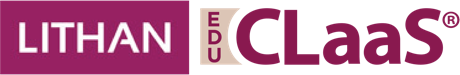
Assignment 2 Brief 

|  |  |
| --- | --- |
| **Product Name** | Applied Degree in Software Engineering (BDSE) |
| **Qualification Name** | Applied Degree in Software Engineering/ Higher Diploma in Software Engineering |
| **Assignment title** | Develop Project Proposal |
| **Module Name (BDSE)** | Develop Enterprise Applications |

|  |  |
| --- | --- |
| **Student Name** | **Mentor Name** |
| **Wildan Luqmanul Hakim** | Arvinder Kaur |

|  |  |
| --- | --- |
| **Project Title** | Meals on Wheels Software Design Document. |

|  |
| --- |
| **Learner declaration** |
| I certify that the work submitted for this assignment is my group work, and research sources are fully acknowledged.      Student signature: Date: January 25, 2023 |

[**Application Flowchart:**](#_akle5tbnka53) **3**

[**Use Case Diagram**](#_6ofoaokd9c92) **5**

[**Module Wireframe**](#_3r5uuau1r4xy) **6**

[**Module Storyboard**](#_6shw2tjhd7nl) **16**

[**Login as a:**](#_nwo47jmwu035) **16**

[**DB Design:**](#_1pc0ppb8vdl2) **17**

[**Software Development Life Cycle**](#_kuefd04owhle) **18**

[SDLC models](#_2226a8tusfvd) 18

[Waterfall Model](#_a631dzdkt1kn) 18

[Spiral Model](#_jybtdouebvfb) 18

[V-Model](#_1ao94t77y021) 19

[Agile](#_iydirokehku) 20

[SDLC Comparison](#_h9nyw4gc92bb) 20

[Waterfall](#_kb6j04wb4i4o) 20

[Spiral](#_6t3lssfeqyqt) 21

[V-Model](#_k1eu0b10bq7n) 21

[Agile](#_tfa9zd1dnd5u) 21

[When to use](#_ry5lx98y82vz) 22

[SDLC Chosen Method](#_5iffytu4dg7c) 22

[**Front-End Framework:**](#_g1gnri308j8s) **23**

[AngularJs](#_vtaazrtojqvl) 23

[ReactJS](#_rpqc9szhwdbo) 23

[VueJS](#_mtdxwjj9fqrs) 24

[Comparison](#_6whg83ocn8hv) 25

[**Back-end Framework:**](#_nhrobxx6zfne) **26**

[Spring boot](#_wbs2vl3p98m) 26

[**Database:**](#_ahomvtlau5b) **27**

[MongodB](#_uov2w856jzin) 27

[MySQL](#_stz8i4b7vjkh) 27

[Comparison](#_56oggzivttrl) 28

[**Integrated Development Environment:**](#_yflleoin3ar6) **29**

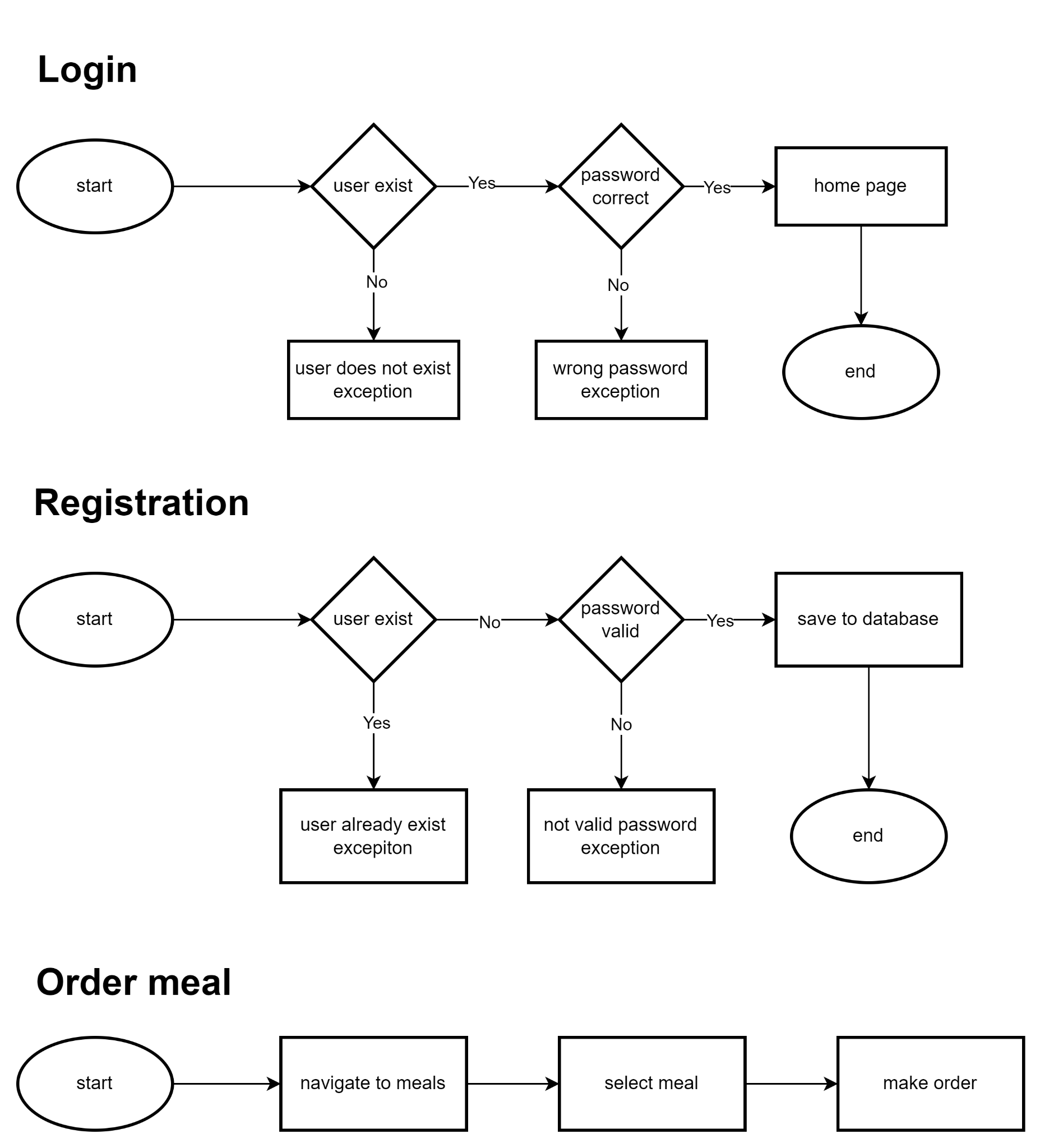
[IntelliJ](#_fn570miynnw6) 29

[Eclipse](#_41kb50gqcwyg) 29

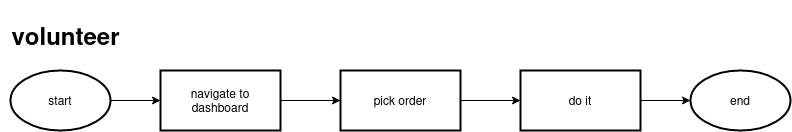
[STS](#_pwji7alxd7jz) 29

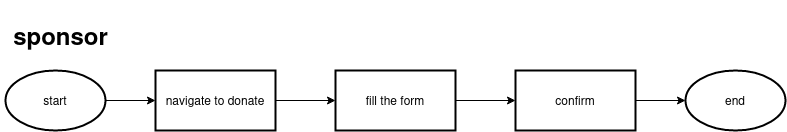
[**Bibliography**](#_ur63suhs989r) **31**

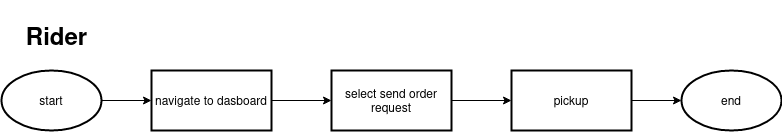
## Application Flowchart:

****

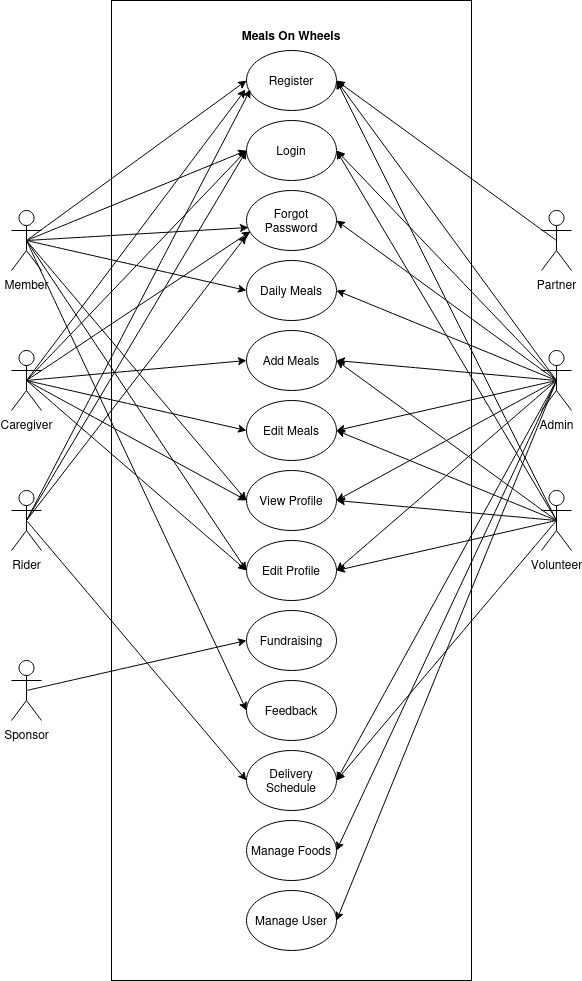
****

****

****

****

## Use Case Diagram

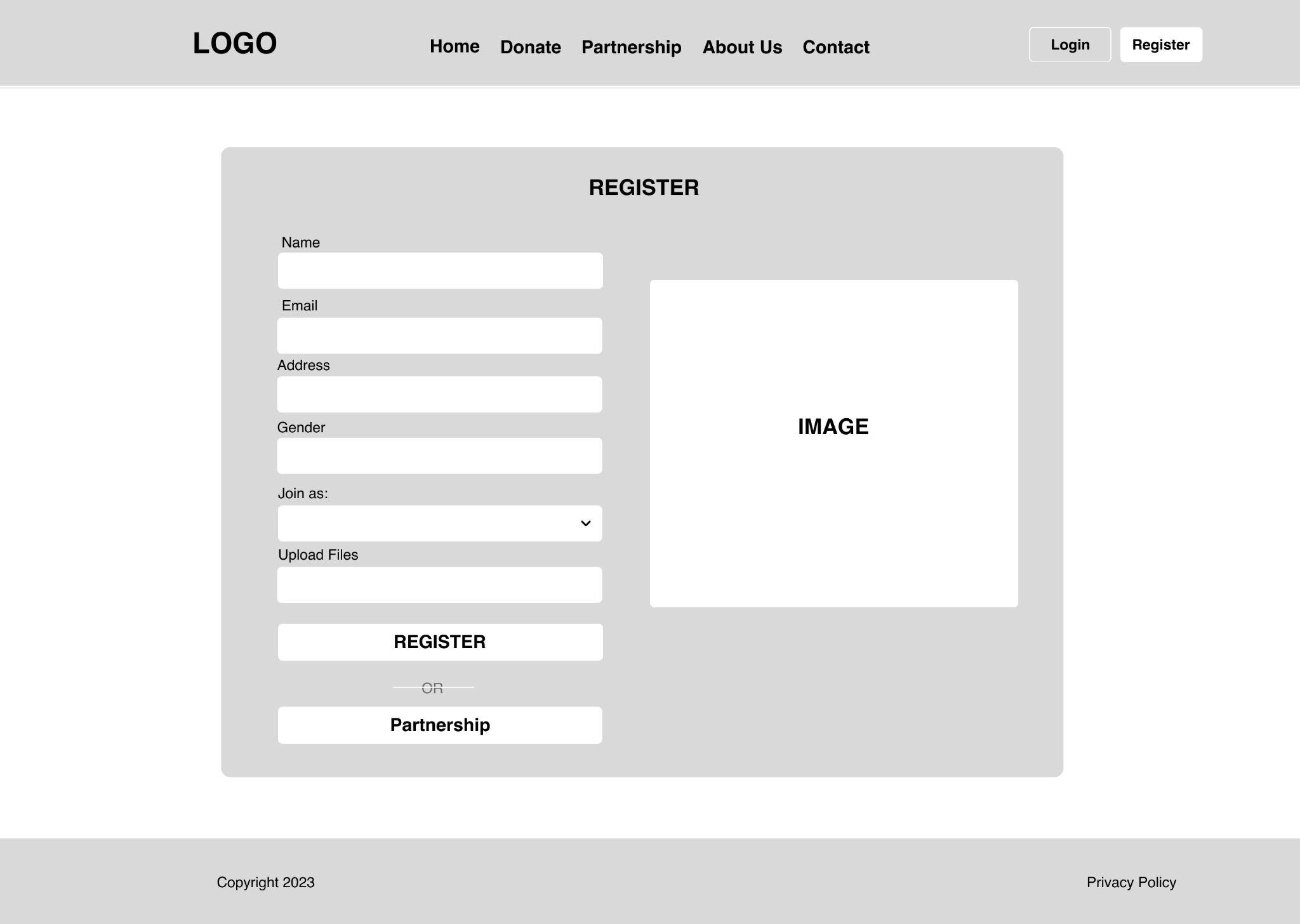


## Module Wireframe

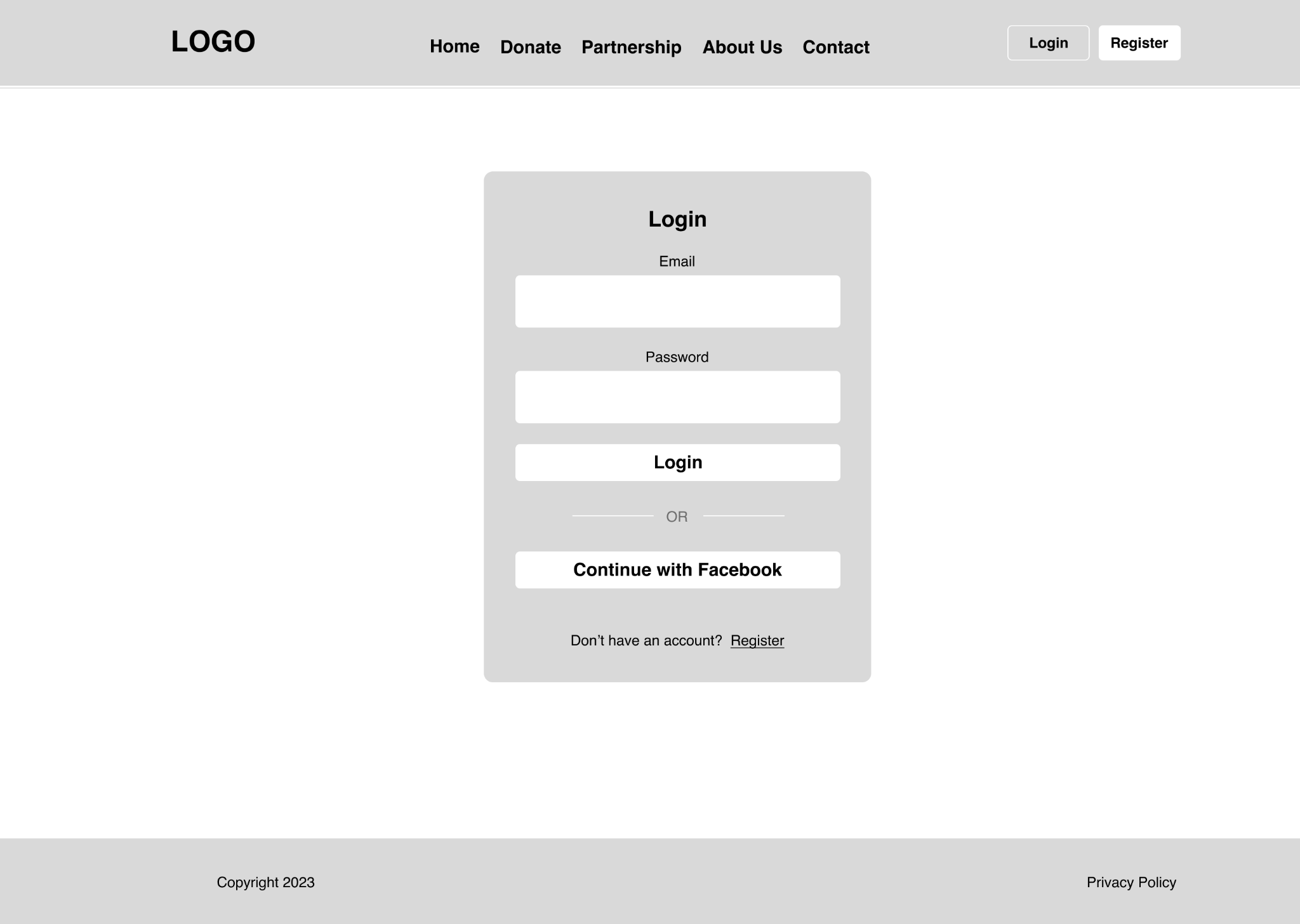
Landing Page



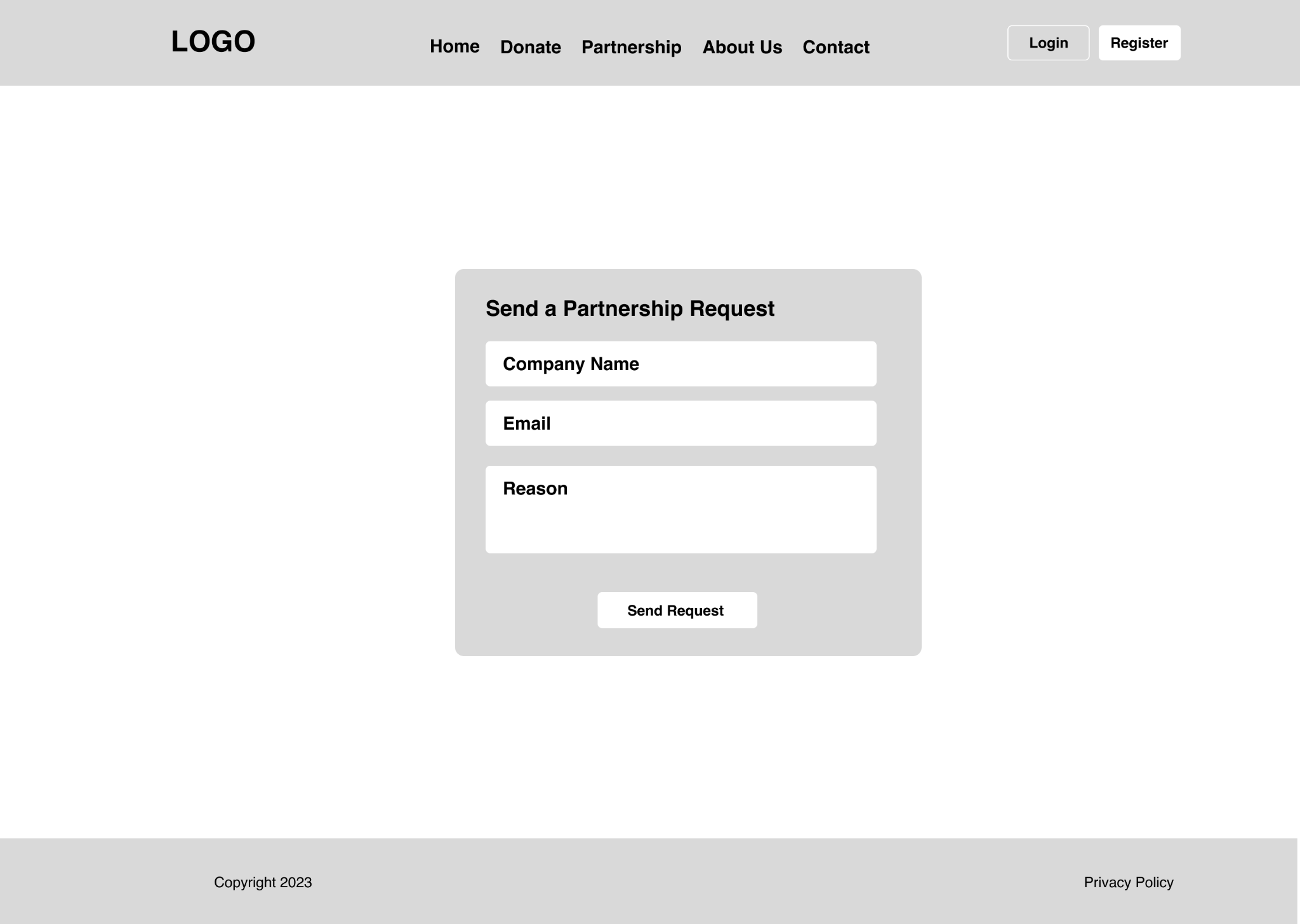
Register Page

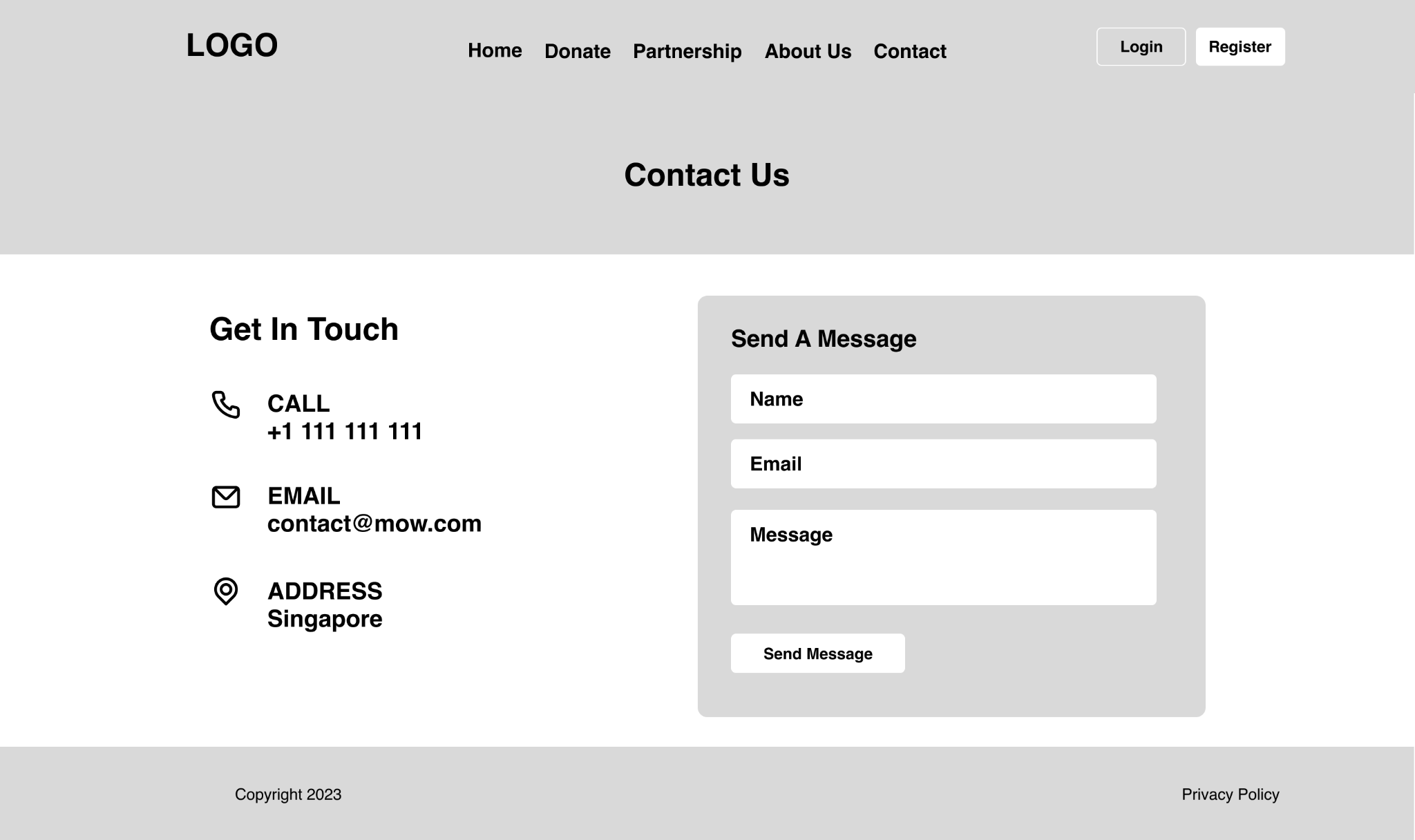


Login Page

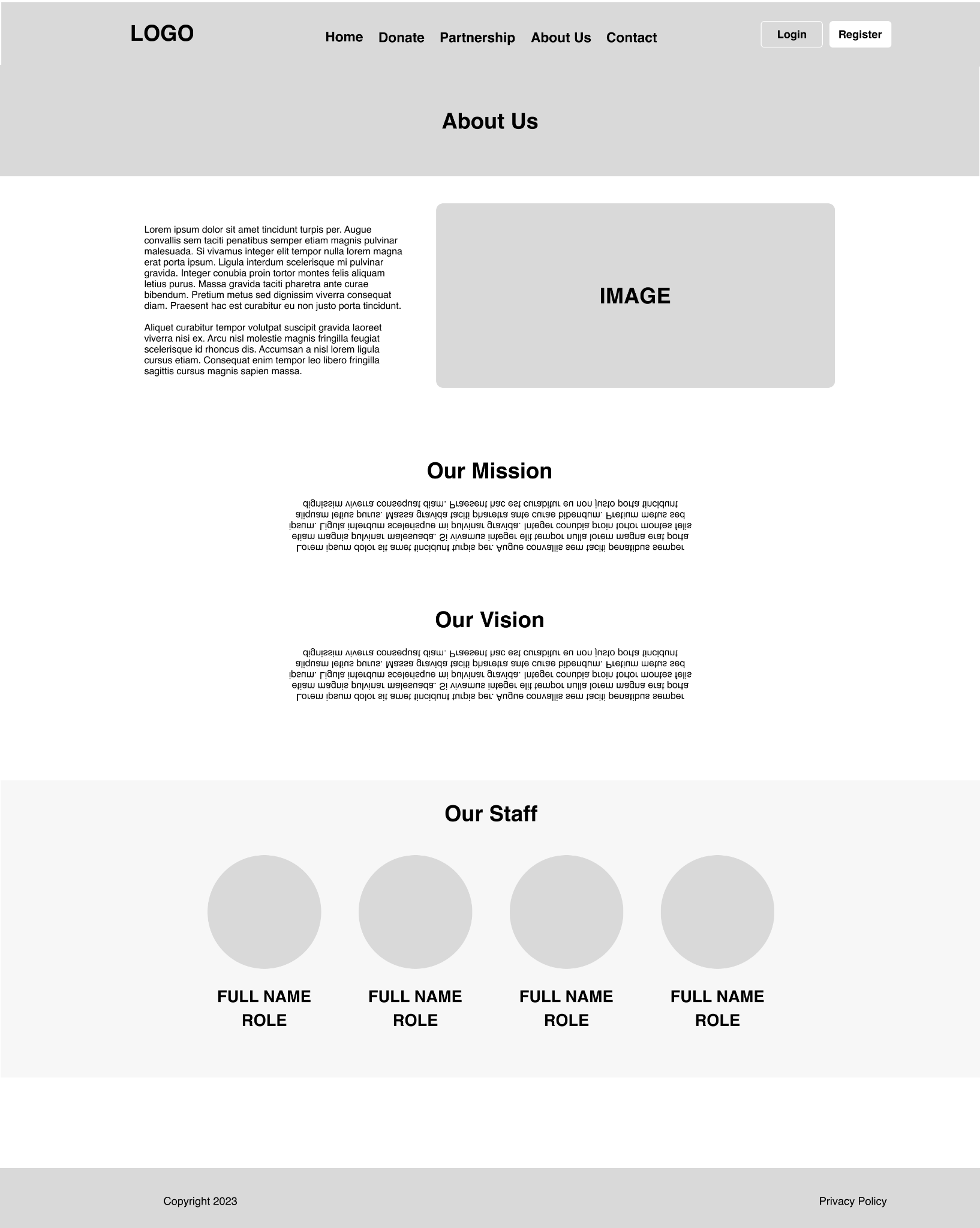


Partnership Request Page

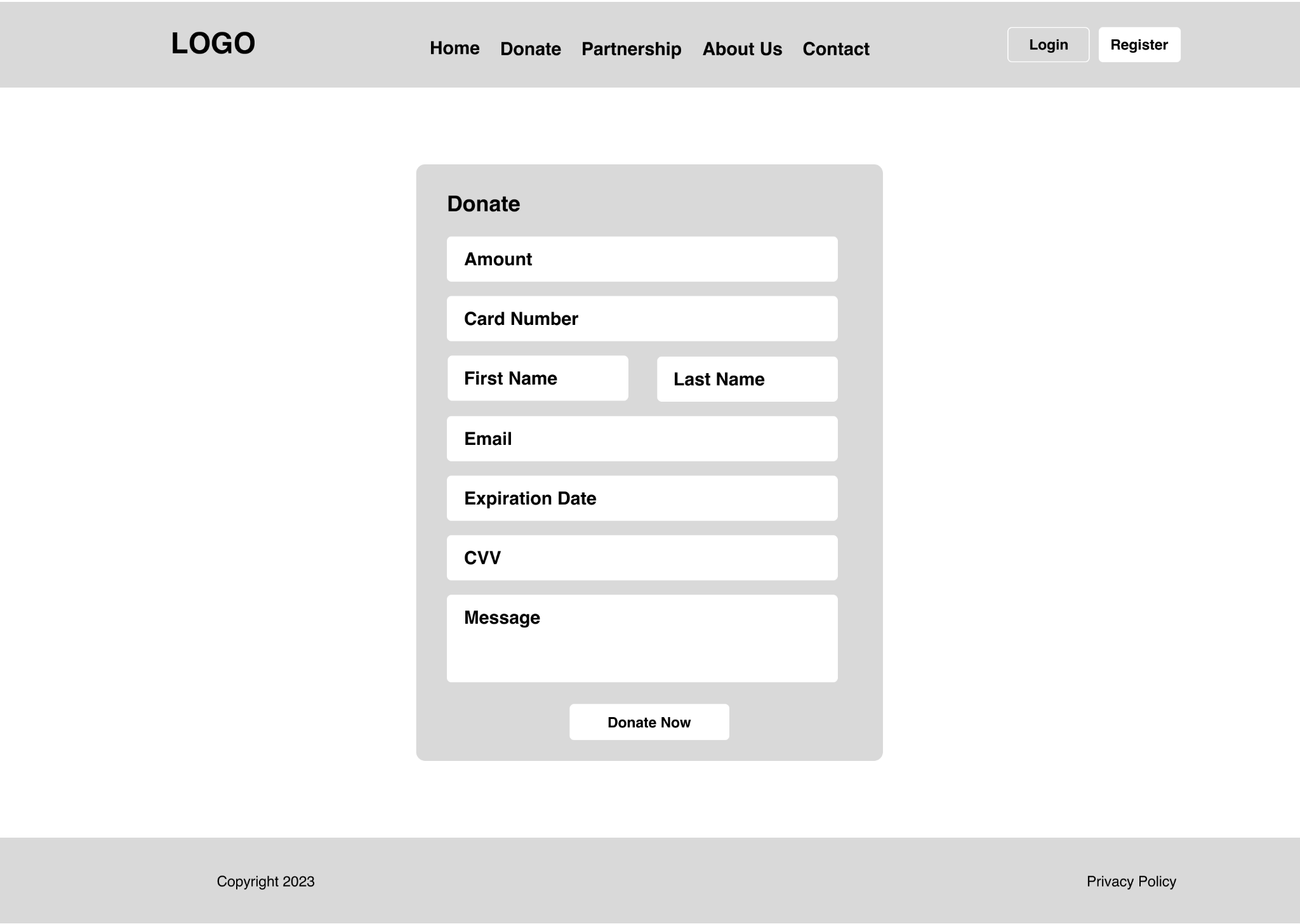
Contact Page



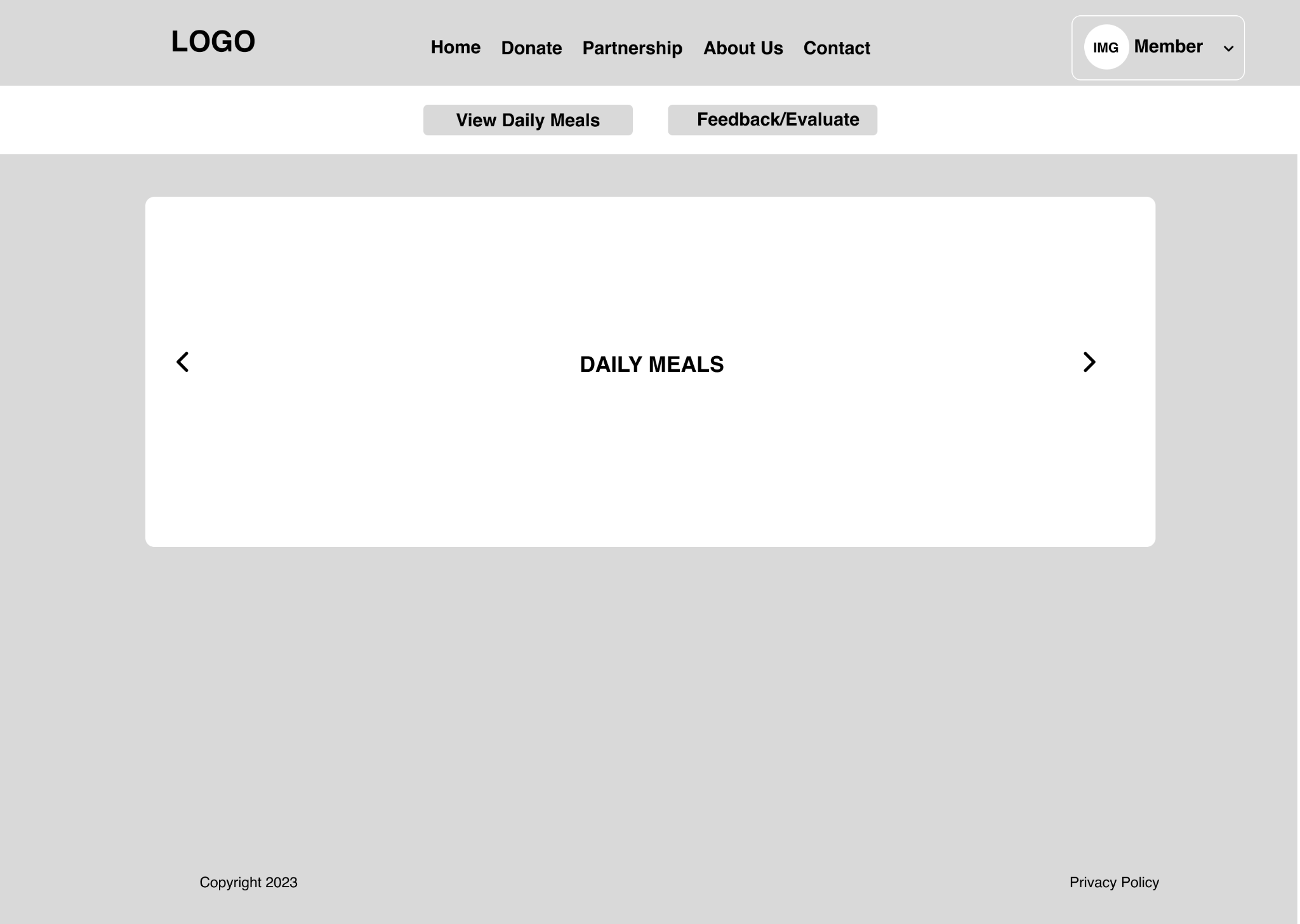
About Us Page



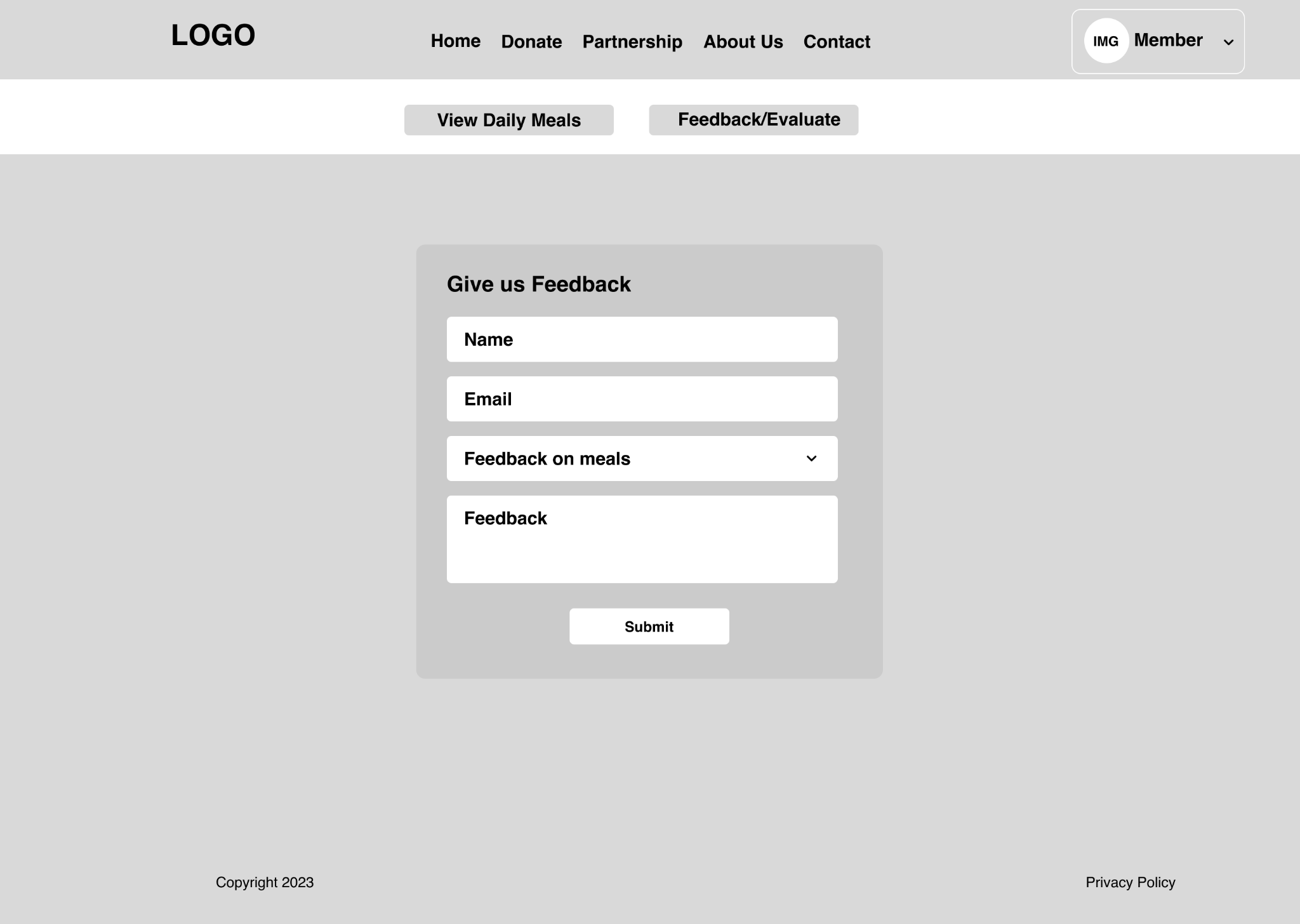
Fundraising Page



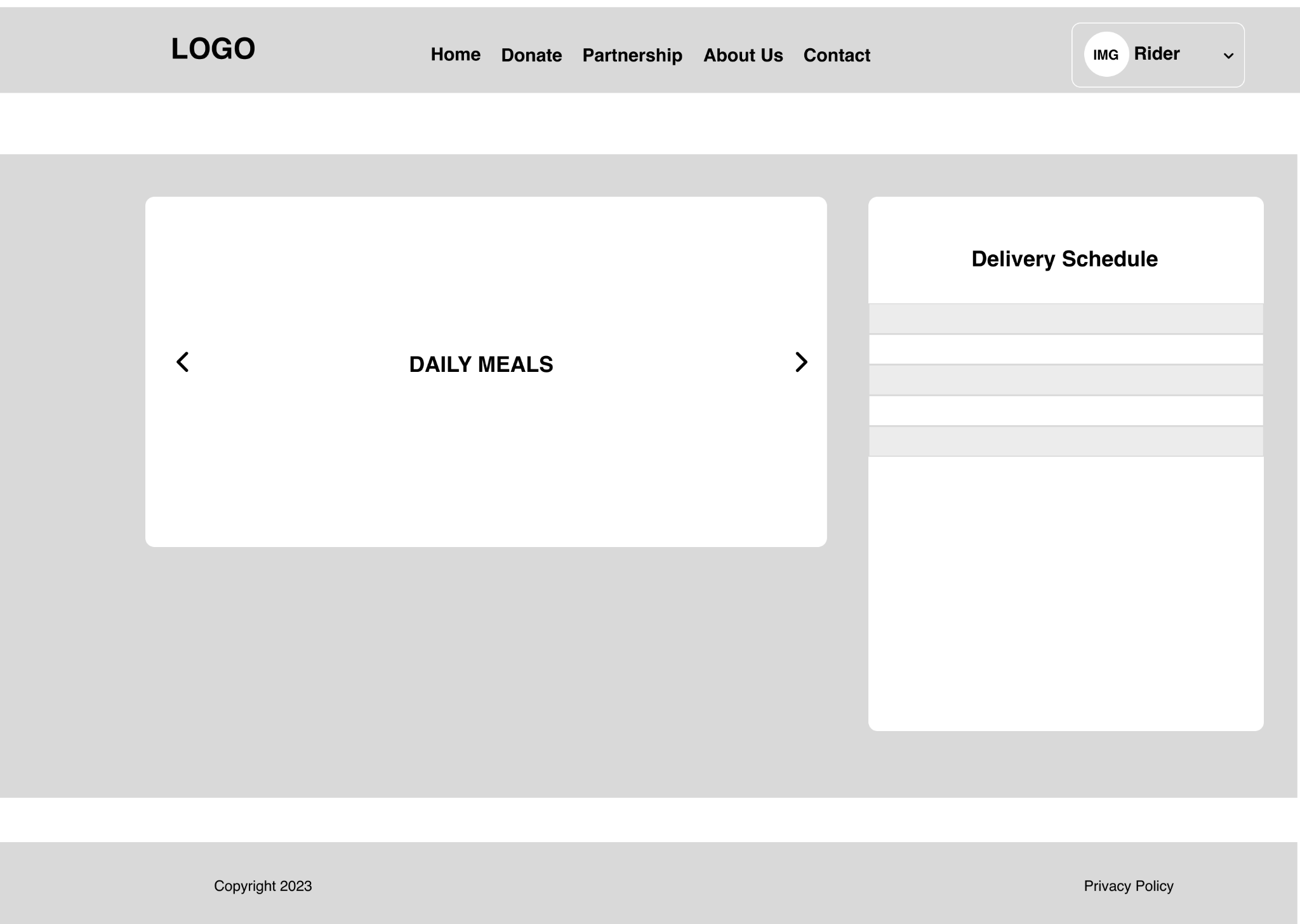
Member: Homepage



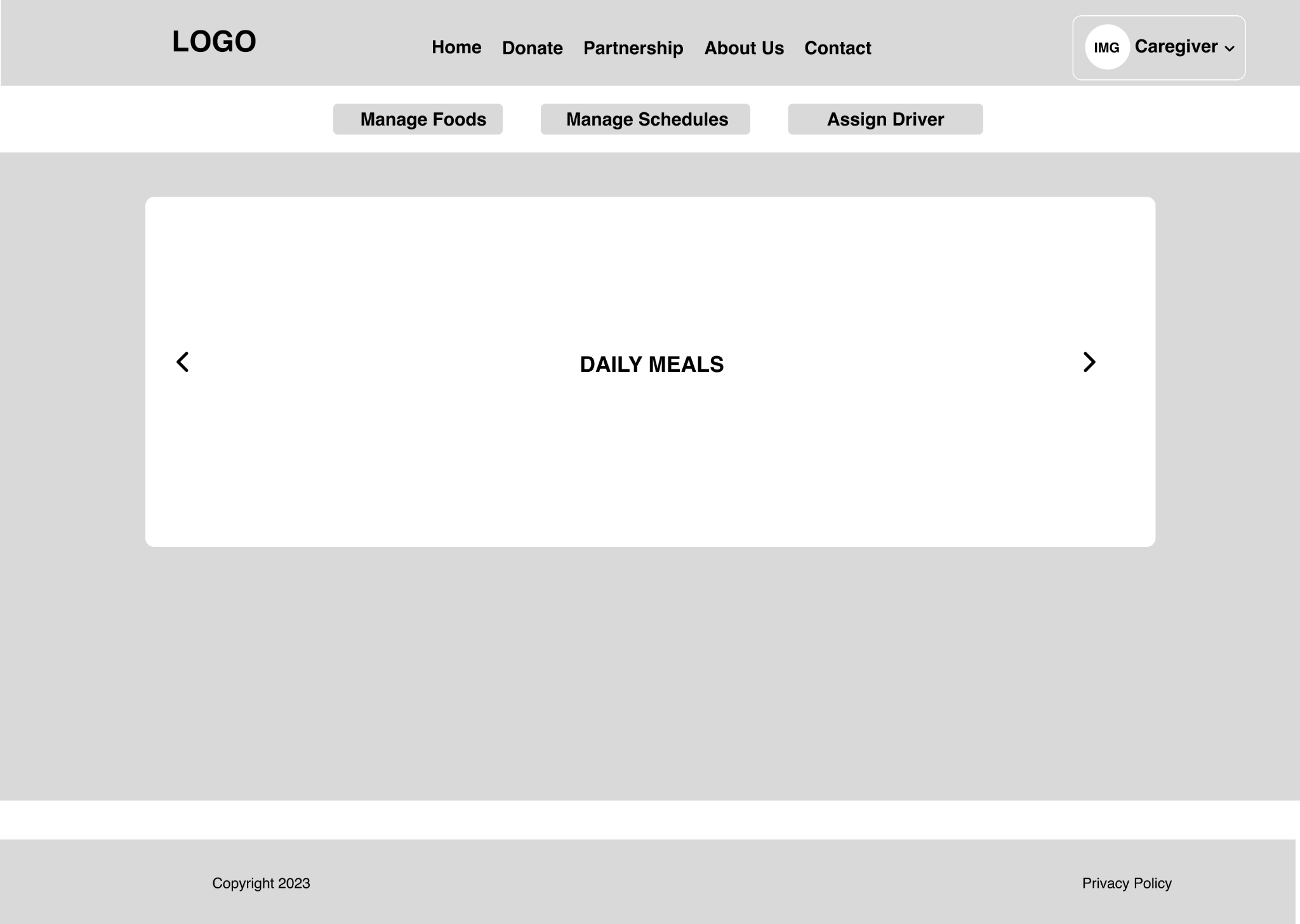
Member: Feedback & Evaluation Page



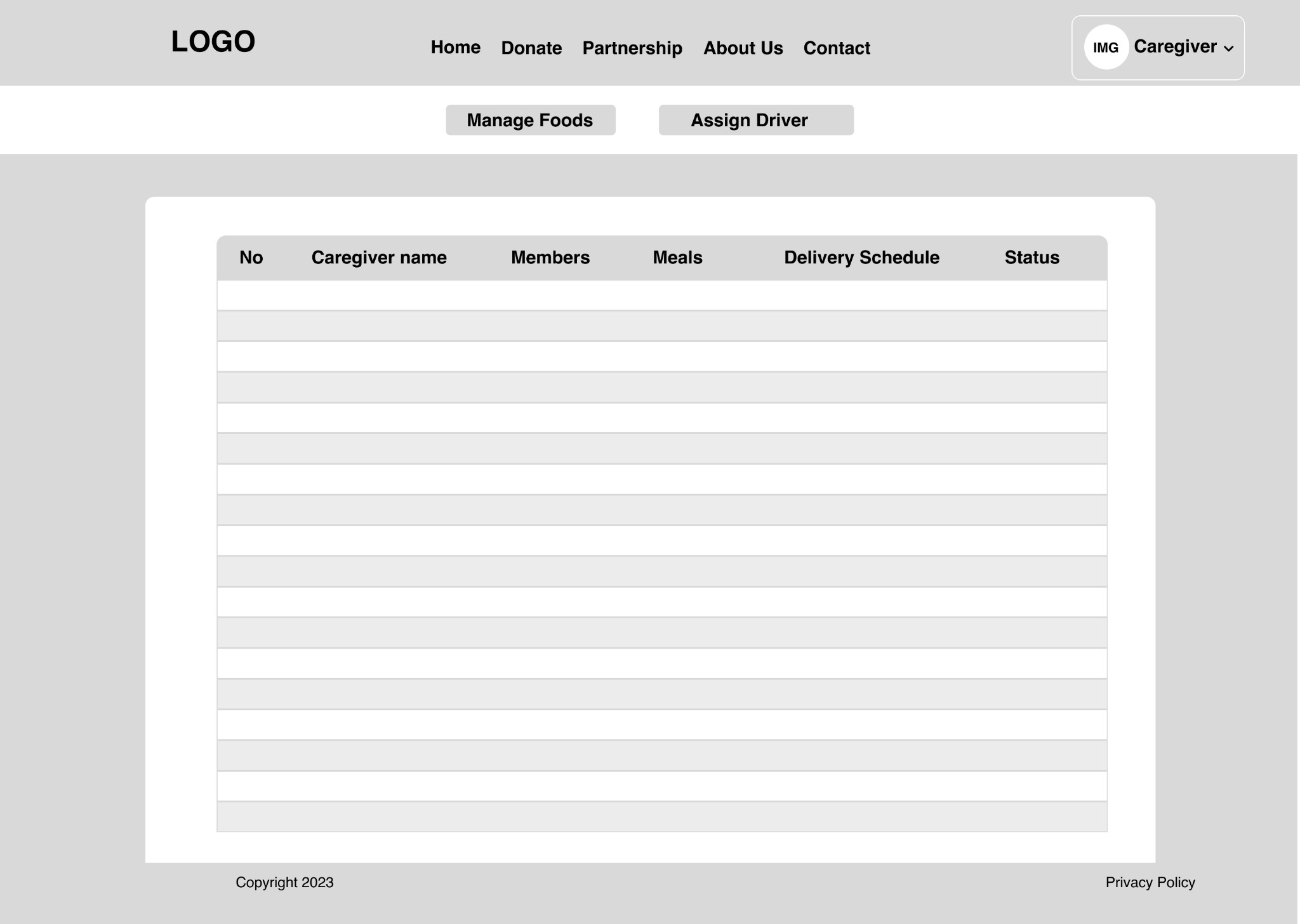
Driver: Homepage



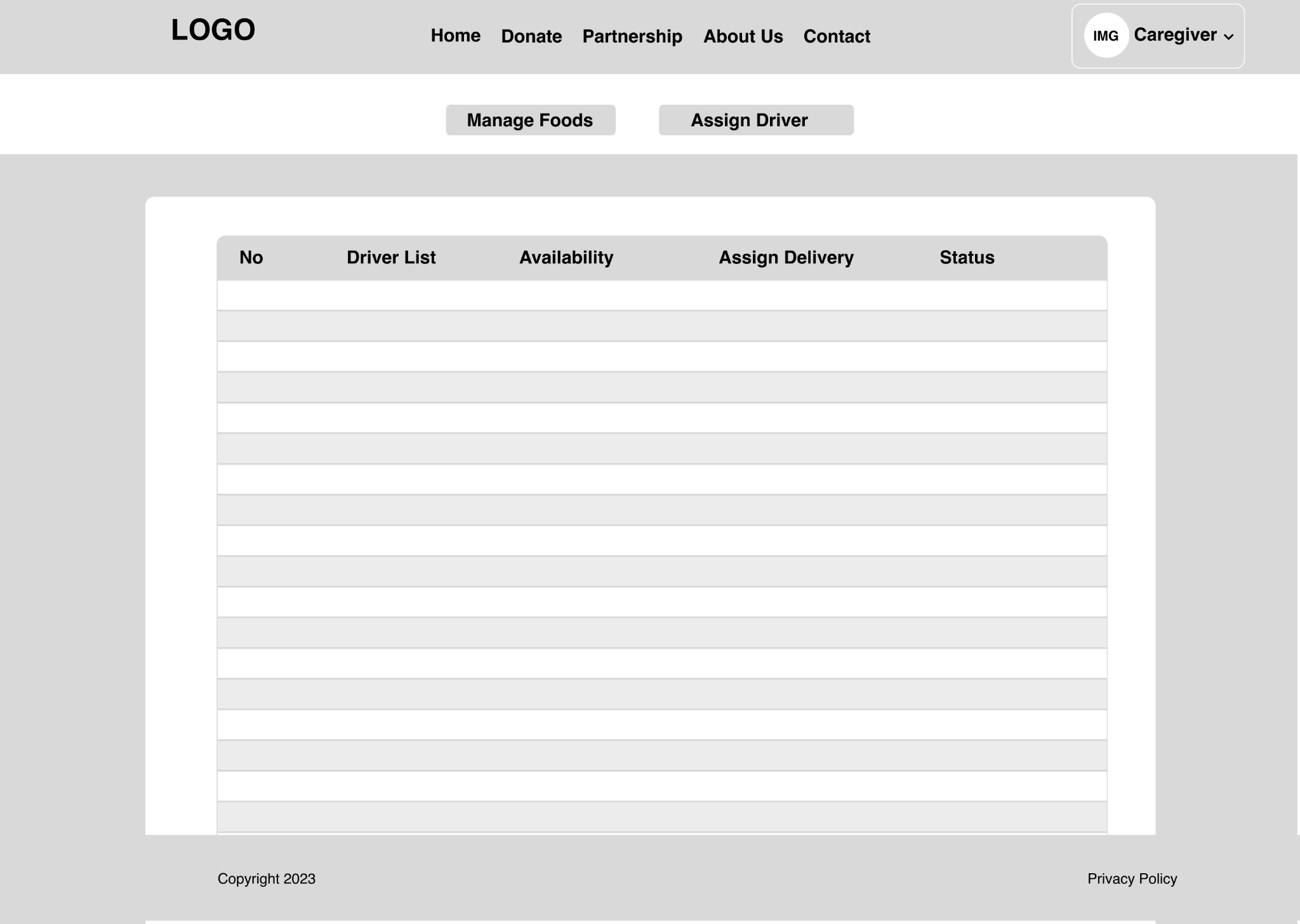
Caregiver: Homepage



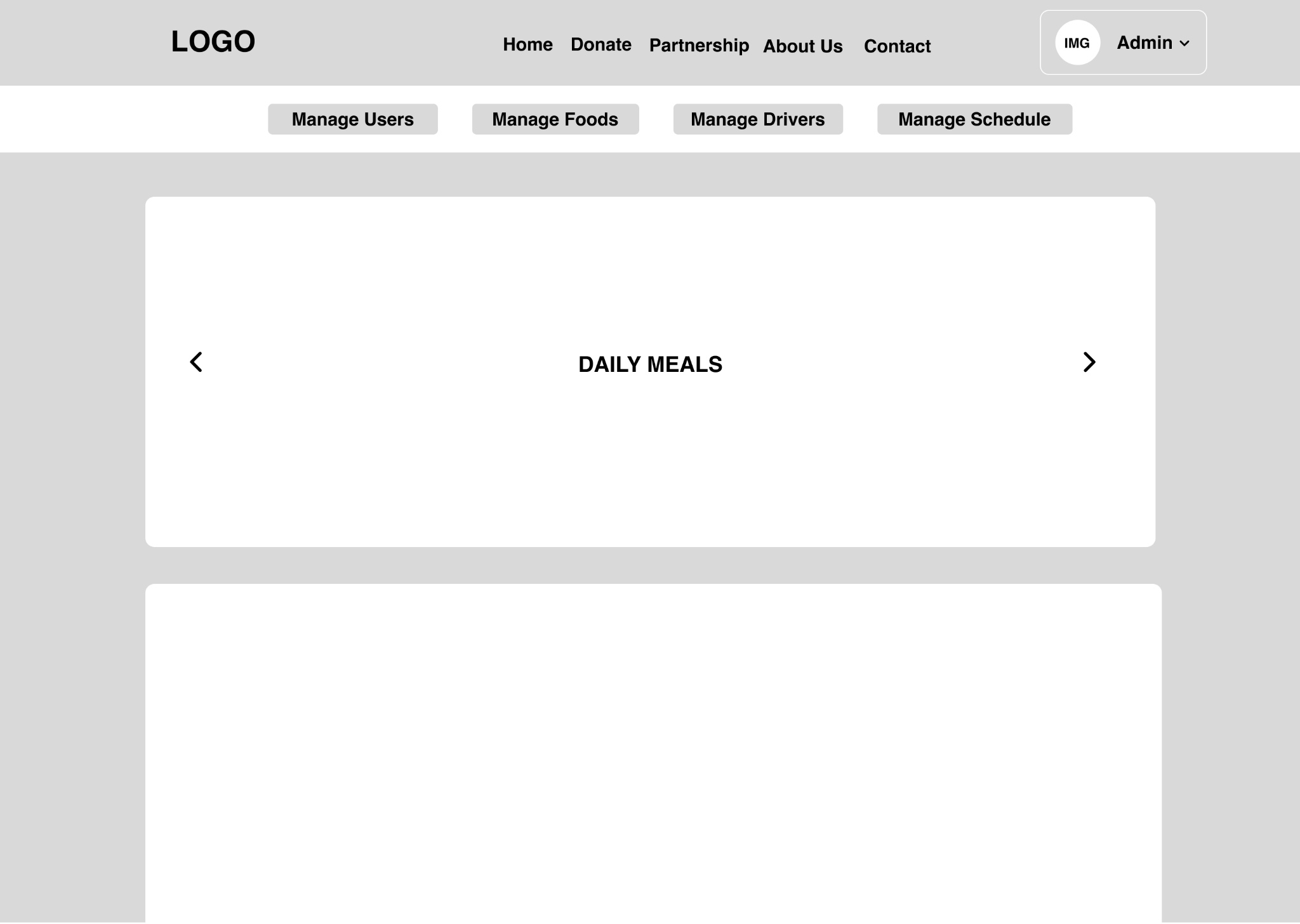
Caregiver: Manage Food Page



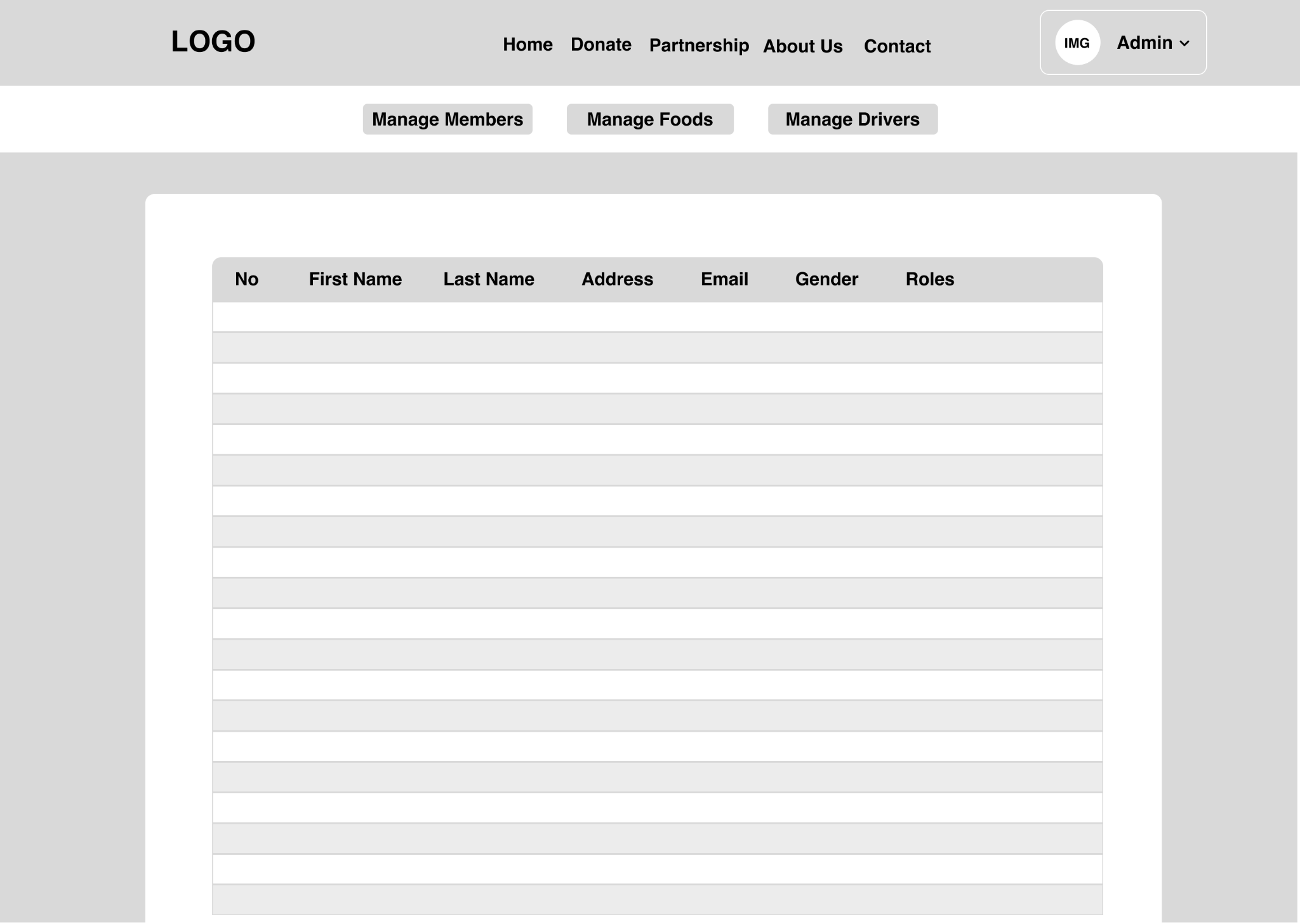
Caregiver: Assign Driver Page



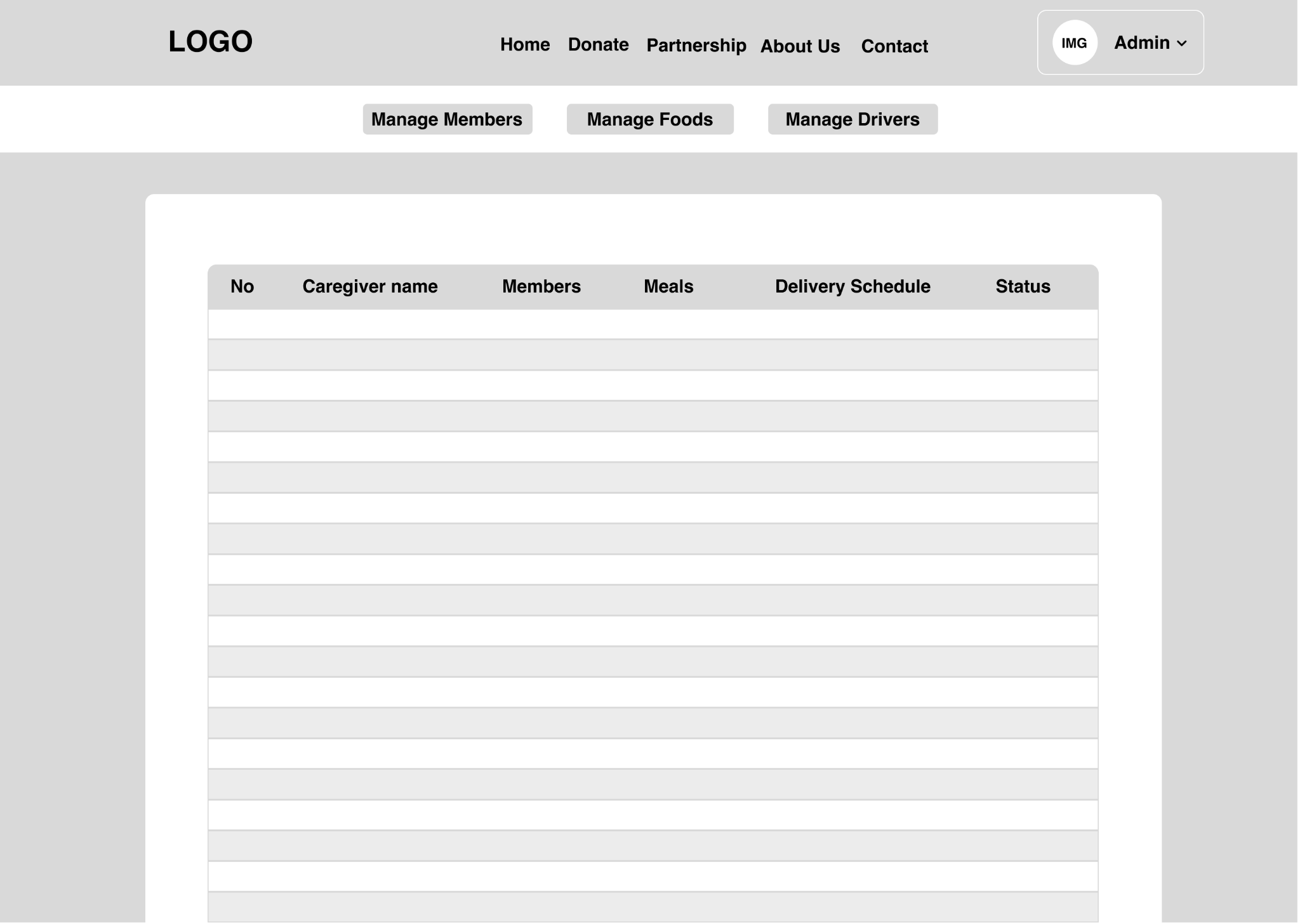
Admin: Homepage



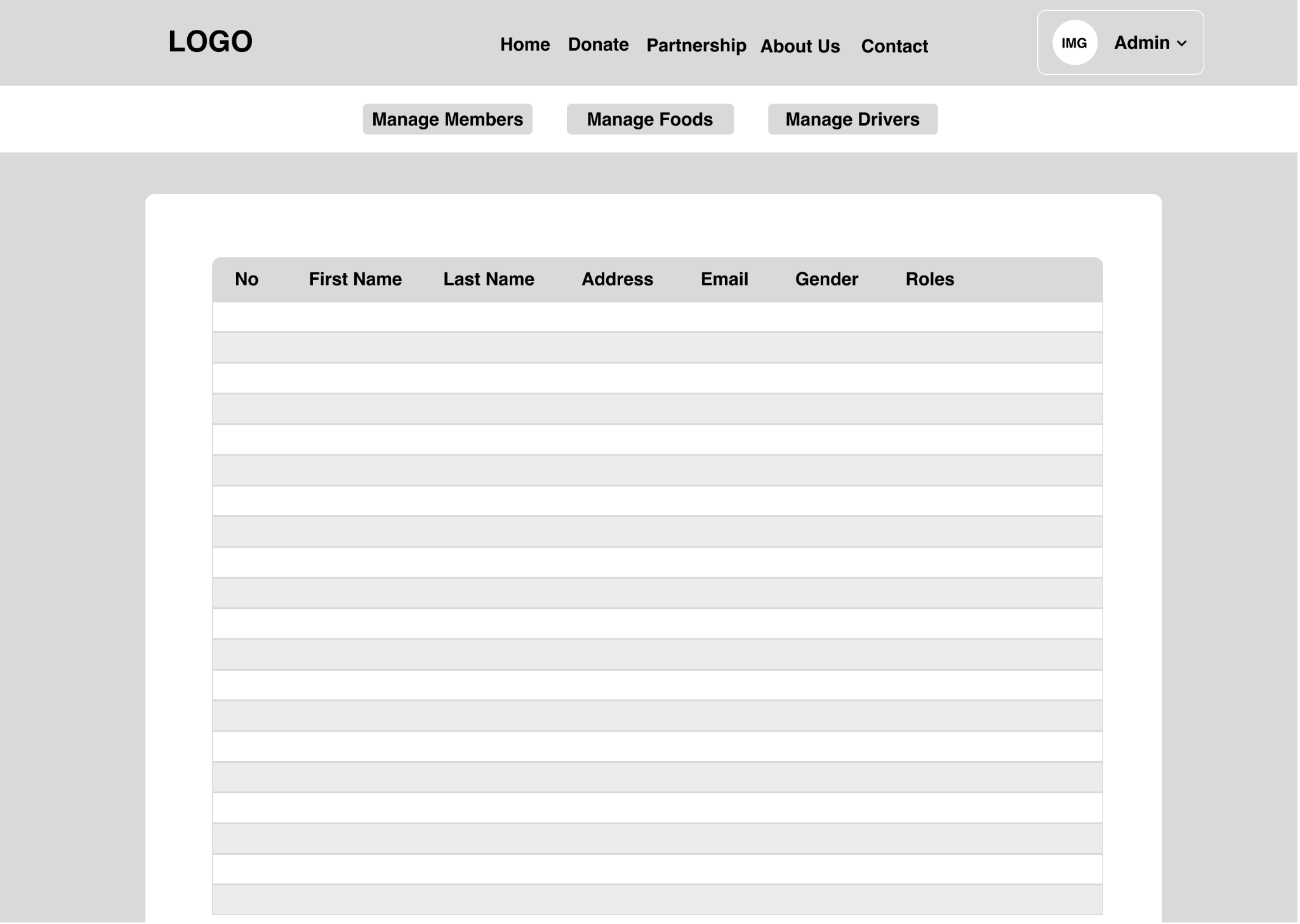
Admin: Manage Members Page



Admin: Manage Food Page

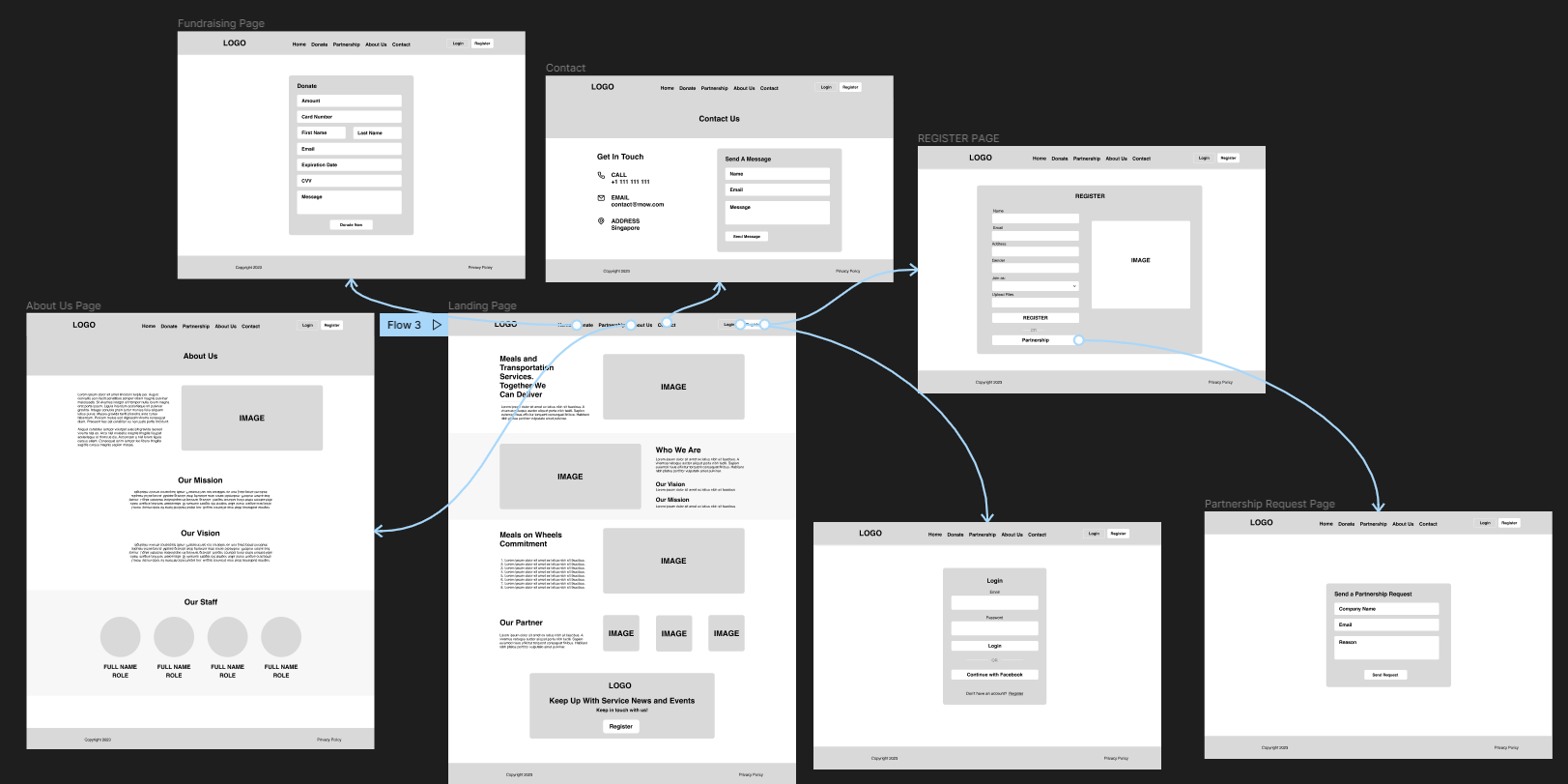


Admin: Manage Driver Page



## Module Storyboard

**All Users before login**



## Login as a:

## 

## DB Design:

## 

## 

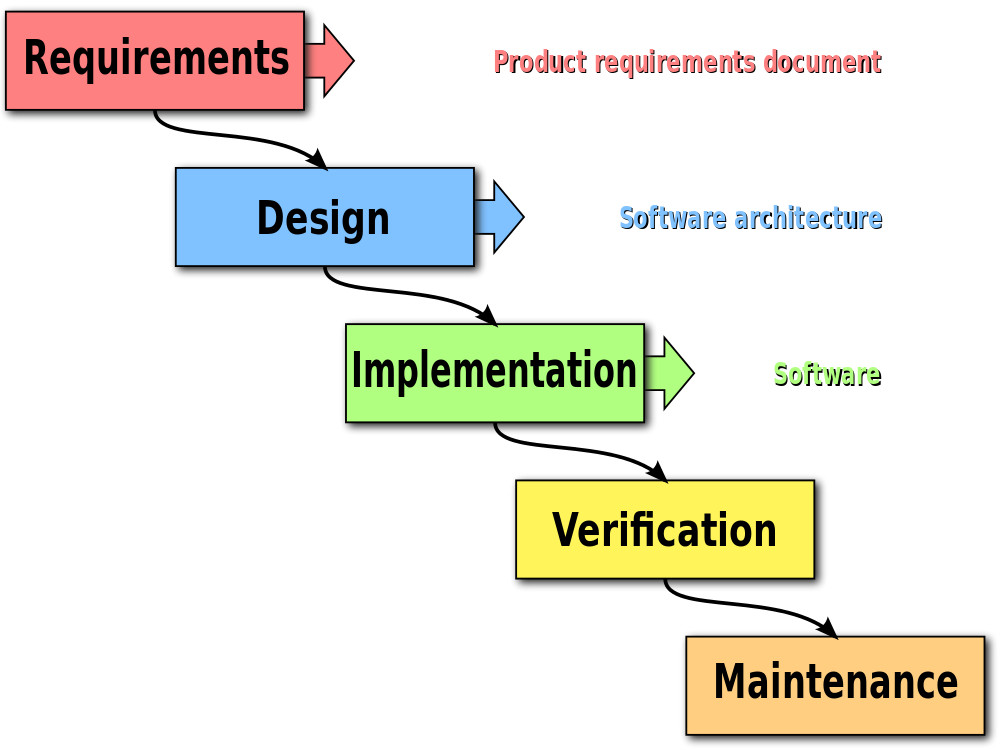
## Software Development Life Cycle

Software development life cycle or SDLC for short is a methodology for designing, building, and maintaining information and industrial systems. So far, there exist many SDLC models, such as the Waterfall model, which comprises five phases to be completed sequentially in order to develop a software solution; another model called the Spiral model, which is visualized as a process passing through some number of iterations.

### SDLC models

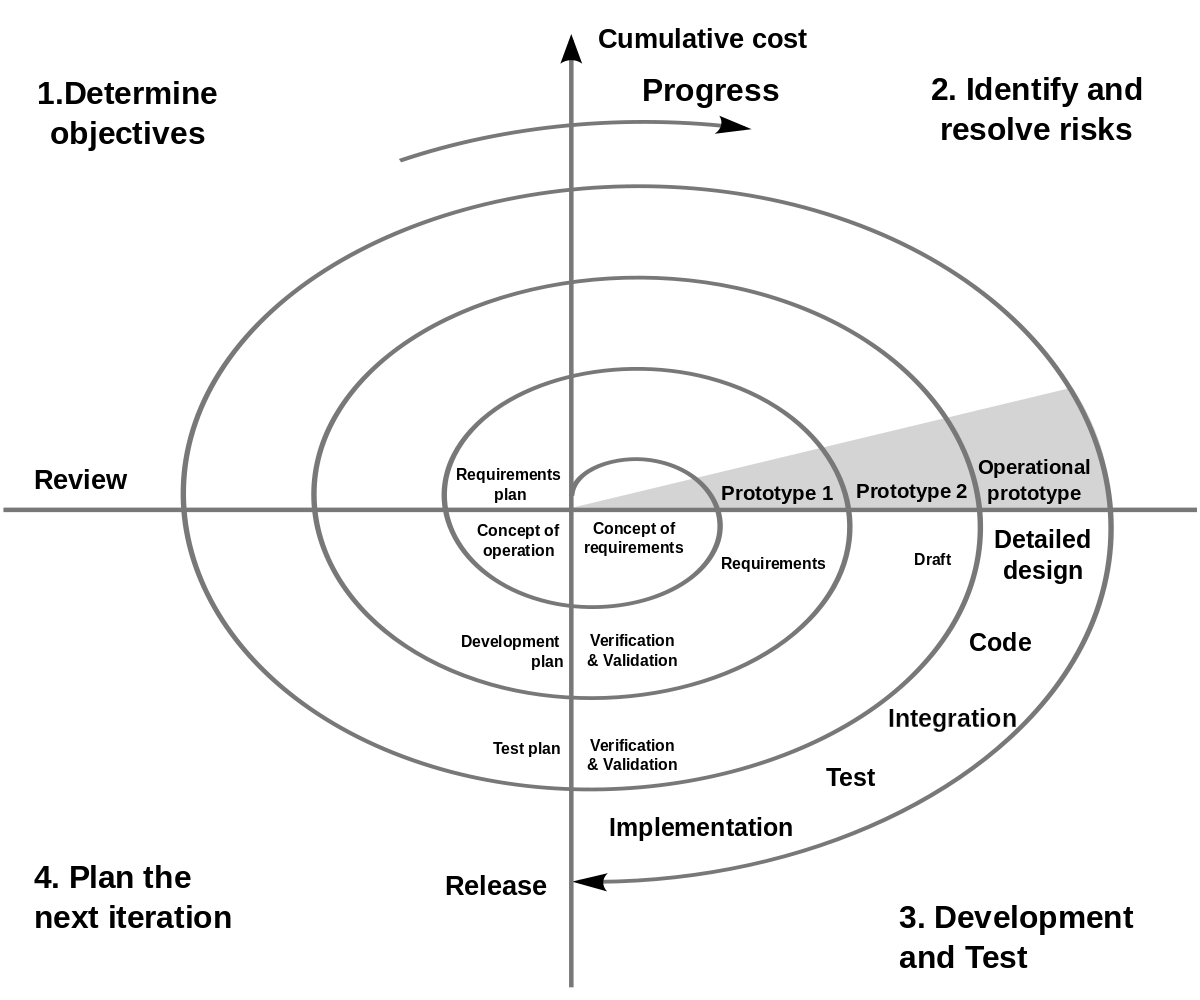
#### Waterfall Model

In the waterfall model, the whole process of Software Development is divided into phases where the output of one phase acts as the input to the next phase. The next phase begins only when the previous phase gets completed



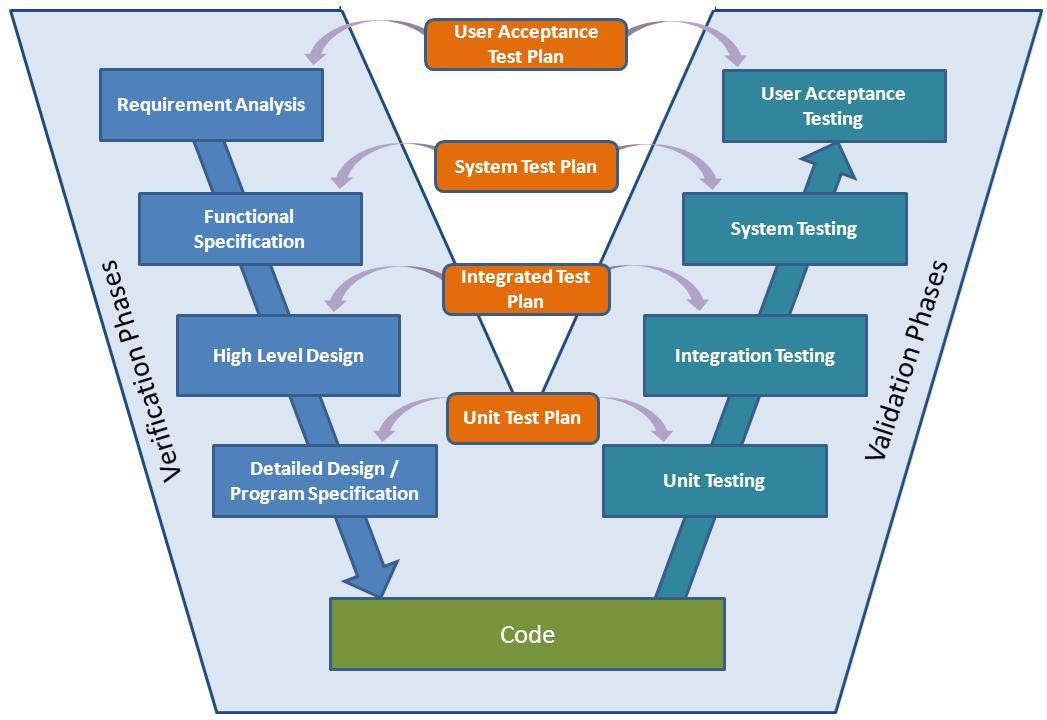
#### Spiral Model

This model is a combination of a Waterfall and Spiral model, and it works in an iterative manner. Based on the risk involved in the project, this model guides the team to adopt elements of one or more SDLC models such as a waterfall or Iterative model. Here the lifecycle of Software is divided into smaller parts, and new functionality can be added to the software even at the late stages of SDLC.



#### V-Model

V model is basically an expansion to the waterfall model where the testing and the development phases are planned in parallel. One side consists of the verification phase, while the other one consists of the validation phase, which is finally joined by coding. The next state starts only when the previous state gets completed.



#### Agile

Agile methodology is a practice which promotes continues interaction of development and testing during the SDLC process of any project. In the Agile method, the entire project is divided into small incremental builds. All of these builds are provided in iterations, and each iteration lasts from one to three weeks.



### SDLC Comparison

#### Waterfall

|  |  |
| --- | --- |
| **Strengths** | **Weaknesses** |
| * Simple to use and understand * Management simplicity thanks to its rigidity: every phase has a defined result and process review. * Development stages go one by one * Perfect for the small or mid-sized projects where requirements are clear and not equivocal. * Easy to determine the key points in the development cycle. * Easy to classify and prioritize tasks. | * The software is ready only after the last stage is over. * High risks and uncertainty. * Not the best choice for complex and object-oriented projects. * Inappropriate for the long-term projects * The progress of the stage is hard to measure while it is still in the development * Integration is done at the very end, which does not give the option of identifying the problem in advance. |

#### Spiral

|  |  |
| --- | --- |
| **Strengths** | **Weaknesses** |
| * Lifecycle is divided into small parts, and if the risk concentration is higher, the phase can be finished earlier to address the treats. * The development process is precisely documented yet scalable to the changes. * The scalability allows to make changes and add new functionality even at the relatively late stages. * The earlier working prototype is done, sooner users can point out the flaws. | * Can be quite expensive. * The risk control demands involvement of the highly skilled professionals. * Can be ineffective for the small projects. * Big number of the intermediate stages requires excessive documentation. |

#### V-Model

|  |  |
| --- | --- |
| **Strengths** | **Weaknesses** |
| * Every stage of V-shaped model has strict results so it’s easy to control. * Testing and verification take place in the early stages. * Good for the small projects, where requirements are static and clear. | * Lack of the flexibility. * Bad choice for the small projects. * Relatively big risks. |

#### Agile

|  |  |
| --- | --- |
| **Strengths** | **Weaknesses** |
| * Corrections of functional requirements are implemented into the development process to provide the competitiveness. * Project is divided by short and transparent iterations. * Risks are minimized thanks to the flexible change process. * Fast release of the first product version. | * Difficulties with measuring the final cost because of permanent changes. * The team should be highly professional and client oriented. * New requirements may conflict with the existing architecture. * With all the corrections and changes there is possibility that the project will exceed expected time. |

#### When to use

|  |  |
| --- | --- |
| **Model** | **When to Use** |
| Waterfall | * The requirements are precisely documented. * Product definition is stable. * The technologies stack is predefined, which makes it not dynamic. * No ambiguous requirements * The project is short |
| Spiral | * The customer isn’t sure about the requirements. * Significant edits are expected during the software development life cycle. * Risk management is highly essential for the project. |
| V Model | * For the projects where accurate product testing is required. * For the small and mid-sized projects, where requirements are strictly predefined. * The engineers of the required qualification, especially testers, are within easy reach. |
| Agile | * The users’ needs change dynamically. * Less price for the changes implemented because of the numerous iterations. * It requires only initial planning to start the project. |

### SDLC Chosen Method

**Waterfall Model**

Because in the waterfall model, the whole process is divided into phases where the output of one phase acts as the input to the next phase. It makes waterfall simple to use and we can easily classify and prioritize tasks and easily determine the key points in the development cycle.

Management simplicity thanks to its rigidity: every phase has a defined result and process review, development stages go one by one, and this model is perfect for the small or mid-sized projects where requirements are clear and not equivocal, also refer to table above waterfall is worth to use.

## Front-End Framework:

### AngularJs

AngularJS is not a single piece in the overall puzzle of building the client-side of a web application. It handles all of the DOM and AJAX glue code you once wrote by hand and puts it in a well-defined structure. This makes AngularJS opinionated about how a CRUD (Create, Read, Update, Delete) application should be built. But while it is opinionated, it also tries to make sure that its opinion is just a starting point you can easily change. AngularJS comes with the following out-of-the-box:

* Everything you need to build a CRUD app in a cohesive set: Data-binding, basic templating directives, form validation, routing, deep-linking, reusable components and dependency injection.
* Testability story: Unit-testing, end-to-end testing, mocks and test harnesses.
* Seed application with directory layout and test scripts as a starting point.

### ReactJS

React is a life-changing frontend framework that comes to the rescue of many UI developers. Using React for front-end development, you get access to the community of more than 56,000 developers, and at least 8,787 industry leaders use the popular JavaScript library.

React works with a virtual DOM. It doesn’t need to refresh all the components on a page whenever the user interacts with it. This gives React the edge for dynamic websites which suffer a drop in performance with the conventional DOM refresh.

There’s also no coding nightmare with React frontend framework. It’s built with reusability in mind, and developers can build new features without rewriting existing codes. UI components built with React are also easy to debug.

With all these benefits, It didn’t take long for React to cement itself as the preferred UI development tool amongst JavaScript developments. The highly-popular framework was downloaded 7 million times weekly.

React Features:

* SEO Friendly, the clean and simple code of React minimizes page load time and page load time is a critical factor in SEO.
* React JS apps are created as a series of components, with functionalities being passed from parent component to child component in the form of arguments. This is called one-way data binding or the unidirectional flow of data. Because of this one characteristic, it is very convenient to make changes in React apps.
* Since React JS is essentially a library, you can just plug the React code wherever you need it, irrespective of infrastructure. This will have absolutely no effect on performance.

### VueJS

VueJS is a framework and ecosystem that covers most of the common feature needed in frontend development. But the weeb is extremely diverse - the things we build on the web may vary drastically in form and scale. With that in mind, Vue is designed to be flexible and incrementally adoptable. Depending on your use case, Vue can be used in different way:

* Enhancing static HTML without a build step
* Embedding as Web Components on any page
* Single-Page Application (SPA)
* Fullstack / Server-Side Rendering (SSR)
* Jamstack / Static Site Generation (SSG)
* Targeting desktop, mobile, WebGL, and even the terminal

### 

### Comparison

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Angular** | **React** | **Vue** |
| Initial Release | 2016 | 2011 | 2014 |
| Support | Google | Facebook | Community |
| Type | Framework | Library | Framework |
| Size | Medium | Small | Very small |
| Language | TypeScript | JavaScript | JavaScript |
| Performance | Good | Good | Good |
| Data Binding | Both | Unidirectional | Bidirectional |
| Popular Websites | Paypal, Samsung, Upwork | Netflix, Twitter, Amazon | Alibaba, Grammarly, GitLab |

**Chosen Framework:**

**ReactJS**

* React offers an easy debugging process. The code is reusable.
* It’s easy to learn because of its easy and simple design.
* It has faster updates with both server-side and front-end support.
* React has great performance.
* Broad community support from github and stackoverflow

## 

## Back-end Framework:

### Spring boot

Java Spring Framework (Spring Framework) is a well-liked, open-source, enterprise-level framework for building standalone, high-quality Java Virtual Machine applications (JVM). It is built on top of the conventional spring framework. So, it provides all the features of spring and is yet easier to use than spring.

Through its three main features, Java Spring Boot (Spring Boot) is a tool that accelerates and simplifies the development of web applications and microservices with the Spring Framework.

* **Autoconfiguration**

Applications are initialized with pre-set dependencies through autoconfiguration, so you don't have to manually configure them.

* **An opinionated approach to configuration**

Based on the requirements of our project, Spring Boot takes an opinionated approach to adding and configuring starter dependencies. Instead of requiring you to make all those decisions and configure everything manually, Spring Boot decides which packages to install and which default values to use.

* **The ability to create standalone applications**

Spring Boot enables programmers to build ready-to-use applications. By integrating a web server like Tomcat or Netty into your app during the initialization process, it specifically enables you to create standalone applications that run independently, without relying on an external web server. As a result, by simply selecting the Run command, you can run your application on any platform. To create applications without an embedded Web server, you can disable this feature.

## 

## Database:

### MongodB

MongoDB is built on a scale-out architecture that has become popular with developers of all kinds for developing scalable applications with evolving data schemas.

As a document database, MongoDB makes it easy for developers to store structured or unstructured data. It uses a JSON-like format to store documents. This format directly maps to native objects in most modern programming languages, making it a natural choice for developers, as they don’t need to think about normalizing data. MongoDB can also handle high volume and can scale both vertically and horizontally to accommodate large data loads.

MongoDB was built for people building internet and business applications who need to evolve quickly and scale elegantly. Companies and development teams of all sizes use MongoDB for a wide variety of reasons.

### MySQL

The most widely used open-source database worldwide is MySQL. MySQL is the second-most popular database, after Oracle Database, according to DB-Engines. Numerous of the most popular apps, such as Facebook, Twitter, Netflix, Uber, Airbnb, Shopify, and Booking.com, are powered by MySQL.

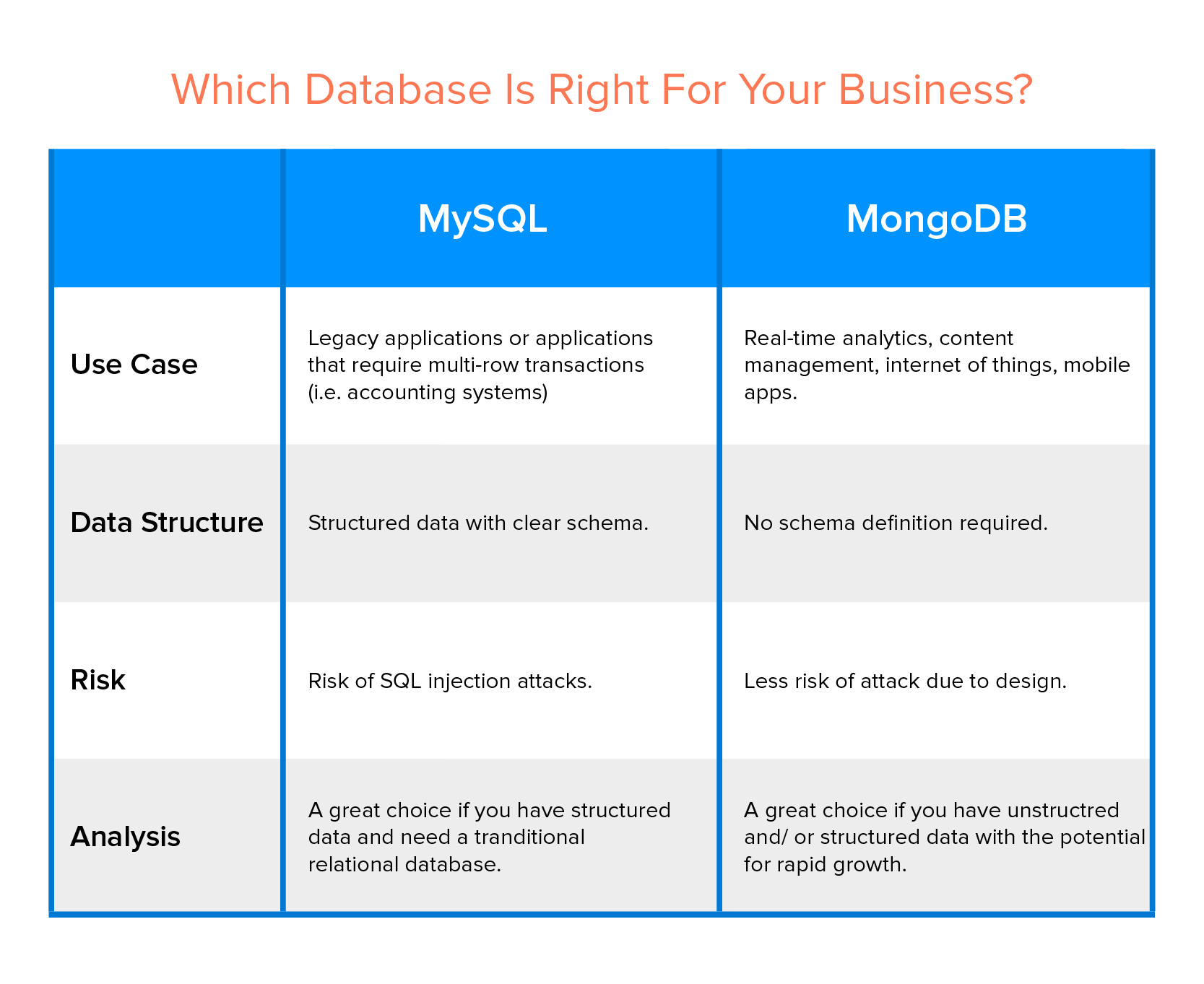
Because MySQL is open source, it has many features that have been created over more than 25 years in close collaboration with users. Therefore, it is very likely that MySQL Database supports your preferred application or programming language.

**MySQL advantages**

MySQL is quick, trustworthy, scalable, and simple to use. It was initially created to quickly handle large databases and has long been used in extremely demanding production environments.

Despite being constantly improved, MySQL offers a comprehensive and practical set of functions. Accessing databases on the internet is a task that MySQL is well suited for due to its connectivity, speed, and security.

### Comparison



**Chosen database:**

**MySQL**

* Open-source and compatible, it is highly compatible with a wide range of systems, programming languages and database models.
* Fast and reliable
* Easy to use and administer
* Large online support and documentation with [dev.mysql.com](https://dev.mysql.com/)

### 

## Integrated Development Environment:

### IntelliJ

IntelliJ IDEA is undoubtedly the top-choice IDE for software developers. Efficiency and intelligence are built into the design, which enables a very smooth development workflow experience, from design, implementation, building, deploying, testing, and debugging, to refactoring! It is loaded with features and also offers a plethora of plugins that we can integrate into the editor. I switched to using IntelliJ IDEA 5 years ago and have never looked back. It has certainly made my life easier. I am producing more with less effort (Mary Grygleski).

### Eclipse

Released by IBM in 2001, the open-source IDE Eclipse was initially created to eclipse Microsoft’s Visual Studio. Since then, Eclipse has become a major platform used in both open source and commercial projects. Highly versatile and flexible, Eclipse was designed to serve the needs of complex enterprise projects and embedded system application development. It offers support for a variety of programming languages in addition to Java, as well as popular frameworks. With a plethora of plugins, thorough documentation and a large developer community, Eclipse has rightfully gained a loyal following.

### STS

STS is an Eclipse-based development environment that is customized for the development of Spring applications. It provides a ready-to-use environment to implement, debug, run and deploy your applications. It also includes integration for Pivotal tc Server, Pivotal Cloud Foundry, Git, Maven and AspectJ. STS is built as an addition on top of the latest Eclipse releases.

**Features Overview:**

Eclipse STS validates your project and provides quick fixes for your applications. For example, when working with Spring Data JPA, the IDE may be used to validate query method names

STS also provides a graphical view on all bean methods and their mutual relationships. You may want to have a closer look at the graphical editors that come with STS by looking into the views that are available under the menus window, show view and then Spring respectively. STS also offers other additional useful features that are not limited to Spring applications only.

**Chosen IDE:**

**STS**

* The Spring Tool Suite understands your Spring projects. It parses your configuration files and displays detailed information about the beans that are being defined, their dependencies among each other, used namespaces, and extracts overviews for certain stereotypes like request controllers, aspects, services, and more.
* Because the Spring Tool Suite understands your Spring projects, it provides a comprehensive set of validations that are being applied automatically. Those validations indicate errors in your configurations directly within the IDE, long before you actually run the app. Finding problems and misconfigurations gets a lot easier.
* Refactoring support is one of the most important parts of todays software engineering. Therefore the Spring Tool Suite provides advanced support for refactoring Spring applications. Not only the well-known Java refactorings are reflected in your Spring config files, the IDE adds new refactorings for Spring elements (like renaming of Spring beans, for example).
* It doesn't matter whether you are writing Spring XML configuration files or implement JavaConfig Spring apps, whether you are using the core Spring framework alone or together with all the various additional Spring projects, the Spring Tool Suite provides you with meaningful content-assist all over the place, together with quick-fixes for common errors and problems.

**Code Editor:**

**Visual Studio Code**

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages and runtimes (such as C++, C#, Java, Python, PHP, Go, .NET). We will be using Visual Studio Code for editing our React JS source code.

## Bibliography

“Agile Model (Software Engineering) - javatpoint.” *Javatpoint*, <https://www.javatpoint.com/software-engineering-agile-model>. Accessed 25 January 2023.

“AngularJS: Developer Guide: Introduction.” *AngularJS: API*, [https://docs.angularjs.org/guide/introduction#](https://docs.angularjs.org/guide/introduction) . Accessed 25 January 2023.

Castro, Santiago. “5 Reasons Why MySQL Is Still The Go-To Database Management System.” *Jobsity*, <https://www.jobsity.com/blog/5-reasons-why-mysql-is-still-the-go-to-database-management-system> . Accessed 25 January 2023.

“Difference between Spring and Spring Boot.” *GeeksforGeeks*, 22 December 2022, <https://www.geeksforgeeks.org/difference-between-spring-and-spring-boot/> . Accessed 25 January 2023.

“Documentation for Visual Studio Code.” *Visual Studio Code*, <https://code.visualstudio.com/docs> . Accessed 25 January 2023.

Galik, Olga. “When and Why You Should Use React.” *Uptech*, <https://www.uptech.team/blog/why-use-react> . Accessed 25 January 2023.

“A Guide to Spring in Eclipse STS.” *Baeldung*, 16 April 2022, <https://www.baeldung.com/eclipse-sts-spring> . Accessed 25 January 2023.

“A Guide to Spring in Eclipse STS.” *Baeldung*, 16 April 2022, <https://www.baeldung.com/eclipse-sts-spring> . Accessed 25 January 2023.

“IntelliJ IDEA vs Eclipse: Which Is Better for Beginners.” *Tabnine*, 18 March 2019, <https://www.tabnine.com/blog/intellij-idea-vs-eclipse/> . Accessed 25 January 2023.

“Introduction.” *Vue.js*, https://vuejs.org/guide/introduction.html#the-progressive-framework. Accessed 25 January 2023.

Joshi, Mohit. “Angular vs React vs Vue: Core Differences.” *BrowserStack*, 23 December 2022, <https://www.browserstack.com/guide/angular-vs-react-vs-vue> . Accessed 25 January 2023.

“Memahami System Development Life Cycle – Accounting.” *Accounting*, 19 May 2020, <https://accounting.binus.ac.id/2020/05/19/memahami-system-development-life-cycle/> . Accessed 25 January 2023.

Patadiya, Jaydeep. “Angular vs React | Angular vs Vue | React vs Vue - Know the Difference.” *Radixweb*, 12 September 2022, <https://radixweb.com/blog/angular-vs-react-vs-vue> . Accessed 25 January 2023.

Pedamkar, Priya. “What is SDLC | Different Phases and Models of SDLC.” *eduCBA*, <https://www.educba.com/what-is-sdlc/> . Accessed 25 January 2023.

Raval, Nihar. “React vs Angular: Which One is Best for Your Next Front-end Project?” *Radixweb*, 18 October 2022, <https://radixweb.com/blog/react-vs-angular> . Accessed 25 January 2023.

“SDLC Models: Agile, Waterfall, V-Shaped, Iterative, Spiral.” *Existek*, 4 August 2017, <https://existek.com/blog/sdlc-models/#types> . Accessed 25 January 2023.

“SDLC Models - javatpoint.” *Javatpoint*, <https://www.javatpoint.com/software-engineering-sdlc-models> . Accessed 25 January 2023.

“What Developers Say About Us.” *JetBrains*, <https://www.jetbrains.com/company/customers/testimonials.html#product=idea> . Accessed 25 January 2023.

“What is Java Spring Boot?” *IBM*, <https://www.ibm.com/topics/java-spring-boot> . Accessed 25 January 2023.

“What is Java Spring Boot?” *IBM*, <https://www.ibm.com/topics/java-spring-boot> . Accessed 25 January 2023.

“What Is MySQL?” *Oracle*, <https://www.oracle.com/mysql/what-is-mysql/> . Accessed 25 January 2023.