

# WeShare

## Data Transmission Project

2023

Made by

Oprea Sergiu Daniel  
Pleşca Evelyn Iulia  
Tăslăuan Alexandra



# Table of contents

**01**

**Introduction**

**02**

**Goals**

**03**

**Functions**

**04**

**Application Design**

**05**

**Application  
Implementation**

**06**

**Conclusion**

# Introduction

Today, especially since Covid-19, the trend is for people to go online and spend as much time there. Even though that might not be the best solution in some cases, in other cases is the only solution: when we are far from home, far from family and friends and we want to keep in touch with them.

We have created a social media platform designed to connect people, foster collaboration, and inspire meaningful interactions.



# Goals

The main goal of our application was to create a digital space that promotes positive engagements, enables authentic connections, and empowers users to make a difference in their communities and beyond.



# Functions

One of the features we decided it was crucial working with was the database..

Second of, we conceptualized this application to be as easy as possible to use and to have a great user experience, having the following features:



# Functions

## Login, Logout and Registration pages:

- we want to keep your experience tailored to your interests by having a safe account.

## Comment on posts:

- don't be scared to comment on posts and let people know how great it is seeing them.



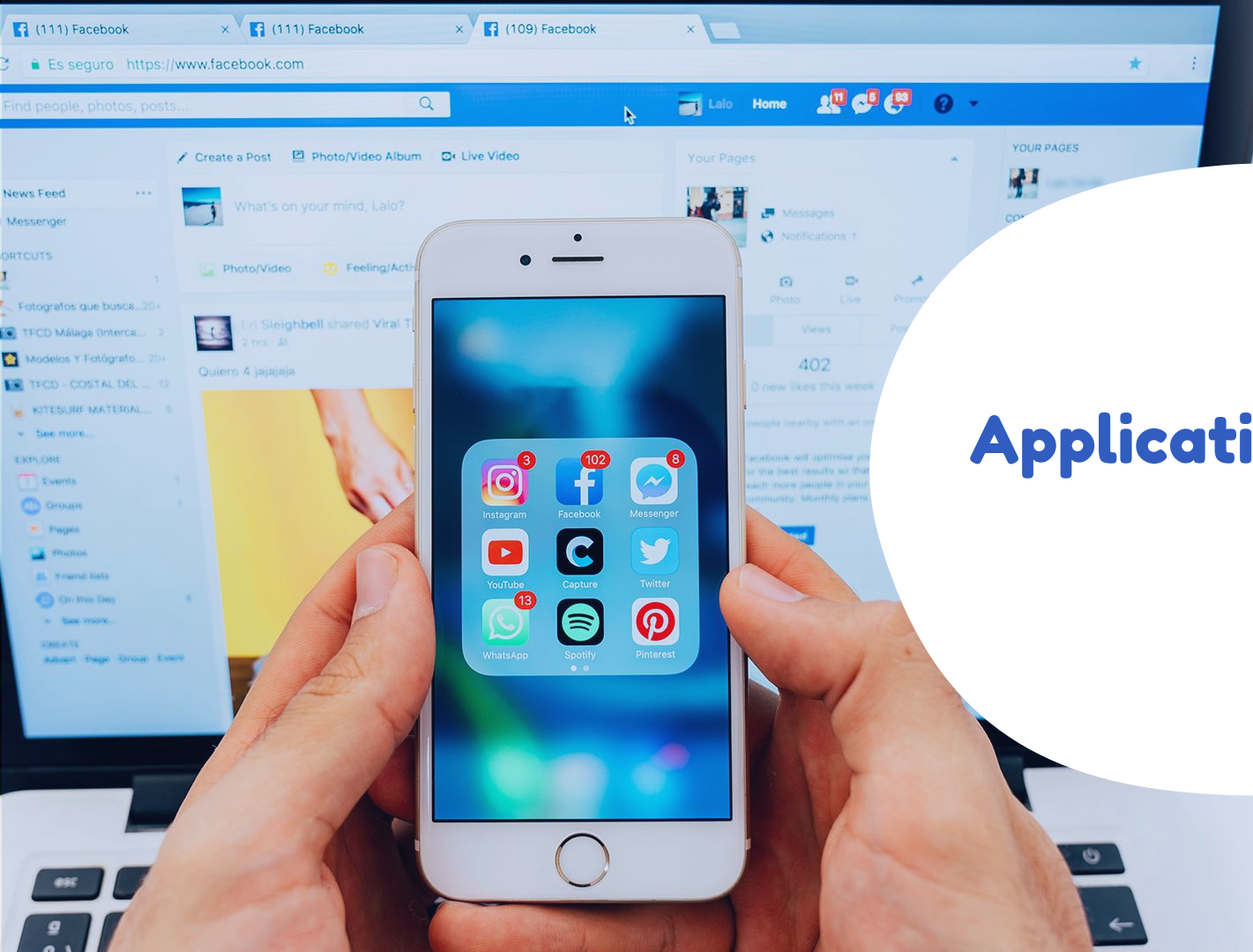
## Change profile and cover pictures:

- let other people see you by changing these representative images of yourself whenever you want to.



## Like other people's posts:

- let people know how much you appreciate seeing their posts.

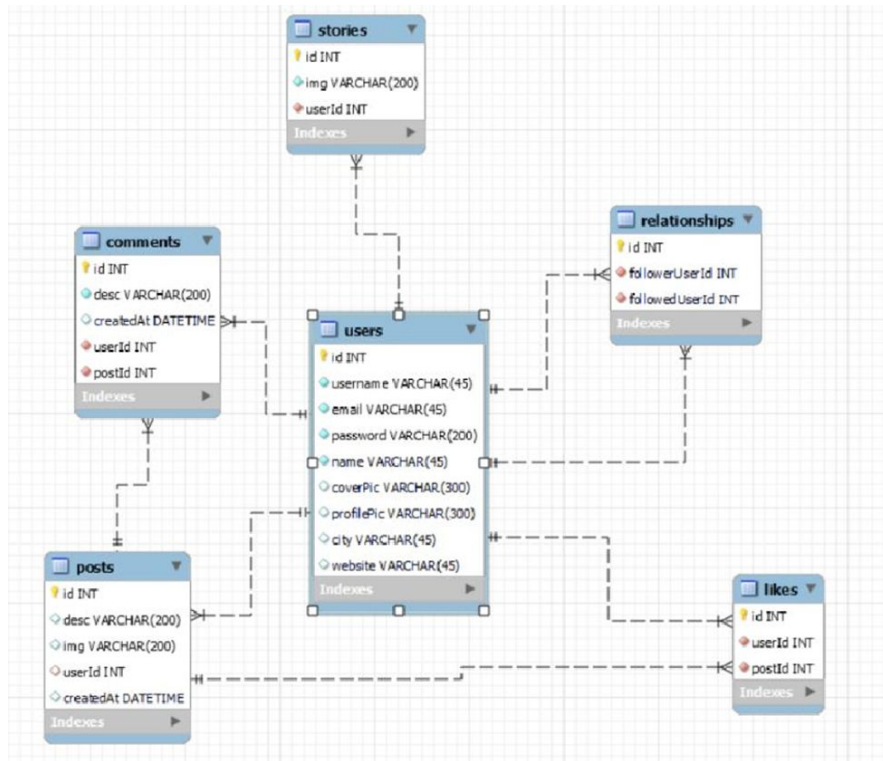


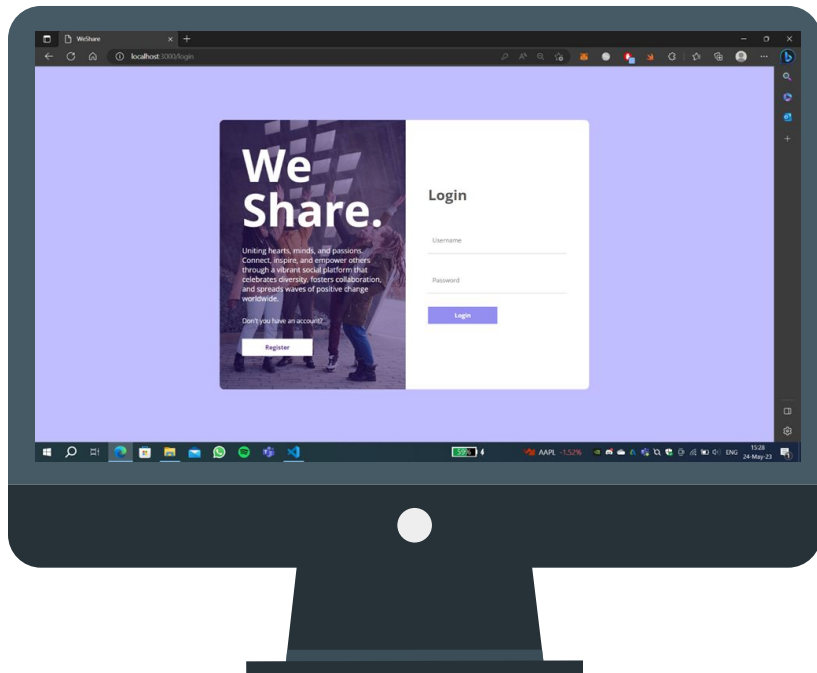
# Application Design

# Database Design

Table	Columns	Properties
Comments	id [PK] desc createdAt userId postId	int varchar(200) datetime int int
Likes	id userId postId	int int int
Posts	id desc img userId createdAt	int varchar(200) varchar(200) int datetime



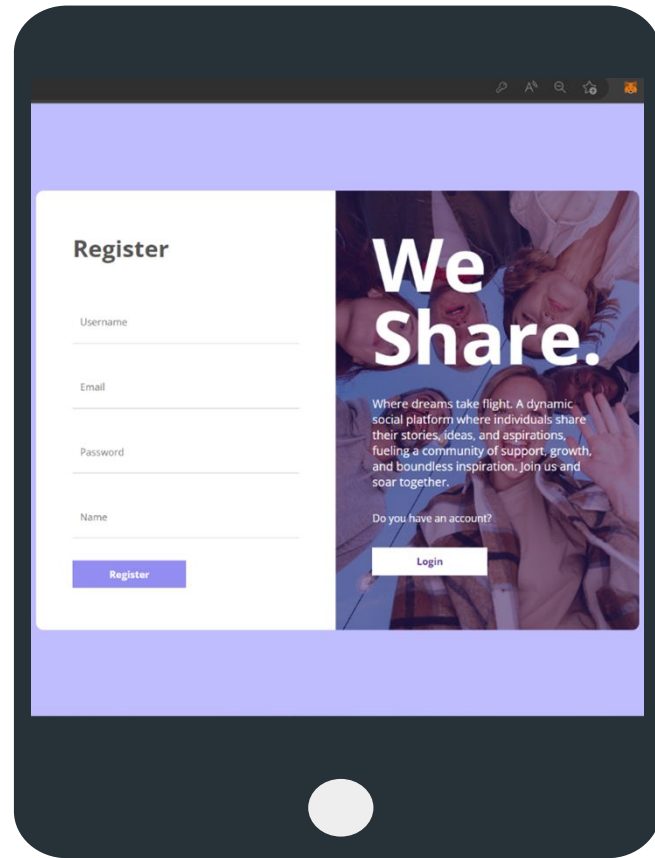




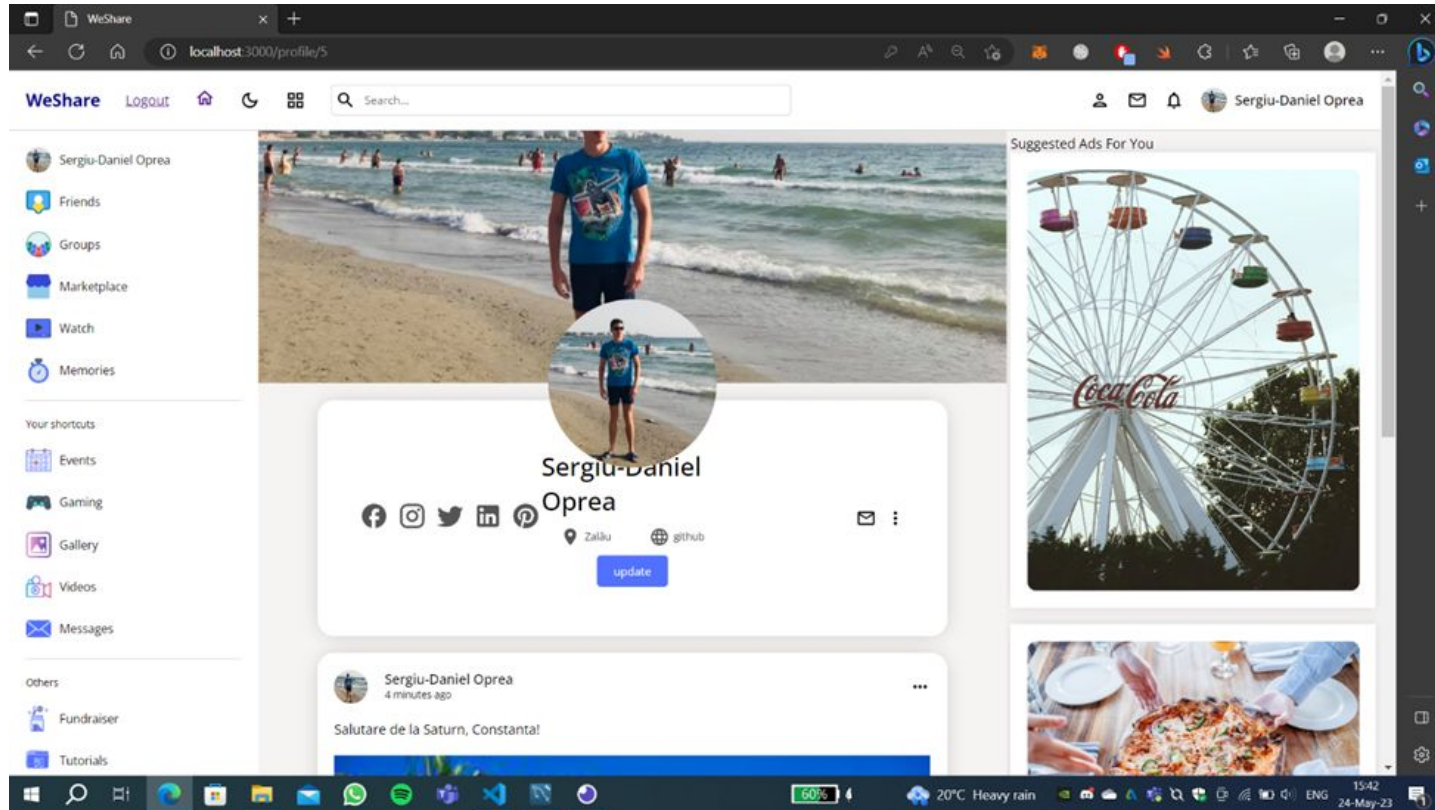
# GUI Design

Login Page

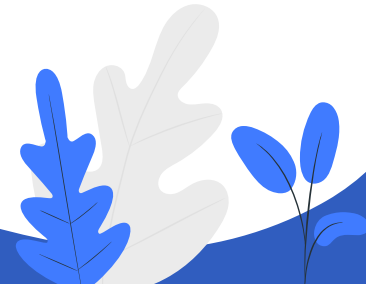
# Register Page



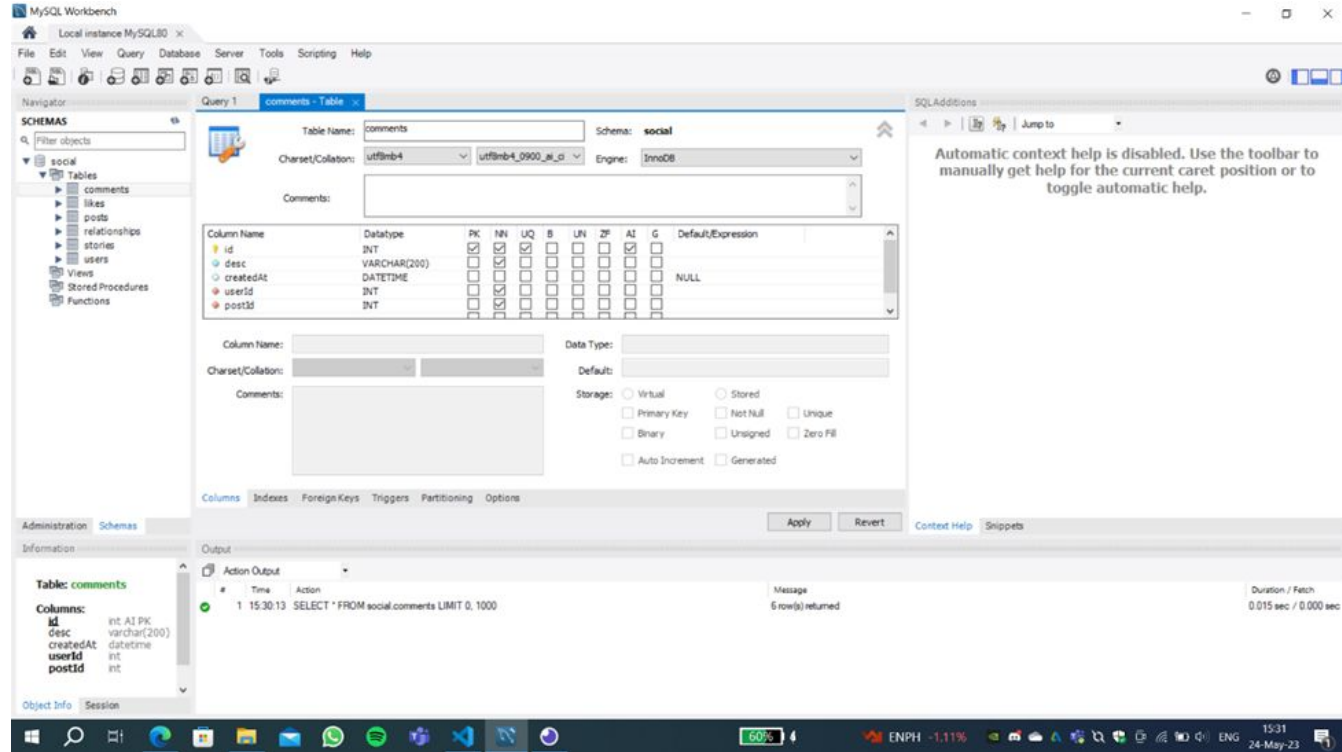
# Main Profile



# **Application Implementation**



# Database Implementation



The screenshot displays the MySQL Workbench interface for a local instance of MySQL 8.0. The 'Schemas' pane on the left shows the 'social' database selected, with a tree view of its tables: comments, likes, posts, relationships, stories, and users. The 'Table: comments' is selected, showing its structure in the 'Columns' tab. The table has the following columns:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
desc	VARCHAR(200)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
createdAt	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
userId	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
postId	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

The 'Query' tab shows a query named 'Query 1' with the following SQL:

```
SELECT * FROM social.comments LIMIT 0, 1000
```

The 'Output' pane at the bottom shows the query result, indicating that 6 rows were returned. The 'SQLAdditions' pane on the right contains a message: 'Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.'

# Implementation Details

## Some code from App.js



```
import Login from "../pages/login/Login";
import Register from "../pages/register/Register";
import {
  createBrowserRouter,
  RouterProvider,
  Route,
  Outlet,
  Navigate,
} from "react-router-dom";
import Navbar from "../components/navbar/Navbar";
import LeftBar from "../components/leftBar/LeftBar";
import RightBar from "../components/rightBar/RightBar";
import Home from "../pages/home/Home";
import Profile from "../pages/profile/Profile";
import "../style.scss";
import { useContext } from "react";
import { DarkModeContext } from
  "../context/darkModeContext";
import { AuthContext } from "../context/authContext";
import { QueryClient, QueryClientProvider } from
  'react-query'
```

```
function App() {
  const { currentUser } = useContext(AuthContext);

  const { darkMode } = useContext(DarkModeContext);
```

```
  const queryClient = new QueryClient()
```

```
  const Layout = () => {
    return (
      <QueryClientProvider client={queryClient}>
        <div className={theme === "dark" ? "dark" :
  "light"}>
          <Navbar />
          <div style={{ display: "flex" }}>
            <LeftBar />
            <div style={{ flex: 6 }}>
              <Outlet />
            </div>
            <RightBar />
          </div>
        </div>
      </QueryClientProvider>
    )
  }
```

```
const ProtectedRoute = ({ children
}) => {
  if (!currentUser) {
    return <Navigate to="/login" />;
  }

  return children;
};
```

```
const router =
createBrowserRouter([
  {
    path: "/",
    element: (
      <ProtectedRoute>
        <Layout />
      </ProtectedRoute>
    ),
    children: [
      {
        path: "/",
        element: <Home />,
      },
      {
        path: "/profile/:id",
        element: <Profile />,
      },
    ],
  },
  {
    path: "/login",
    element: <Login />,
  },
  {
    path: "/register",
    element: <Register />,
  },
]);

return (
  <div>
    <RouterProvider
      router={router} />
  </div>
);
```

## Conclusion

In conclusion, the WeShare application was developed following each step in the design process. From the idea of an application that can bring together people and help them keep in touch, to the testing of it, the entire team pitched in ideas and helped with solutions to parts that posed difficulties.

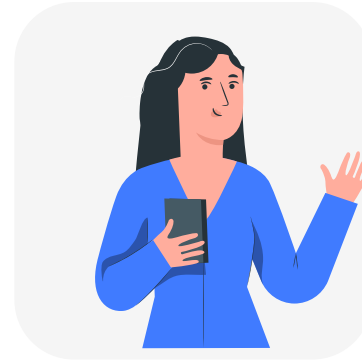




# Our team



**Evelyn Iulia  
Plesca**

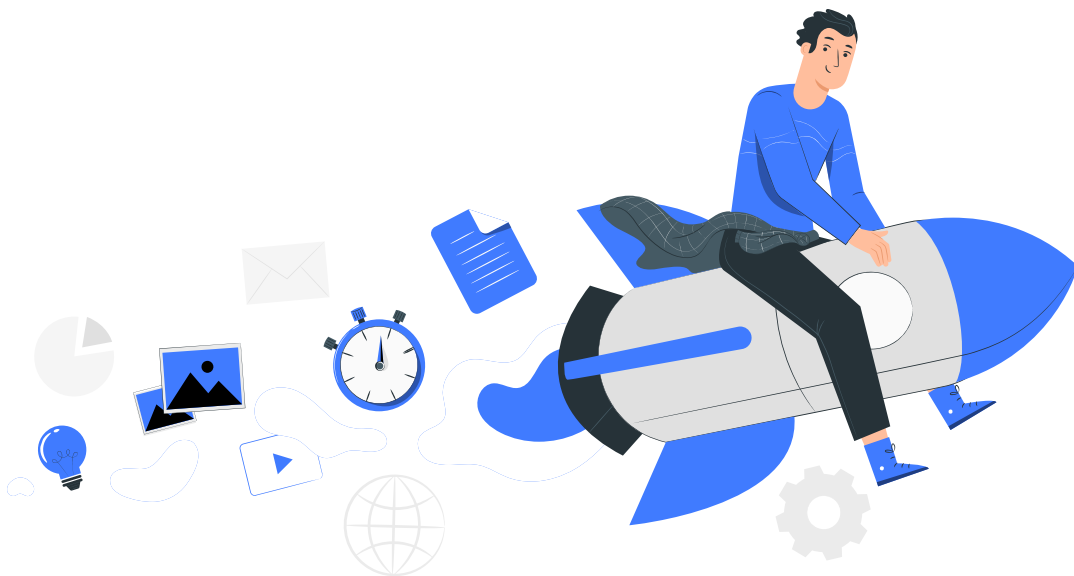


**Alexandra  
Taslaan**



**Sergiu Daniel  
Oprea**

# Thanks



This presentation template was created by  
**Slidesgo**, including icons by **Flaticon**, infographics  
& images by **Freepik** and illustrations by **Stories**