Observa – Milestone 1 Report

Team Members: (*Team 5)*

Ahmed Mustafa Elmorsy Amer	21010189
Ahmed Youssef Sobhy Elgoerany	21010217
Moustafa Esam El-Sayed Amer	21011364
Ebrahim Alaa Eldin Ebrahim	21010017
Ahmed Ayman Ahmed Abdallah	21010048
Ali Hassan Ali Mohamed	21010837

---

1. **Analyzing User's Requirements and Task Description:**
   - Through user scenarios:
     - I am Abdallah, I work till late hours, so I want to view the camera when I am away, so that I can see what/who is at my door at any time.
       - Receive the stream on the server.
       - Stream Camera from bell app.
       - Send the stream to the front end.
       - Show the received stream to the user.
     - I am Ahmed, I want to get notified with the name of the visitor, so that I know who is at the bell without opening the camera.
       - Facial Recognition of visitors.
       - Database with saved visitors faces.
       - Add database to model.
       - Upload images of new visitor faces.
       - View current visitor's name in dashboard.
       - Send notifications through the web.
     - I am Ayman, I work multiple jobs, so I am a bit forgetful. I want to change my email address and password because I forgot them.
       - Add settings page to edit the user info.
       - Add API requests for updating user info.
     - I am Said, I want to see who the last people who visited me are because I am sometimes busy and didn't see who was at my door.
       - Add visitors page to show the known visitors history.
     - I am Mohammed, I don't want to enter my email and password every time I login because it is repetitive and tiring.
       - Generate an authentication token.
       - Add remember me option.
       - Save the authentication token to a cookie.
     - I am Alaa, I want no one to be able to view my data as it invades my privacy.
       - Add authentication layer for each user for all requests.

        - Create login and signup pages.
        - Add login to bell app using api key.
        - Scan API key using QR code.
    - I am Mustafa, I am a 60-year-old senior. I want to have an easy way to access the camera, visitors, etc. because I am a bit tech illiterate.
        - Create a navigation bar for the users.
        - User-manual for the website.
    - I am Ebrahim, I am a person who gets confused and doesn't like complicated things. I want the application interface to be simple and beautiful.
        - Create wireframe designs for website pages.
        - Implement the pages designs with HTML, CSS, JS.
        - Implement the pages designs using VueJS.
        - Test the UI and UX of the pages.

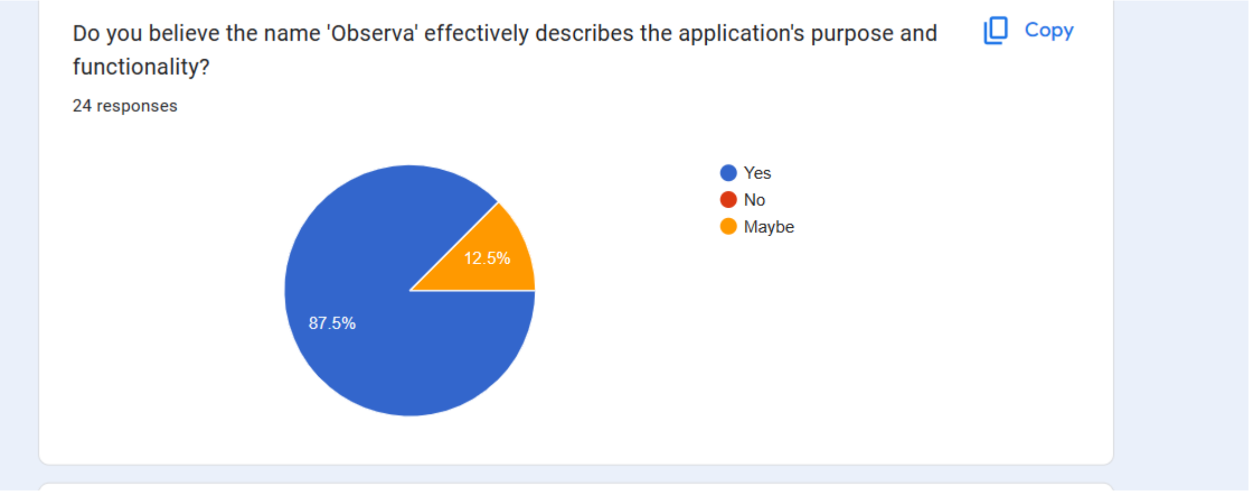
---

2. Data Gathering:

In the process of gathering data for our Observa Smart Doorbell System project, we employed a combination of surveys and collaborative teamwork. The data collection strategy involved two primary methods:

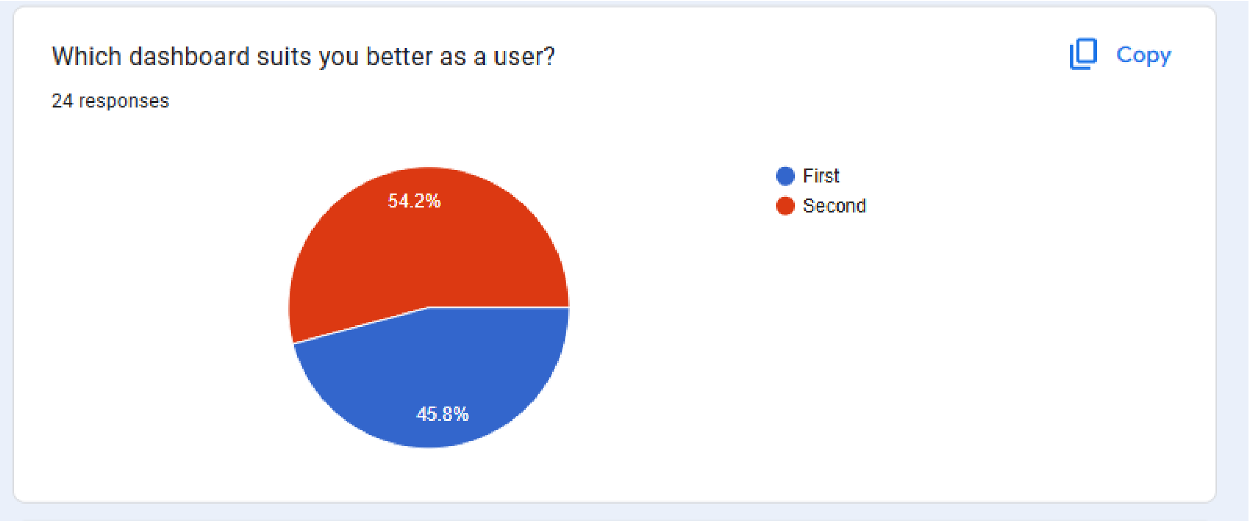
- Surveys:

We conducted surveys to systematically gather insights from potential users, aiming to understand their preferences, expectations, and specific requirements regarding home security systems. The survey responses provided valuable qualitative and quantitative data, informing the design and functionality aspects of the Observa system.



Do you believe the name 'Observa' effectively describes the application's purpose and functionality?

24 responses



Which dashboard suits you better as a user?

24 responses

- Team Collaboration:

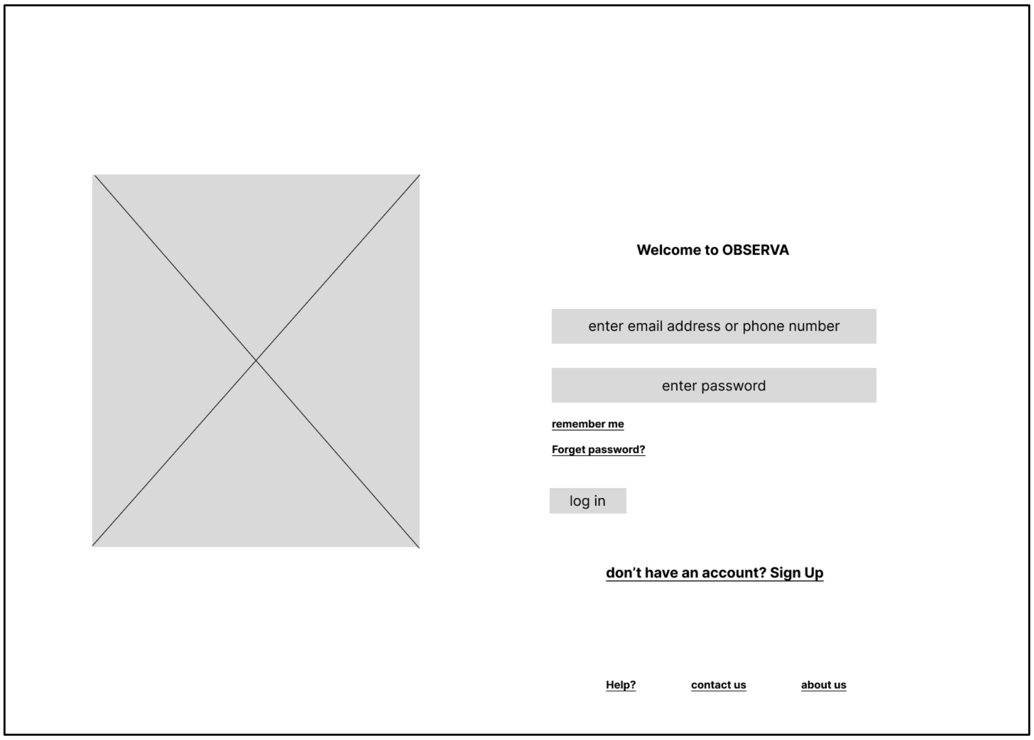
Internally, our team engaged in collaborative discussions and workshops to leverage the collective expertise and diverse perspectives within the group. Through brainstorming sessions and collaborative work, we were able to extract valuable insights, identify key challenges, and establish consensus on various design and functionality aspects of the smart doorbell system.

3. **Prototyping:**
   - In regard to the UI/UX design process, we first designed the wireframes, then came up with UI Design, then implemented some unfinalized front-end codes and got feedback from potential users, so we will work on the enhancements before the next milestone.

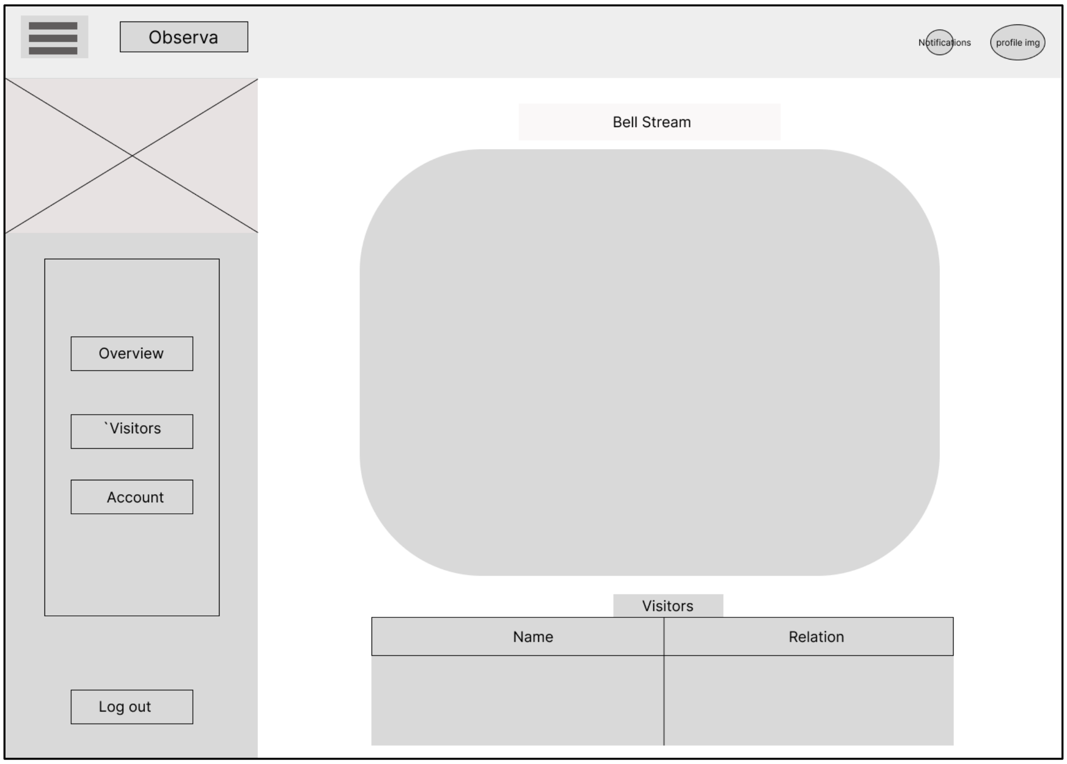
   - **Wireframing:**

The initial phase involved the creation of wireframes. Wireframes are low-fidelity, basic representations of the user interface layout and structure. They serve as a blueprint for the design, focusing on functionality and content placement without delving into visual details.

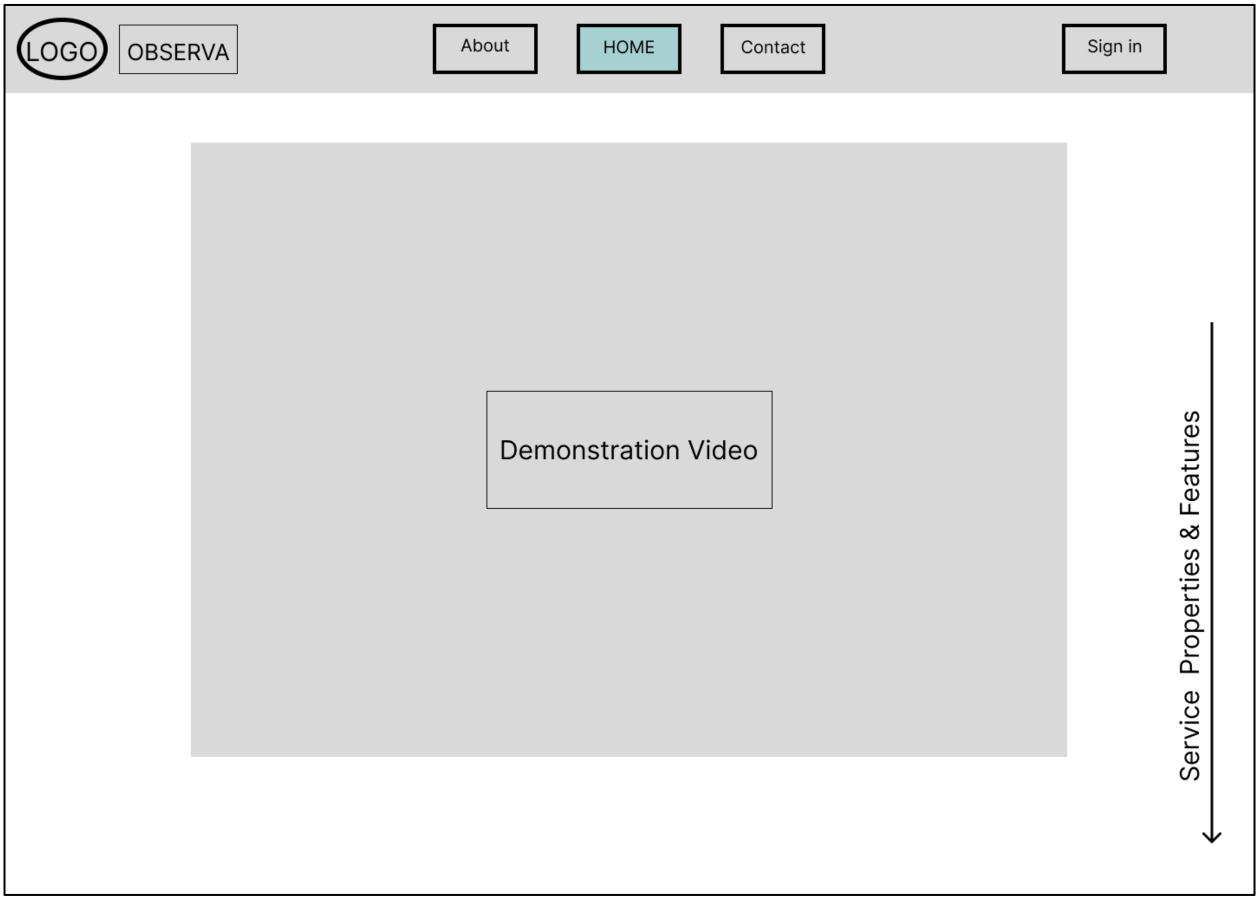
Log In page:



Dashboard (where Livestream appears):



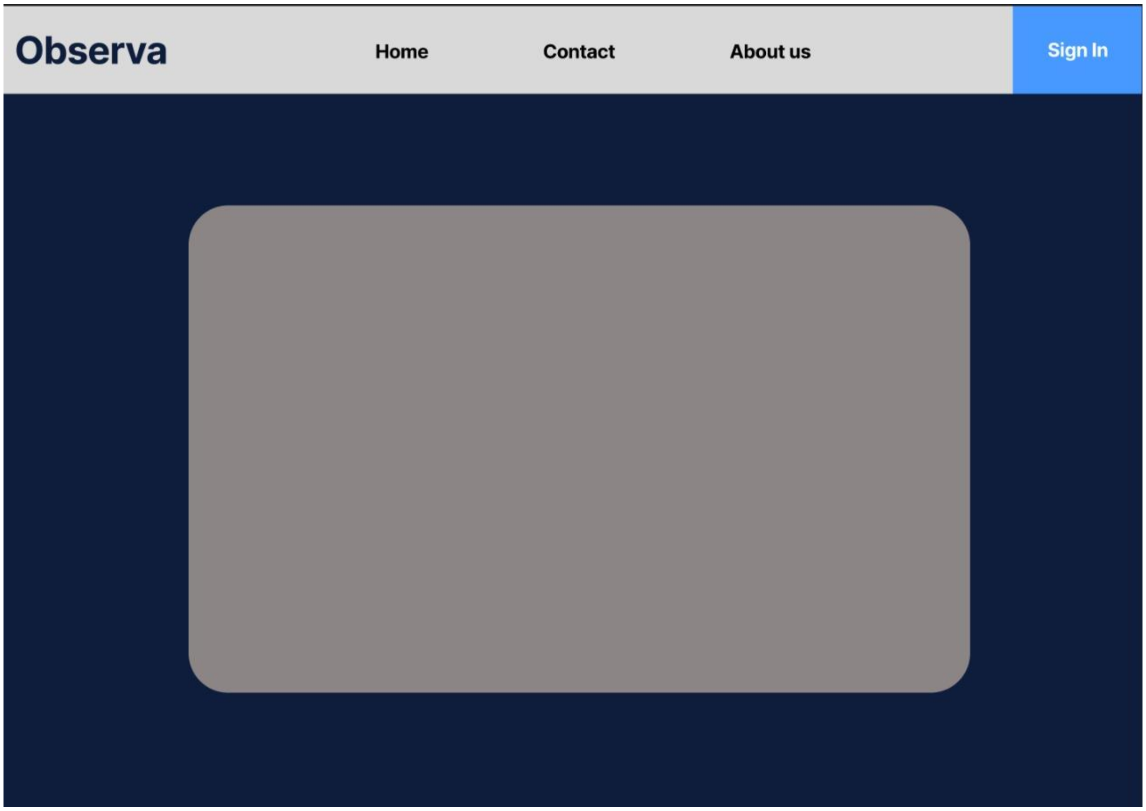
Home page (has How-To demonstration video):



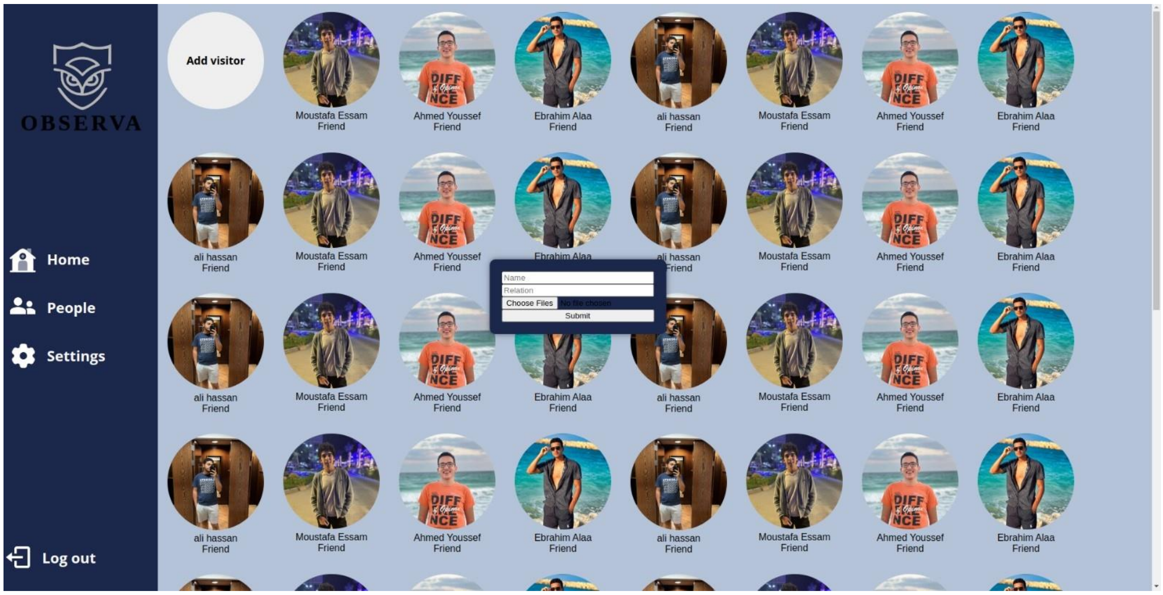
- **UI Design:**

Building upon the wireframes, the next step was the UI design. This phase focused on adding visual elements to the wireframes, such as color schemes, typography, and imagery. The goal was to create a visually appealing and cohesive design that aligns with the brand identity and user preferences.

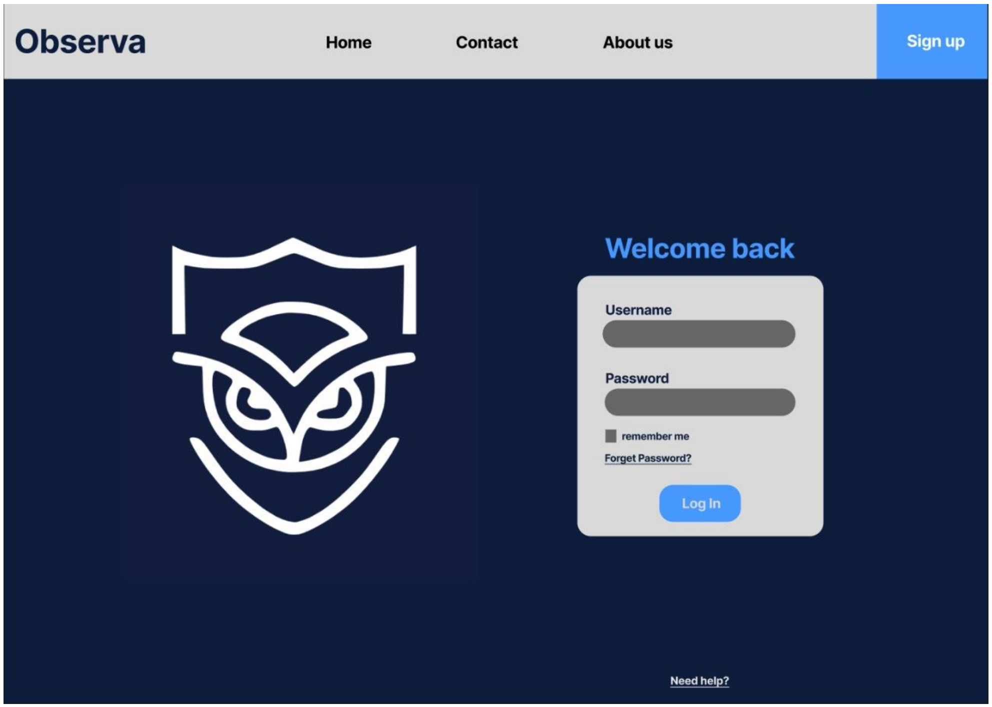
Home page:

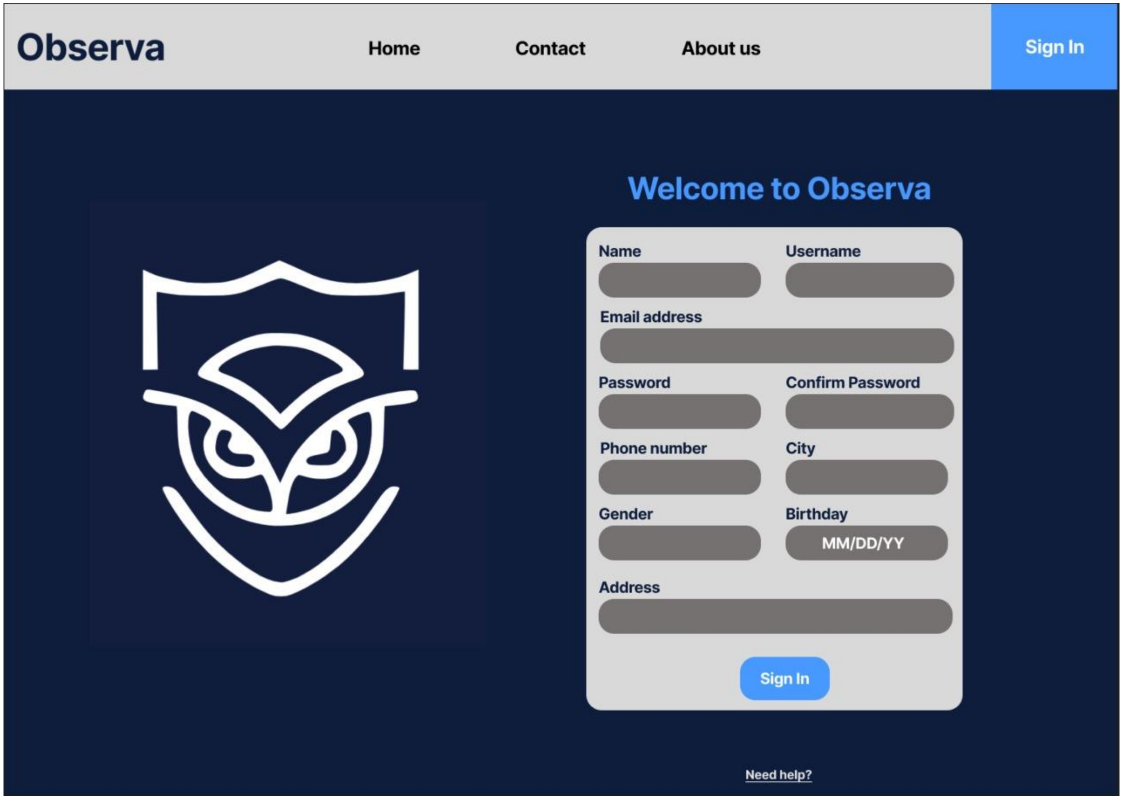


Peoples Page

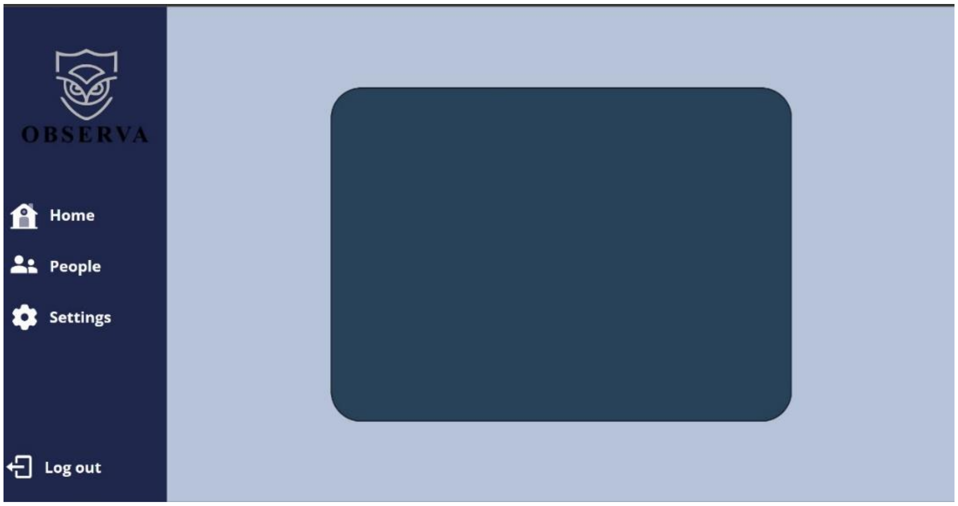


Log In/ Sign Up



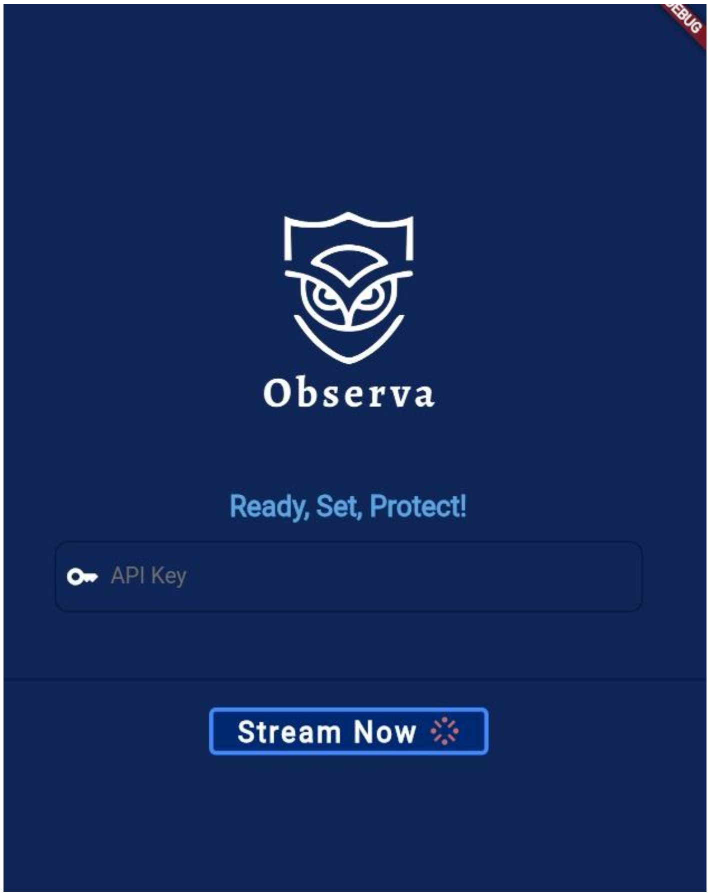


Dashboard



Mobile Bell Application:

- Using Flutter framework



- **Iterative Testing and Feedback:**
  - ➢ The UI design needed a lot of testing and feedback from potential users, so this is not the final design, and we will conduct more surveys to fulfill all requirements that satisfy user's experience and expectations.

---

4. Project Management:

Tools Used:

1. **Notion with Scrum Framework:**
   - Notion served as the central project management hub, leveraging the Scrum framework. Scrum principles provided a flexible and iterative approach, enabling the team to adapt to changing requirements and continuously improve the development process.
   - Notion Workspace

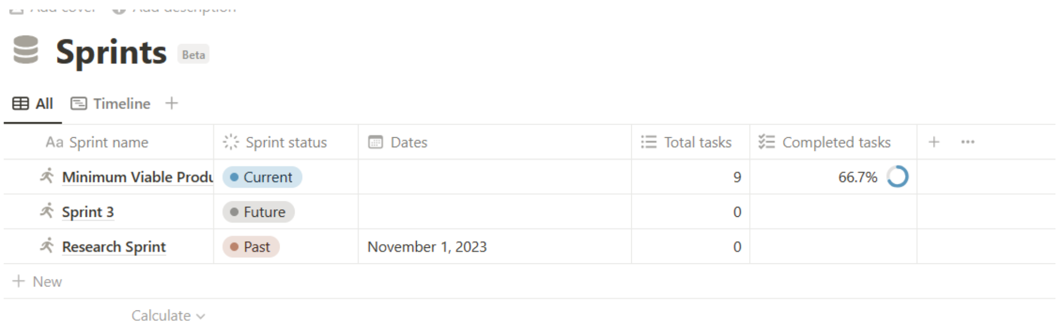
2. **Figma:**
   - Figma played a crucial role in the UI/UX design process. It allowed the team to collaboratively create and iterate on design prototypes. Figma's cloud-based platform facilitated real-time collaboration, streamlining communication among team members involved in the design phase.
   - Figma Workspace

3. **GitHub:**
   - GitHub served as the version control system for the project, enabling collaborative development. It allowed the team to manage and track changes to the codebase, facilitating seamless collaboration among front-end and back-end developers.
   - GitHub Repository Link

Development Phases:

1. **First Sprint - Research:**
   - The project kicked off with a research sprint, focusing on understanding user needs, market trends, and technological requirements. This information guided the subsequent phases of development, ensuring a user-centric and technically feasible solution.
2. **Minimum Viable Product (MVP):**

- The second sprint focuses on developing the Minimum Viable Product. This involved implementing the core functionalities necessary for the system to be functional and valuable to users. The MVP approach allows for quicker delivery of a basic yet functional version for testing and validation.



Sprints Beta

All | Timeline | +

Sprint name	Sprint status	Dates	Total tasks	Completed tasks
Minimum Viable Produ	Current		9	66.7%
Sprint 3	Future		0	
Research Sprint	Past	November 1, 2023	0	

+ New

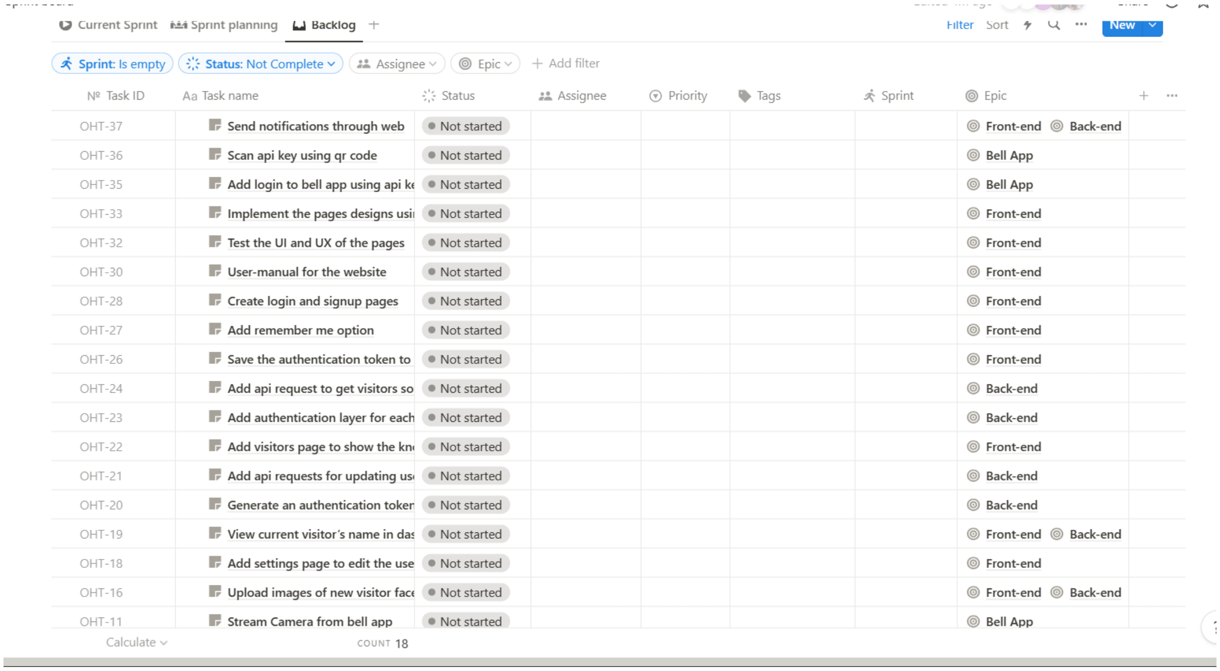
Calculate

Task Distribution and Epics:

- Tasks are distributed across different epics to streamline development efforts. Epics were categorized into front-end and back-end, aligning with the division of responsibilities among team members. This approach facilitated focused development and allowed team members to work efficiently within their respective domains of expertise.

User Stories:

- User stories are employed to define and communicate specific features and functionalities from the end-user's perspective. This user-centric approach ensures that development efforts are aligned with the actual needs and expectations of the target users.



Current Sprint | Sprint planning | Backlog | +

Filter | Sort | New

Sprint: Is empty | Status: Not Complete | Assignee | Epic | + Add filter

Task ID	Task name	Status	Assignee	Priority	Tags	Sprint	Epic
OHT-37	Send notifications through web	Not started					Front-end Back-end
OHT-36	Scan api key using qr code	Not started					Bell App
OHT-35	Add login to bell app using api ke	Not started					Bell App
OHT-33	Implement the pages designs usi	Not started					Front-end
OHT-32	Test the UI and UX of the pages	Not started					Front-end
OHT-30	User-manual for the website	Not started					Front-end
OHT-28	Create login and signup pages	Not started					Front-end
OHT-27	Add remember me option	Not started					Front-end
OHT-26	Save the authentication token to	Not started					Front-end
OHT-24	Add api request to get visitors so	Not started					Back-end
OHT-23	Add authentication layer for each	Not started					Back-end
OHT-22	Add visitors page to show the kn	Not started					Front-end
OHT-21	Add api requests for updating us	Not started					Back-end
OHT-20	Generate an authentication token	Not started					Back-end
OHT-19	View current visitor's name in das	Not started					Front-end Back-end
OHT-18	Add settings page to edit the use	Not started					Front-end
OHT-16	Upload images of new visitor face	Not started					Front-end Back-end
OHT-11	Stream Camera from bell app	Not started					Bell App

Calculate COUNT 18

Progress Tracking:

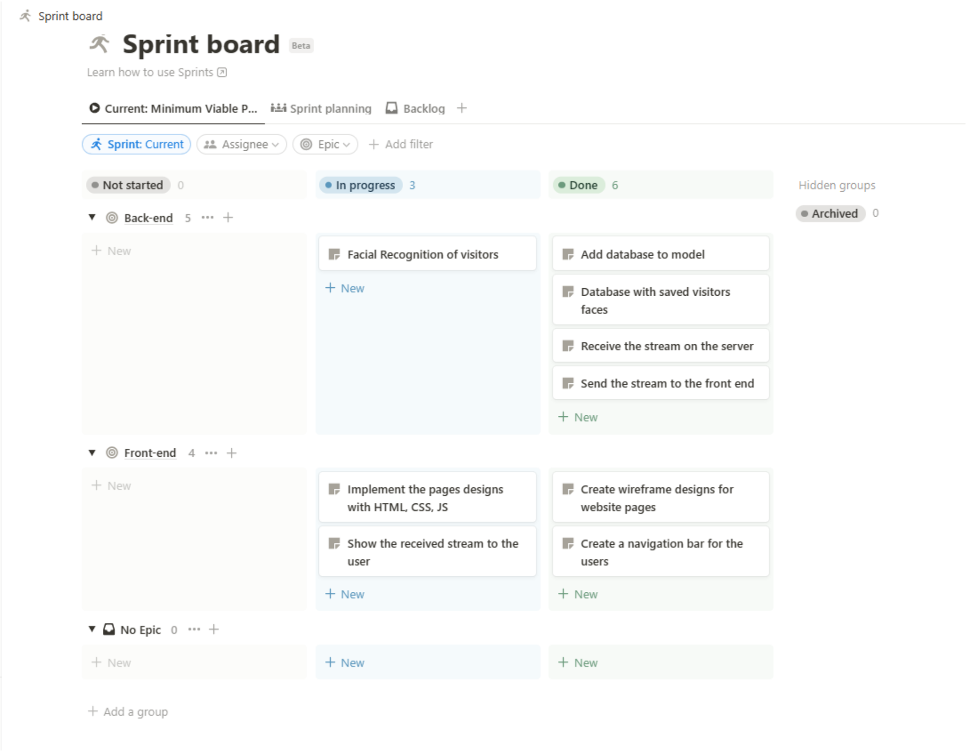
- Progress was tracked using a Kanban board. This visual representation allowed the team to monitor the status of tasks immediately, from "To Do" to "In Progress" and finally "Done." The Kanban board facilitated transparency and enabled quick identification of bottlenecks or areas that required additional attention.

Iterative Development:

- The use of Scrum principles, user stories, and iterative development cycles allowed the team to adapt to changing requirements and continuously improve the product. Regular sprint reviews and retrospectives ensured that the team learned from each iteration, fostering a culture of continuous improvement.

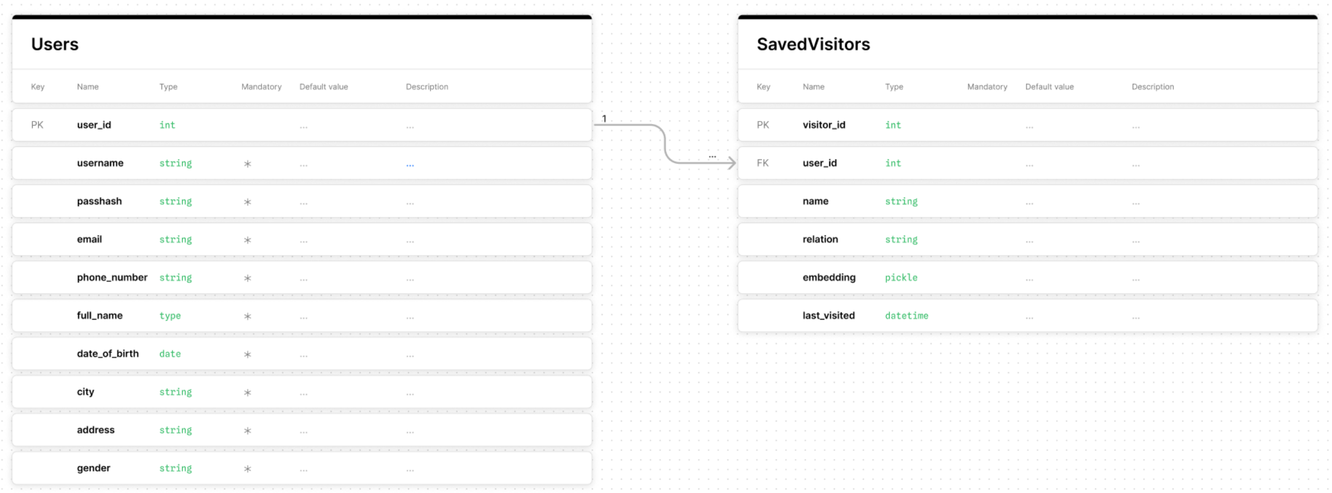
Collaborative Environment:

- The combination of tools like Notion, Figma, and GitHub contributed to a collaborative and efficient work environment. Real-time collaboration in design, version control in development, and organized project management in Notion collectively supported the team throughout the project lifecycle.



	Not started 0	In progress 3	Done 6
Back-end 5	+ New	Facial Recognition of visitors + New	Add database to model Database with saved visitors faces Receive the stream on the server Send the stream to the front end + New
Front-end 4	+ New	Implement the pages designs with HTML, CSS, JS Show the received stream to the user + New	Create wireframe designs for website pages Create a navigation bar for the users + New
No Epic 0	+ New	+ New	+ New

5. Database Management



Users

Key	Name	Type	Mandatory	Default value	Description
PK	user_id	int		...	...
	username	string	*	...	...
	passhash	string	*	...	...
	email	string	*	...	...
	phone_number	string	*	...	...
	full_name	type	*	...	...
	date_of_birth	date	*	...	...
	city	string	*	...	...
	address	string	*	...	...
	gender	string	*	...	...

SavedVisitors

Key	Name	Type	Mandatory	Default value	Description
PK	visitor_id	int		...	...
FK	user_id	int		...	...
	name	string		...	...
	relation	string		...	...
	embedding	pickle		...	...
	last_visited	datetime		...	...