Task:1

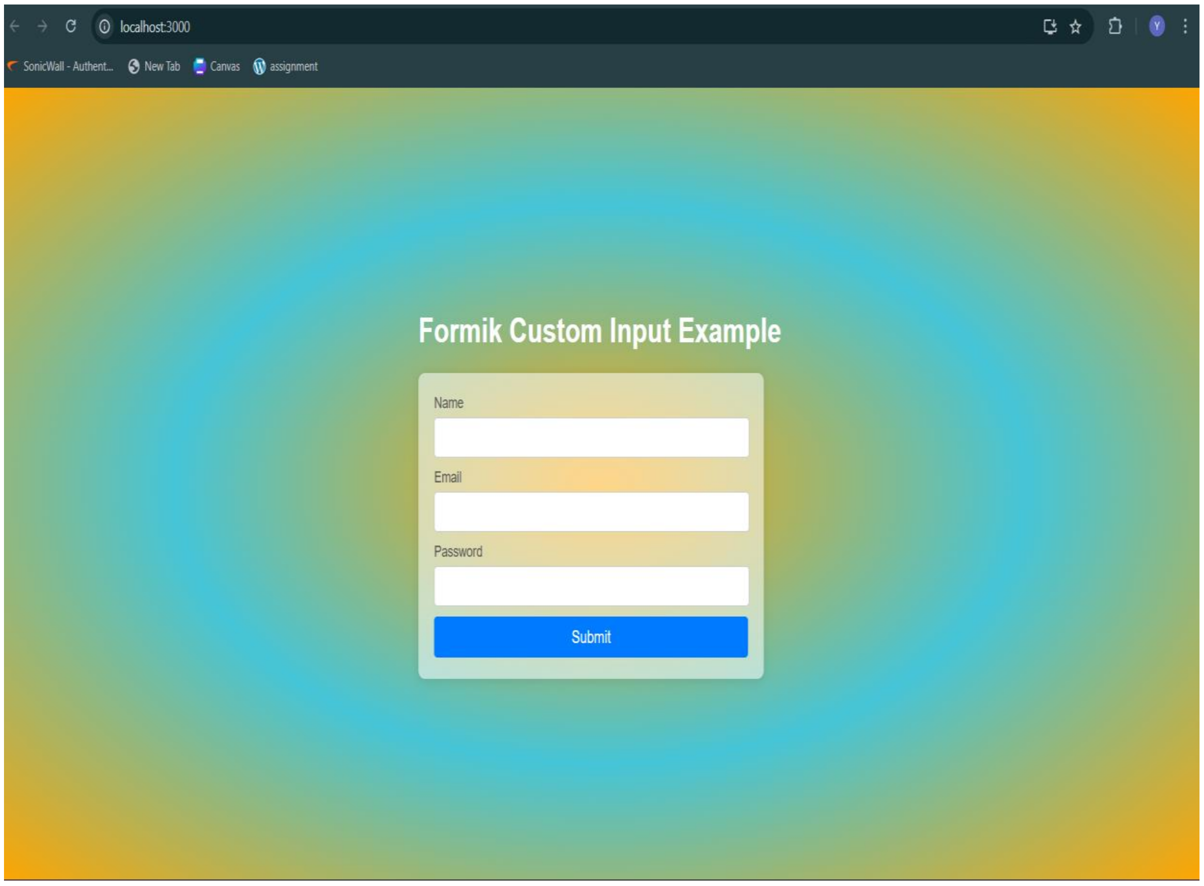
Task Name: Formik

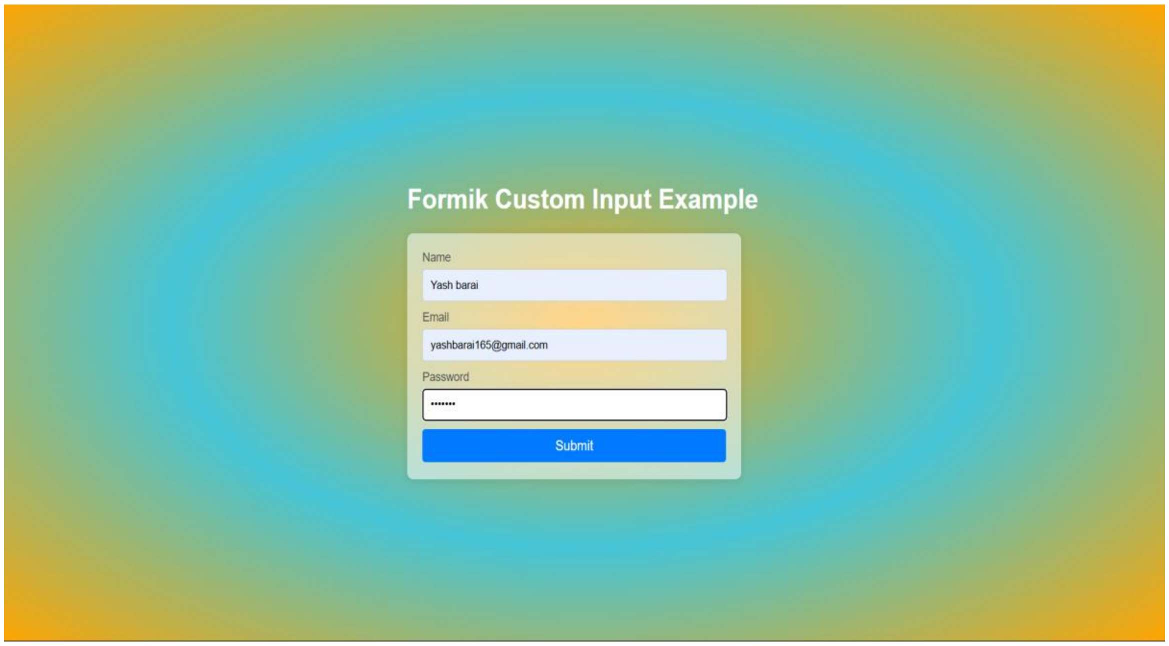
Project Name: Formik-example

1. Task Description

The task involves creating a custom reusable input component using **Formik**, a popular form management library in React, with various validation types such as required fields, min/max length, email format, etc. This component should integrate seamlessly with Formik's form handling and validation, allowing it to be used in multiple places across the application with different validation rules and input types.

2. Task Output Screenshot




Formik Custom Input Example

Name

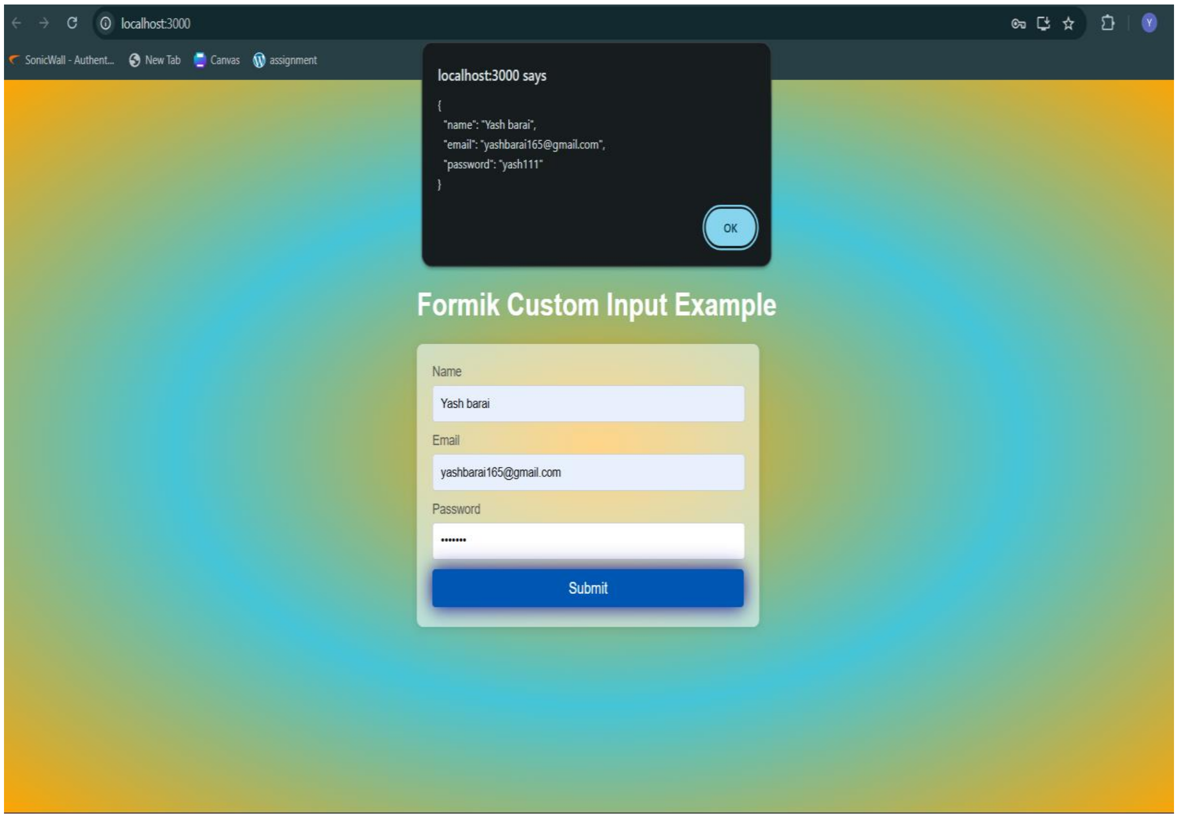
Yash barai

Email

yashbarai165@gmail.com

Password

•••••••



localhost:3000 says

```
{
  "name": "Yash barai",
  "email": "yashbarai165@gmail.com",
  "password": "yash111"
}
```

Formik Custom Input Example

Name

Yash barai

Email

yashbarai165@gmail.com

Password

•••••••

3. Widget/Algorithm Used In Task

- **Formik Field**: Used to manage form field states like value, error, touched, etc.
- **Yup Validation Schema**: Yup is used to define and enforce validation rules (e.g., required fields, email format).
- **ErrorMessage**: Widget from Formik that automatically displays validation errors.

4. Algorithm explanation

- Define a reusable input component that receives props for label, type, placeholder, and validation schema.
- Bind this component to Formik's `<Field>` for managing state and value changes.
- Use Yup to apply dynamic validation rules based on the props passed (e.g., for email, min/max length).
- Show validation errors using `<ErrorMessage>` when input does not meet validation requirements.
- Export the component for use in any form, allowing for flexible validation.

5. How to start:

Install Node Module in frontend and backend folder: Go to terminal "npm install" How to Start: Go to terminal "npm run server" OR "npm start"