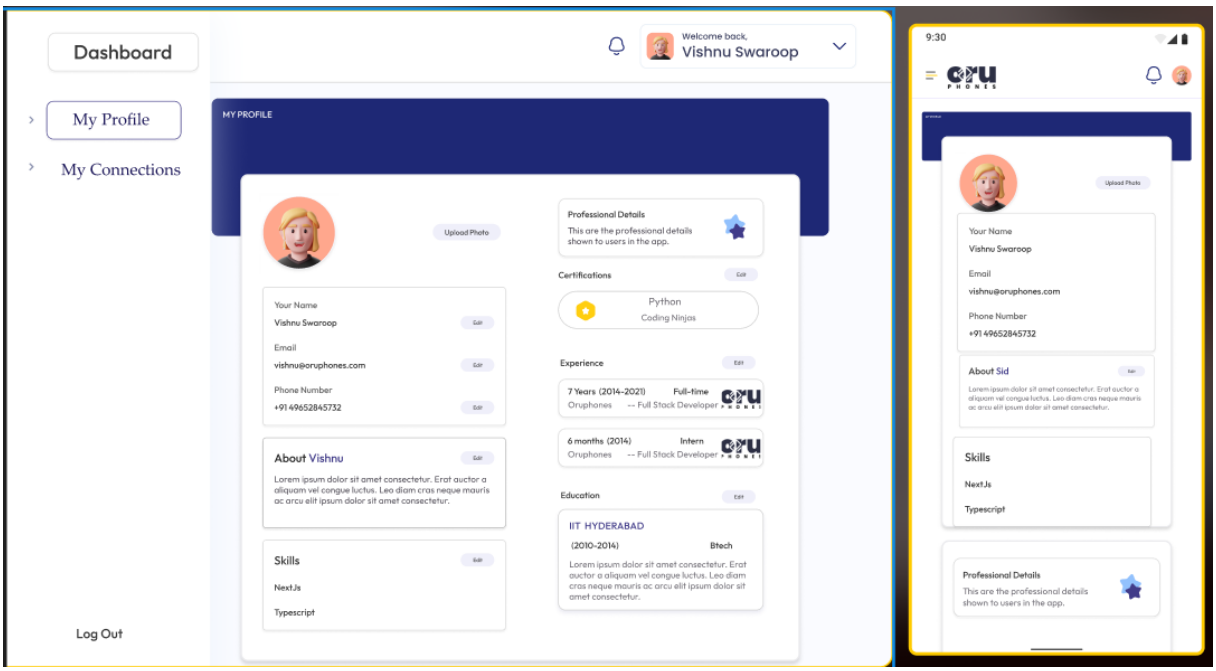
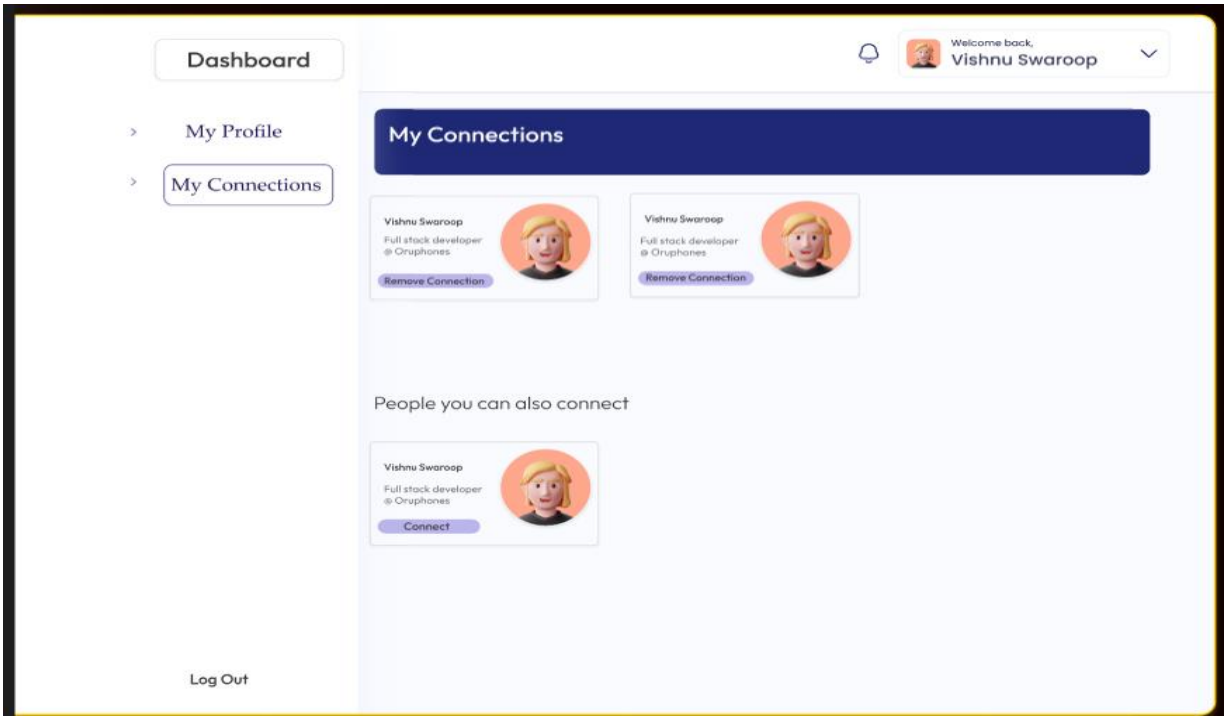
Assignment for position at “Mobilicis India Private Limited” - **Full Stack Developer**

For any queries related to below assignments, reach out to **contactus@oruphones.com** Or call us at **8247490377**





Figma :

https://www.figma.com/file/Y6UGcOCjpiYF21KNjWgzBs/ORUPHONES-ASSIGNMENT-FSD?type=design&node-id=0%3A1&mode=design&t=P0KGyfvHAd2AcndV-1

Objective: The goal of this assignment is to create a dynamic user profile page with two main sections: Profile and Connections. You are required to implement or Replicate the frontend using Tailwind CSS, Next.js, and Nodejs, etc. Additionally, you have the flexibility to design the login and signup process and decide whether to gather basic or professional details in the signup process itself or you can also implement edit popups for profile information.

Instructions:

Frontend Development:

- Create a Next.js project.
- Utilize Tailwind CSS for styling and layout.
- Design the login and signup process:
- Collect basic user details or include options for professional information such as experience, education, skills, etc.
- Implement a user authentication system that allows users to log in and access their profile.
- Design the Profile page:
- Display user information such as name, profile picture, bio, and editable professional details. (For profile picture upload you can use cloudinary)
- Include an "Edit" button that opens a popup for users to modify their profile information.
- Design the Connections page:
- Display a list of user connections or friends.
- Create 10 duplicate users and use them to demonstrate adding and removing connections.
- Implement a mechanism to move connected users to the top and rearrange the list upon connection removal.

Backend Development : (Nodejs + MongoDB)

- Implement API endpoints for user authentication (login and session management).
- Implement API endpoints for fetching and updating user profiles, connections, and professional details.
- Use MongoDB for data storage and retrieval:

Bonus Points: Redis Caching (Optional but Recommended):

- Implement Redis caching to enhance backend performance:
- Check Redis cache before fetching data from MongoDB.
- If data is found in Redis cache, return cached data.
- If data is not found, fetch from MongoDB, cache it in Redis, and return the data.

Deployment:

- Deploy the project to a hosting platform of your choice (e.g., Vercel, Netlify, Heroku) for both frontend and backend.
- Include deployment instructions in the repository's README file (Optional).

Submission Guidelines:

- Create a GitHub repository to host the project code.
- Include setup and deployment instructions in the repository's README file.
- Provide clear documentation on how to access and use the deployed application.
- Share the live deployed project URL on internshala
- Share the Github repository link in internshala (Don't share zip folder of source code).

Evaluation Criteria:

You will be evaluated based on the following aspects:

- Correctness and functionality of backend API endpoints.
- User experience, responsiveness, and visual design of the frontend.
- Proper utilization of MongoDB for data storage and retrieval.
- Implementation and effectiveness of Redis caching (for bonus points).
- Code organization, readability, and adherence to best practices.
- Creativity in design and implementation, including the connections page behavior.

Additional Notes:

- You can choose the level of detail they collect during login and signup.
- The Redis caching implementation is optional but highly recommended to showcase performance optimization skills.