ISSUESTACK

Internal Support Ticket Management System

Assisted by: Prof. Gabriella Mosquera

Presented by: Group 15

OUR TEAM

Name	Role	Email
Gowri Prashanth Kanagaraj	MERN Stack Developer	gowri.kanagaraj@dal.ca
Kuldeep Rajeshbhai Gajera	Designer and Copywriter	kl210309@dal.ca
Darshit Mukeshbhai Dhameliya	Designer and Integrator	dr773233@dal.ca
Nisarg Sharadkumar Vaghela	Database Manager	nvaghela@dal.ca
Dhruvik Maheshbhai Kakadiya	Documents manager	dhruvik.kakadiya@dal.ca
Bhautik Rameshbhai Koshiya	Tester	bh889463@dal.ca

INTRODUCTION

IssueStack is a web app, transforming how we handle internal support requests. It's a one-stop shop for employees to ask for help with IT, HR, admin, or payroll issues. With IssueStack, communication is smoother, problems are solved faster, and productivity gets a boost.
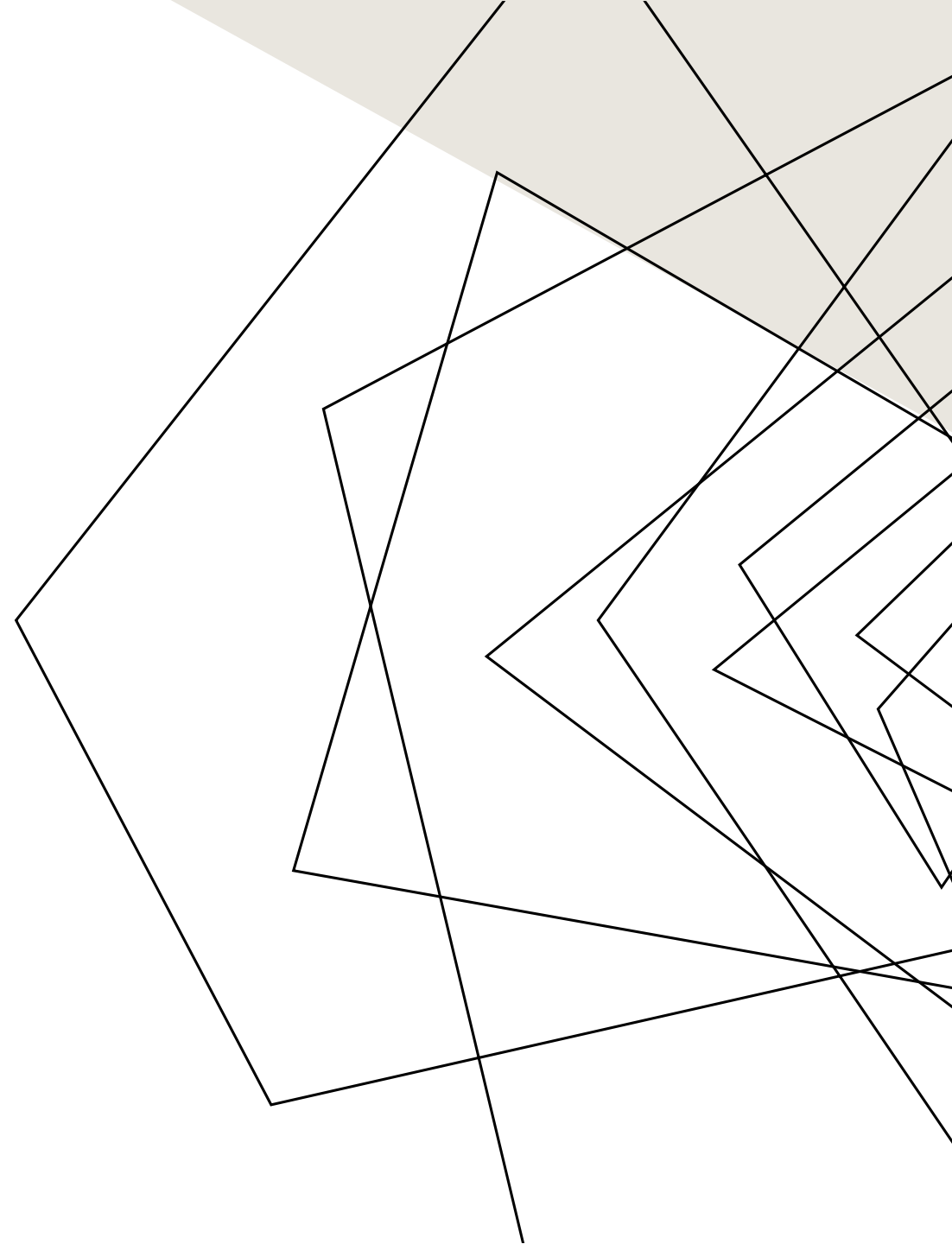
PROJECT SCOPE/GOALS

The goals of IssueStack are as follows:

- **Efficient Ticket Handling:** Streamline the process of raising, assigning, and resolving support tickets to ensure quick and effective problem resolution.

- **Improved Communication:** Facilitate clear and transparent communication between employees and support departments, keeping everyone informed about the status and progress of their requests.

- **Tracking and Analytics:** Provide tools for tracking and analyzing support ticket data to identify trends, bottlenecks, and areas for improvement in the support process.

- **User-Friendly Interface:** Design an intuitive and user-friendly interface that allows employees to easily submit and track their support tickets without extensive training.

PROJECT FEATURES

Must Have Features

1. **Authentication**: Secure user access via both user interface and API integration.

2. **Profile Management/Settings**: Users can manage profiles, passwords, and personal info for a seamless experience.

3. **Manage the Board**: Administrators can flexibly add or remove teams.

4. **Team Assignment/ Team Assignment**: Efficiently assign or remove team members for support distribution.

5. **Raise Ticket**: Intuitive interface for employees to submit support tickets.

6. **Ticket Assignment**: Automatically route tickets to appropriate departments.

7. **Ticket Management**: Prioritize, forward, and update ticket statuses.

8. **Comments on Ticket**: Users and staff can add comments, aiding communication and record-keeping.

9. **Notifications**: Keep users updated on ticket progress via email and in-app notifications.

PROJECT FEATURES

Nice To Have Features

1. **Escalation:** Implement an escalation process where critical issues are automatically assigned to team leads, ensuring prompt attention to high-priority matters.

2. **Knowledge Base:** Create a user-friendly knowledge base with department-specific FAQs to empower users with self-help resources, reducing the volume of support tickets.

3. **Reporting and Analytics:** Implement a visual analytics dashboard displaying charts of ticket distribution by department, offering insights for informed decision-making and process optimization.

APPLICATION DETAILS

IssueStack's brand essence centers on efficiency, transparency, and user empowerment in internal support ticket management. It embodies reliability, professionalism, and user-centricity, reflected in its design prioritizing ease of use and clear communication. Targeting all employees needing support across departments like IT, HR, Admin, and Payroll, IssueStack stands out from competitors like Zendesk and Service-now with its streamlined interface and department-based ticket categorization for efficient resolution. This strategic positioning aims to inspire trust and confidence in users, fostering satisfaction and productivity within the organization.

APPLICATION DETAILS

Technology used:

Our application follows a traditional MERN stack with MVC architecture. MERN is a full stack, following the three-tier architectural pattern, including the front-end view tier (React.js), Business Logic tier (Express.js and Node.js), and database tier (MongoDB).

M   E   R   N
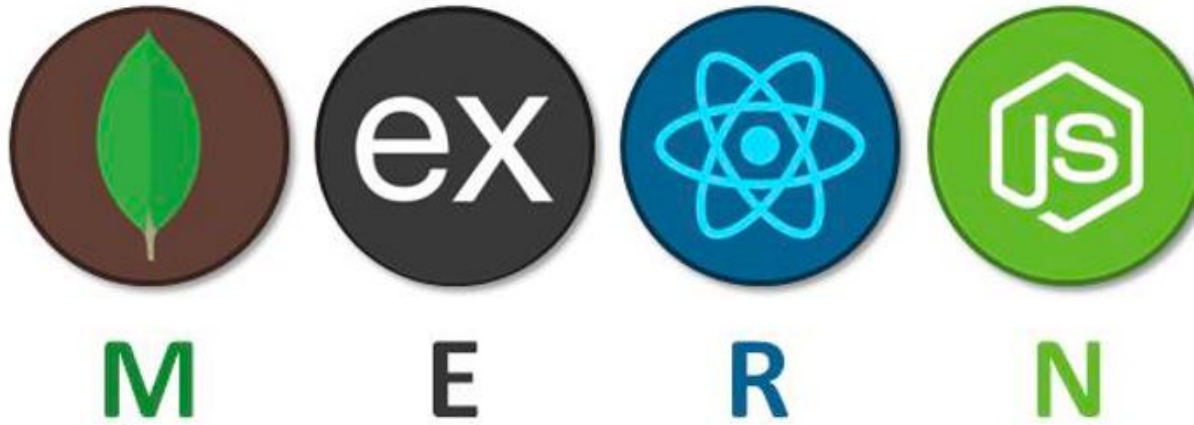
APPLICATION DETAILS
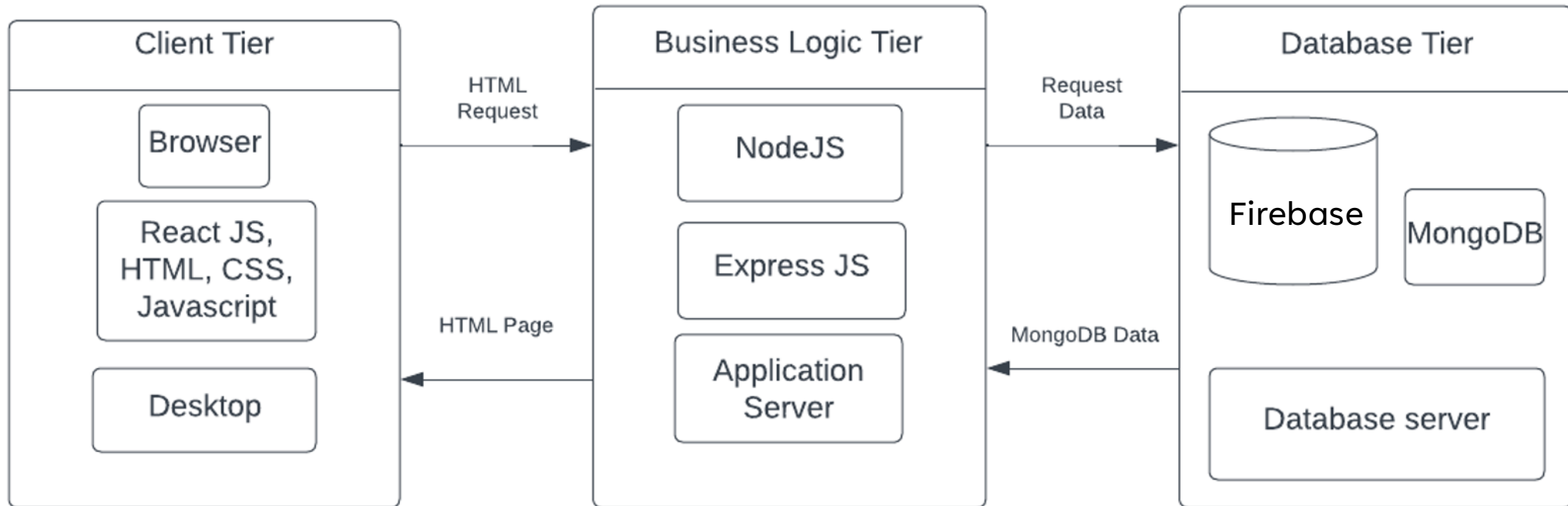
Figure 1: Application Architecture

APPLICATION DETAILS - ARCHITECTURE

View (User Interface):

- Developed using JavaScript, HTML, and CSS.

- ReactJS framework utilized for development.

- Interface through which users interact with the application's features.

Controller (Business Logic Tier):

- Implemented using Node.js and Express.js.

- Acts as the Application Server.

- Handles HTTP requests from users and responds appropriately.

- Provides API endpoints for external integration and interaction with the application's functionalities.

Model (Database Tier):

- Powered by MongoDB.
- Stores all essential data required for effective operation of the application.
- Firebase was used to store all Ticket attachments.

APPLICATION DETAILS

Backend Dependencies:

Dependency Name	Version	Description
bcrypt	^5.1.1	Password hashing library
cors	^2.8.5	Enables Cross-Origin Resource Sharing (CORS)
dotenv	^16.4.5	Loads environment variables from a .env file
express	~4.16.1	Web framework for building web applications
http-errors	~1.6.3	Provides HTTP error handling middleware
http-status-codes	^2.3.0	HTTP status code reference
jsonwebtoken	^9.0.2	Library for creating and verifying JSON Web Tokens (JWT)
mongodb	^6.5.0	Driver for interacting with MongoDB databases
mongoose	^8.2.3	Object Document Mapper (ODM) for MongoDB in Node.js
nodemailer	^6.9.13	Library for sending emails
nodemon	^3.1.0	Utility for watching and restarting Node.js applications

APPLICATION DETAILS

Frontend Dependencies:

Dependency Name	Version	Description
@ant-design/icons	^4.8.1	Icon library for Ant Design React UI framework.
@dnd-kit/sortable	^8.0.0	Drag-and-drop library for creating sortable elements.
@testing-library/jest-dom	^5.14.1	Utilities for testing React components with Jest and DOM manipulation.
@testing-library/react	^11.2.7	Testing library for writing React component tests.
@testing-library/user-event	^12.8.3	Simulates user interactions like clicks, typing, and hovering for React testing.
antd	^4.16.6	Ant Design React UI framework for building complex web interfaces.
apexcharts	^3.27.1	Interactive charting library for web applications.
axios	^1.6.8	Promise-based HTTP client for making API requests.
bcrypt	^5.1.1	Password hashing library for Node.js.
express	^4.19.2	Web framework for building web applications and APIs in Node.js.
firebase	^10.10.0	Google Firebase client libraries for web and mobile app development.
install	^0.13.0	Utility for installing local npm packages.

APPLICATION DETAILS

Frontend Dependencies:

Dependency Name	Version	Description
jsonwebtoken	^9.0.2	Library for creating and verifying JSON Web Tokens (JWT) for authentication.
moment	^2.30.1	Library for working with dates and times.
mongoose	^8.2.3	Object Document Mapper (ODM) for MongoDB in Node.js.
nodemailer	^6.9.13	Library for sending emails from Node.js applications.
react	^17.0.2	JavaScript library for building user interfaces.
react-apexcharts	^1.3.9	React component for integrating ApexCharts library.
react-dom	^17.0.2	DOM manipulation library for React applications.
react-nice-avatar	^1.5.0	React component for generating nice-looking avatars.
react-router-dom	^5.2.0	Routing library for React applications.
react-scripts	^5.0.1	Collection of tools and configurations for creating new React projects.
react-type-animation	^3.2.0	React component for creating animated typing effects.
styled-components	^5.3.0	CSS-in-JS library for styling React components.
web-vitals	^1.1.2	Library for measuring core web vitals of a web application.

PROTOTYPE

Figure 2: Prototype of sign in

ISSUESTACK

🔔 Alex

Notification 1	✖
Notification 2	✖
Notification 3	✖
Notification 4	✖

Figure 3: Prototype of in-app notification

Figure 4: Prototype of employee profile

Figure 5: Prototype of raise a ticket

Figure 6: Prototype of ticket details
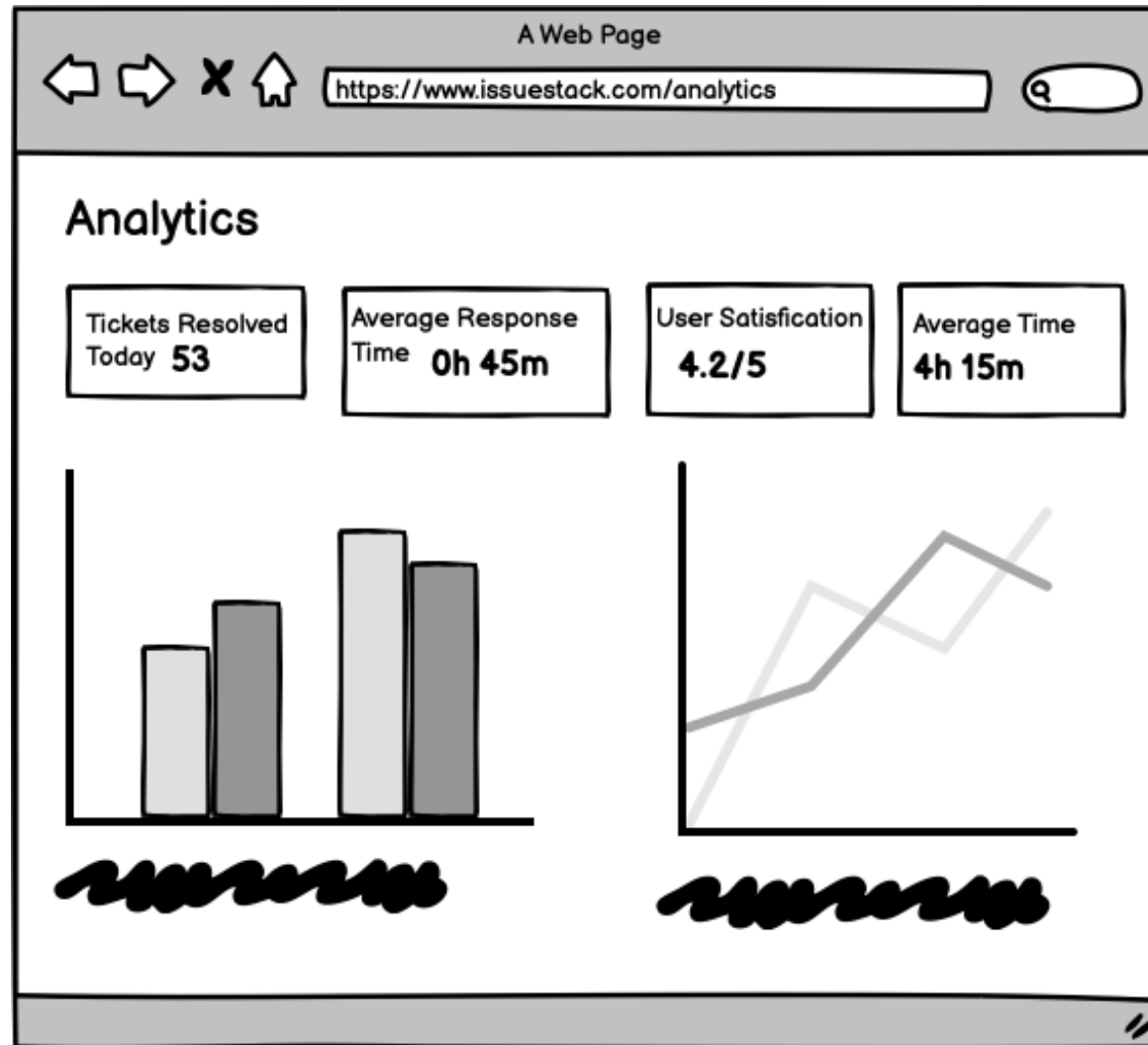
Figure 7: Prototype of Analytics

Figure 8: Prototype of settings
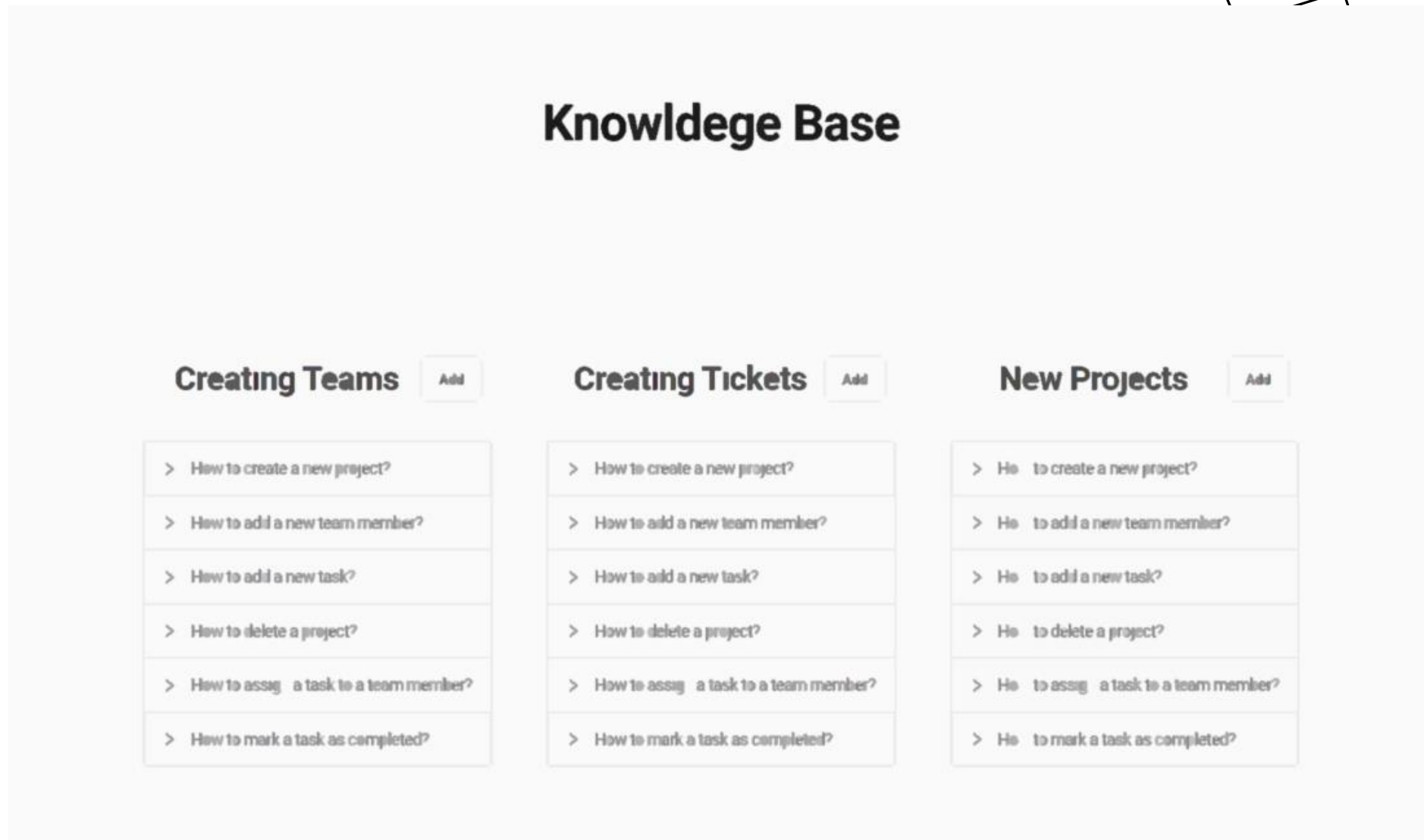
Figure 9: Prototype of knowledge base

WEBSITE DESIGN
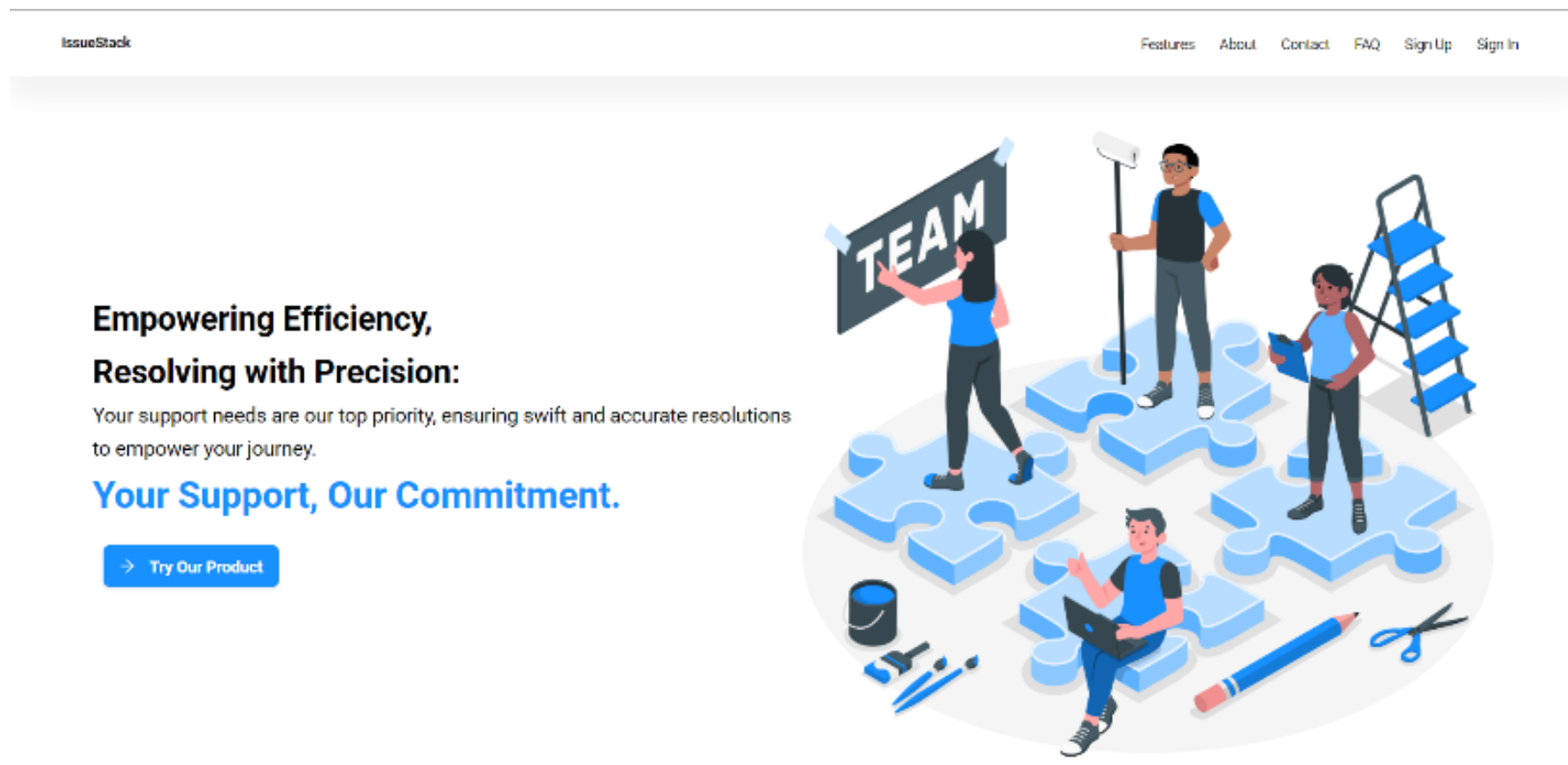
Figure 10: Landing Page

Figure 11: Sign Up Page

IssueStack

Sign In

Email

Password

SIGN IN

Don't have an account? **Sign Up**

Forgot Password? **Forgot Password**

Company   About Us   Blogs

Figure 12: Sign in Page
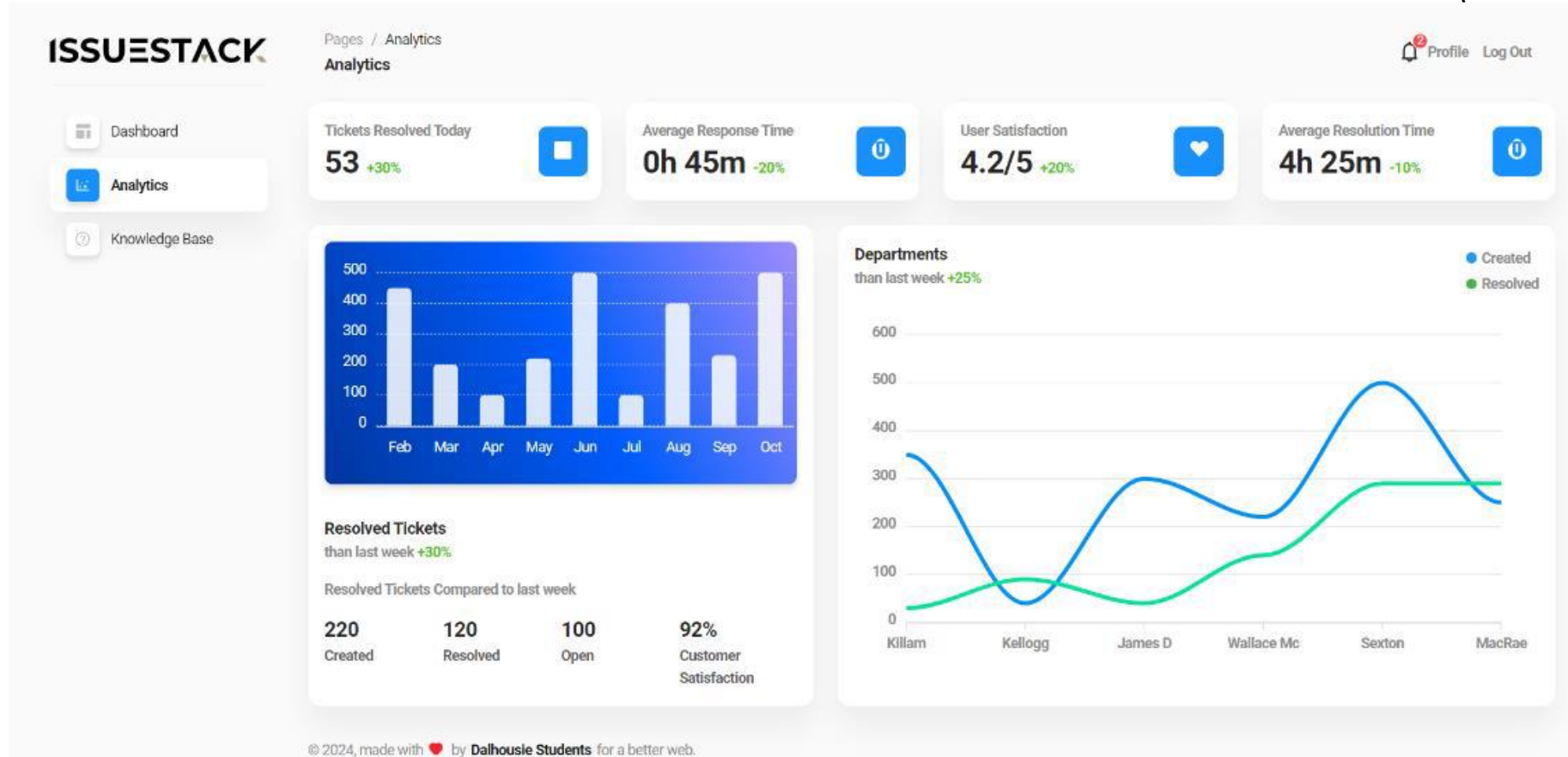
Figure 13: Update Profile Page

Figure 14: Analytics Page
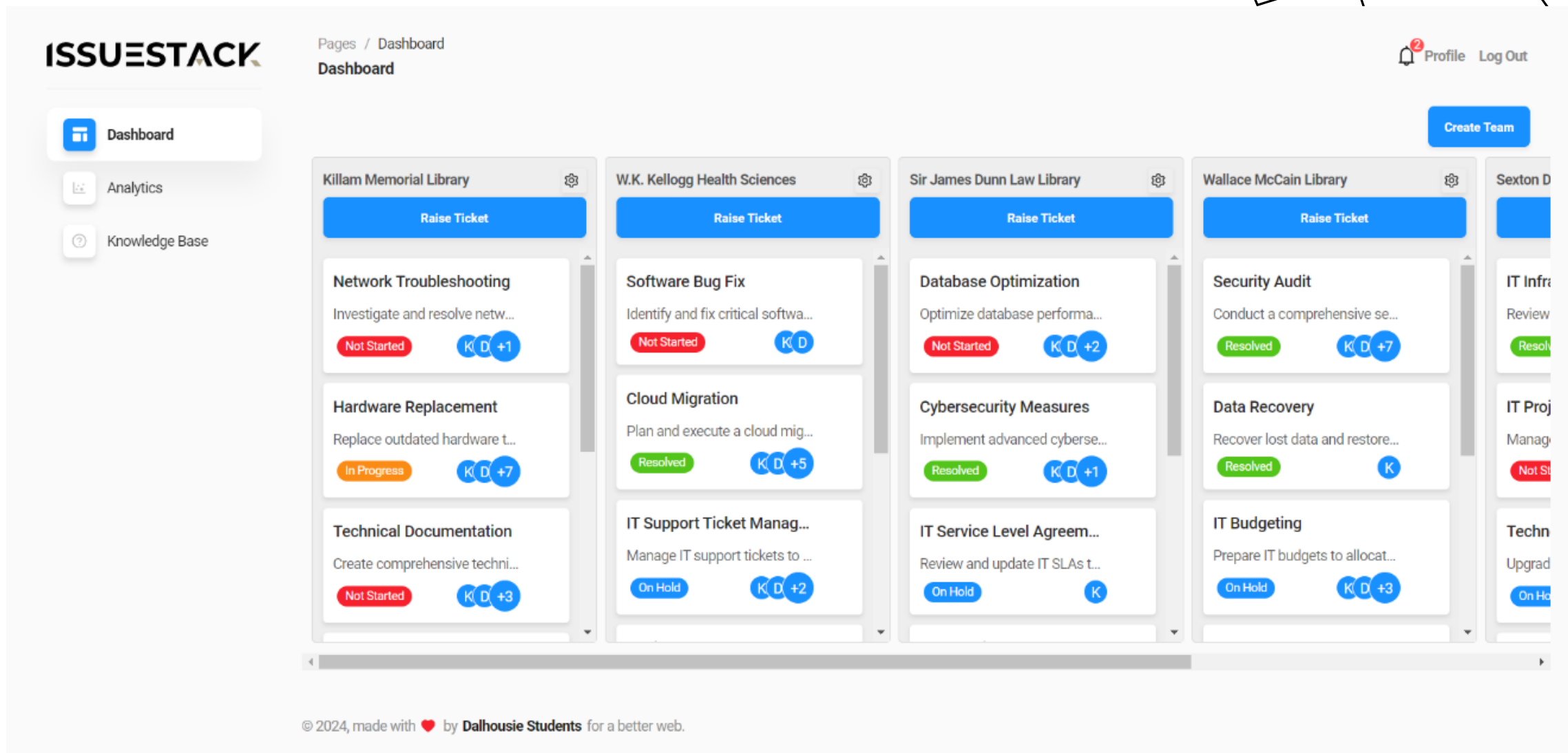
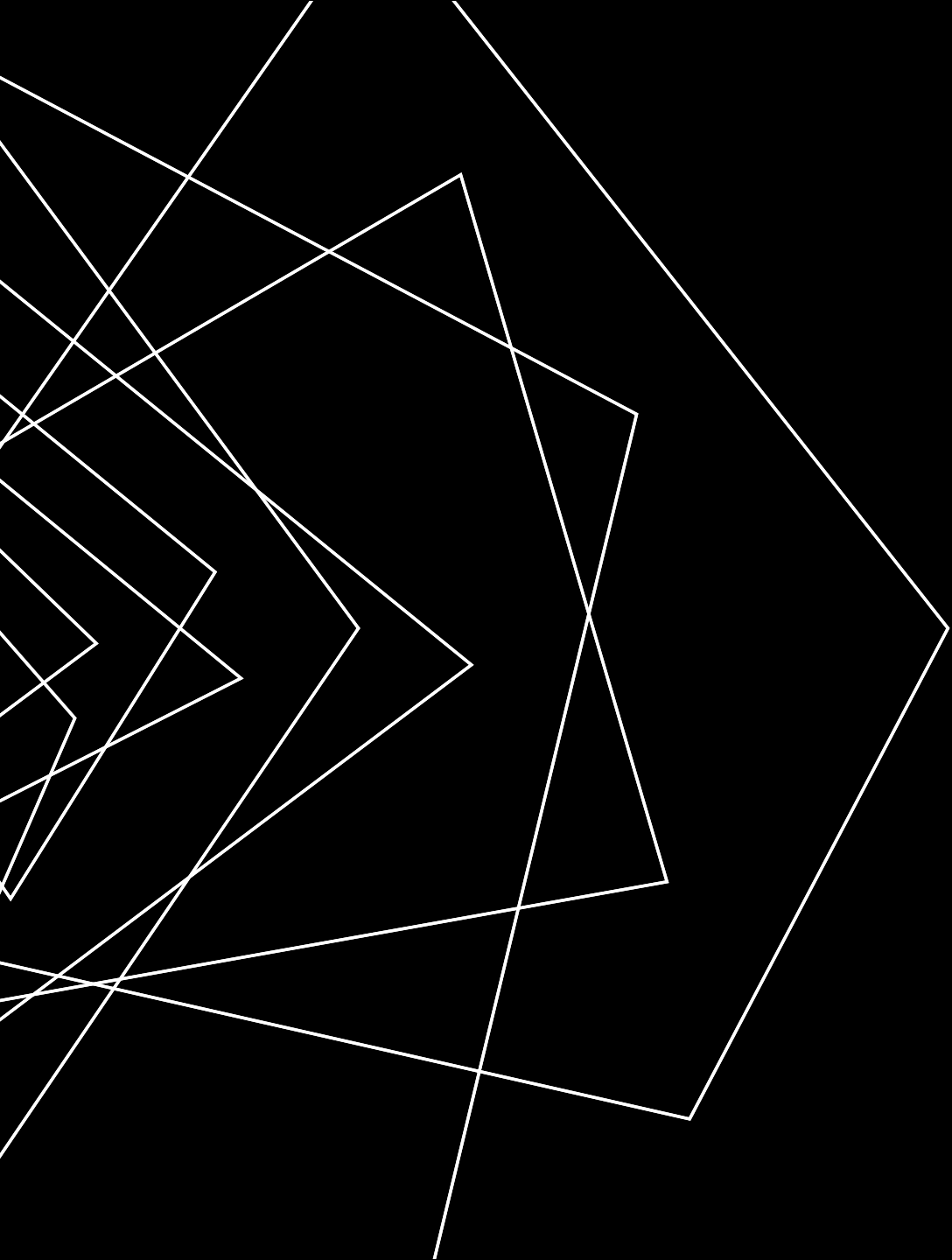**Figure 15:** Issue Board
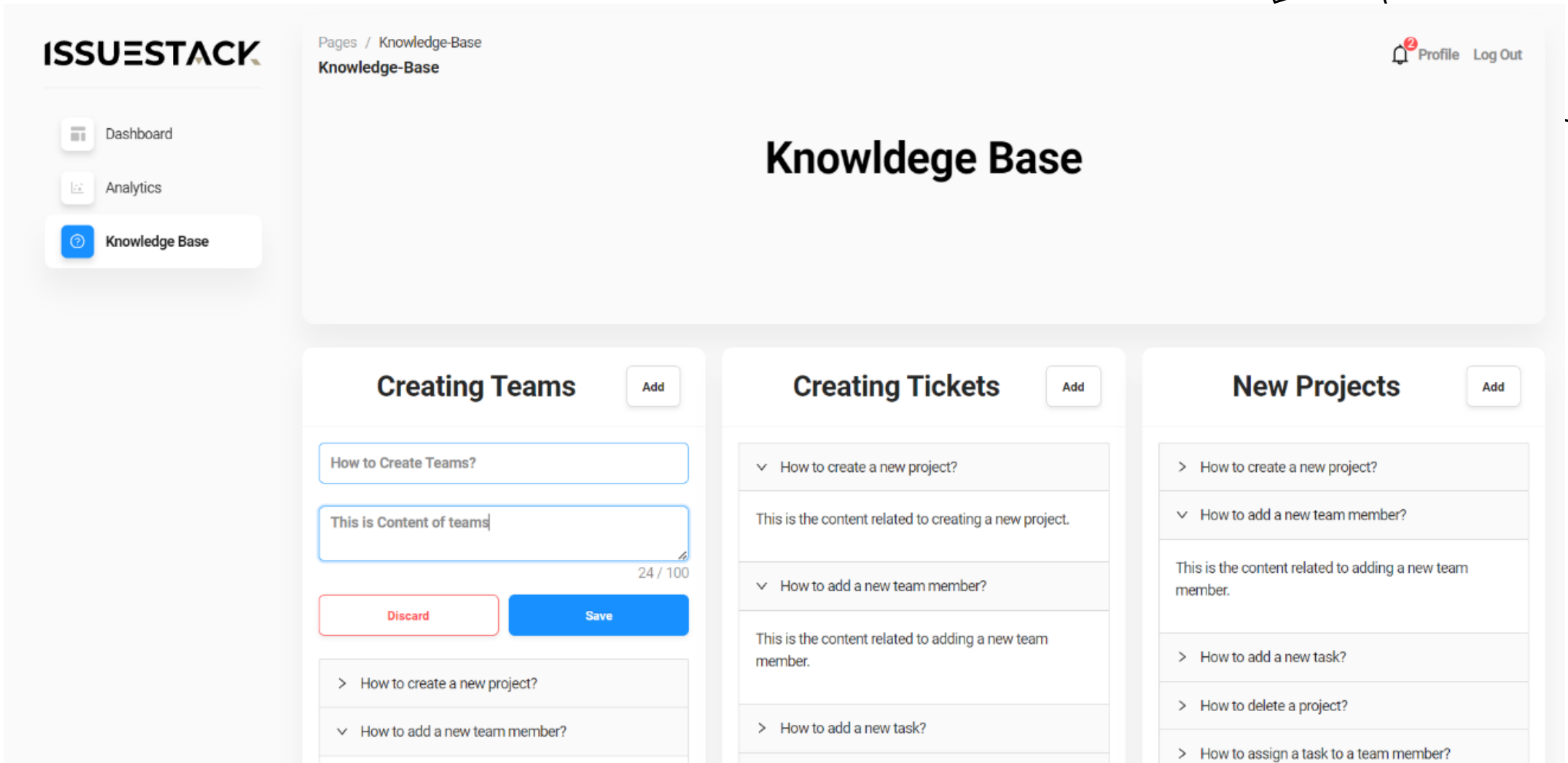
Figure 16: Knowledge Base

DEPLOYMENT LINK

- **FRONTEND - https://csci5709-web-project.netlify.app/**

- **BACKEND - https://csci5709-project.onrender.com/**

- **Gitlab Repo - https://git.cs.dal.ca/kanagaraj/csci-5708-grp-15**

THANK YOU