Questions

What are the components that are absolutely necessary to build any kind of system no matter the requirement or the scale or the users?

For a very basic system, we need the following components for sure.

Database - Persistence is something that is required for every backend application. So, A Database is utmost necessary
Server - For any system that needs to scale, we need to have a server, so that we can have an API driven architecture so that if the frontend changes, the new frontend can easily consume the existing API's.
Network - We need a Network as a component for the communication between the client and the server and between the server and the database.

Not necessary

Components like a frontend are not necessary because even a console will work or even a curl command will work.
Stuff like Load Balancers, Caches are there for efficient and highly scalable systems, they are not required for basic applications.

what load balancer do if one server fails?

If one server fails, a load balancer will typically detect the failure through a health check mechanism that periodically pings the server to ensure that it is still responding. Once the failure is detected, the load balancer will stop forwarding traffic to the failed server and redirect it to the remaining healthy servers.
There are several ways that a load balancer can handle a failed server, depending on its configuration and the specific implementation. Here are some common methods:

- Round-robin: Load balancer will simply skip over the failed server and forward requests to the next available server in the rotation.
- Weighted distribution: Load balancer can be configured to adjust the distribution of traffic to each server based on their capacity or health status. For example, if one server fails, the load balancer can increase the traffic to the remaining servers to compensate for the loss of capacity.
- Hot standby: In a hot standby configuration, the load balancer will maintain a backup server that is ready to take over if one of the active servers fails. The backup server can be configured to take over automatically, or the load balancer administrator can manually initiate the failover.

In general, the goal of a load balancer is to ensure that traffic is distributed evenly and efficiently across multiple servers.