

- ID (PK)
- Project_ID (FK)
- File_URL (or direct storage reference)

5. Key Processes & Workflows

1. Sponsor submits project → record created
2. Students view projects → submit ranked preferences
3. Instructor views aggregated rankings → assigns students to projects → final approval
4. File uploads supported on project profiles (images/videos/documents)
5. System sends email notifications (future phase)

Multi-Semester Handling

- Treat each semester as a separate logical project/assignment cycle
- Copy forward relevant project records when continuing
- No full re-submission required from sponsors (renewal path exists)

6. Development & Collaboration Tools

- **Scrum / Progress Tracking**
 - Blackboard group discussion threads (one per team)
 - Dr. Anderson will create the group discussion areas
 - Each team member posts updates in their team's thread (Scrum journal logs)

7. Infrastructure & Resources

- **Initial Environment:** Docker-based local/dev setup (fastest way to start)
- **Testing/Hosting:** University VM requested for early testing
 - Fallback/parallel: Google Cloud (or equivalent)
- **Backup Strategy:** Use VM/container snapshots for database & application state
- **File Storage:** Leverage PostgreSQL large object/TOAST support initially

8. Open / Future Decisions

- Exact file storage strategy (database vs. object storage bucket)

- Full OAuth flow & UA directory integration timeline
- Email notification implementation details & provider
- Production cloud deployment plan & timeline
- Any additional admin dashboards/reports