**Book Bridge**

**ISM 6225: APPLICATION DEVELOPMENT FOR ANALYTICS**

**GROUP-5** : Priyanka Jammu, Hriday Reddy Purma, Gopi Dodda, Umeaiman Merchant

**Deployed Application URL:**  <https://bookbridgeapp-ccehg2bhe0cvhaat.canadacentral-01.azurewebsites.net/>

**ENTITY RELATIONSHIP DIAGRAM:**

A screenshot of a computer

AI-generated content may be incorrect.

**GITHUB LOGS:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**SELF-REFLECTION DOCUMENT:**

**GOPI DODDA**

My contribution to the project centered on data handling, storage strategy, and infrastructure planning. My core responsibilities included:

* Implemented a singleton service to manage persistent in-memory storage for book records throughout the app’s lifecycle.
* Designed the final Entity-Relationship Diagram (ERD) that formed the backbone of our data model and ensured consistency across components.
* Provided technical assistance and logical flow structuring for the About Us page, particularly the integration of team roles and the data model visualization.
* Helped ensure that the application’s data architecture could support a future move to full-stack functionality.

**Challenges Faced and How They Were Addressed:**

* **Structuring for Scalability:** A major challenge was designing a scalable data layer using only frontend logic. This was addressed by introducing singleton-based JavaScript modules to simulate persistent services.
* **Data Model Alignment:** Ensuring our logical data model reflected all relationships correctly, especially user-to-book and book-to-borrow\_records, required several iterations and team discussions.

**Key Learnings:**

* Improved understanding of designing persistent storage logic on the frontend
* Learned to visualize and document database schema effectively
* Strengthened collaborative problem-solving and cross-role coordination

**Estimated Time Spent:**

* Data modeling and diagramming: 4 hours
* Singleton service setup and integration: 6 hours
* About page structure and data logic: 3 hours
* Team coordination and revisions: 3 hours
* Total: ~16 hours

**UMEAIMAN MERCHANT**

I led backend development for the application, focusing on core logic, structure, and user flow. My main tasks involved:

* Established an MVC (Model-View-Controller) architecture to organize our codebase cleanly and modularly.
* Implemented full CRUD operations, enabling users to add, read, update, and delete book entries using structured form flows.
* Ensured robust data validation to prevent incomplete or malformed submissions.
* Simulated backend logic and user-based control to mirror real-world application behavior.

**Challenges Faced and How They Were Addressed:**

* **Integrating MVC into a Static Stack:** It was tricky to simulate MVC principles using only frontend tools. I addressed this by mimicking controllers and services in JavaScript and using modular code design.
* **Maintaining Clean Logic:** Avoiding redundant code was difficult during rapid development. I solved this by extracting reusable functions and applying template-like components.

**Key Learnings:**

* Deepened knowledge of MVC structure and its implementation in a mock environment
* Improved skills in JavaScript-based validation and routing simulation
* Developed a clear understanding of the importance of separation of concerns

**Estimated Time Spent:**

* MVC setup and flow design: 6 hours
* CRUD logic implementation: 6 hours
* Validation and control logic: 3 hours
* Testing and adjustments: 2 hours
* Total: ~17 hours

**PRIYANKA JAMMU**

I led the frontend design and styling of the application, ensuring usability, responsiveness, and aesthetic consistency. My contributions included:

* Designed and styled key pages like Home, Read, and Create, focusing on layout balance and clarity.
* Created a responsive interface using Flexbox and CSS Grid, optimized for both desktop and mobile views.
* Integrated data from API responses into the frontend forms, enhancing the interactivity and usability of the Create Book feature.
* Developed JavaScript interactions to enhance user experience with dynamic content and dropdowns.

**Challenges Faced and How They Were Addressed:**

* **Ensuring Pixel-Perfect Layout:** Creating a visually consistent layout across pages and screen sizes required many iterations. This was addressed through rigorous testing and refinement.
* **Integrating Dynamic Data:** Incorporating Google Books API results into styled forms was complex. I overcame this by matching API field mappings to form fields and applying fallback styles.

**Key Learnings:**

* Enhanced proficiency in responsive CSS and modern layout systems
* Learned how to manage dynamic content within static HTML environments
* Strengthened understanding of UX design principles

**Estimated Time Spent:**

* UI/UX design and development: 7 hours
* Responsive layout implementation: 4 hours
* API and JS integration: 3 hours
* Styling polish and testing: 2 hours
* Total: ~16 hours

**HRIDAY REDDY PURMA**

I was responsible for deploying the final web application and ensuring its accessibility online. My key contributions included:

* Deployed the full application to Microsoft Azure, managing hosting setup and version compatibility.
* Resolved deployment issues, including static file pathing, CORS errors, and asset loading.
* Managed our GitHub project repository, resolving merge conflicts and keeping the project up to date.
* Designed and compiled the final presentation materials, ensuring all key features were communicated clearly.

**Challenges Faced and How They Were Addressed:**

* **Azure Configuration Complexity:** Deploying a multi-page static site with JS dependencies on Azure required adjusting configuration files. I resolved this through iterative testing and documentation checks.
* **Version Control Coordination:** Synchronizing final changes with ongoing updates from teammates was challenging. I established a clear Git merge protocol and time windows for pushing changes.

**Key Learnings:**

* Gained hands-on experience deploying static websites using Azure
* Improved GitHub workflow management and conflict resolution
* Learned to translate technical work into visual presentations effectively

**Estimated Time Spent:**

* Azure deployment and testing: 6 hours
* GitHub merge management: 3 hours
* Presentation creation and organization: 3 hours
* Issue resolution and debugging: 3 hours
* Total: ~15 hours