

```
[
  {
    "_id": "67c50ce33633593a527a3b4d",
    "name": "ABC Organization",
    "shortname": "ABC",
    "description": "A description of ABC Organization.",
    "address": "123 Main St, City, Country",
    "representation": "Main",
    "operationDetails": {
      "number_of_members": {
        "male": 150,
        "female": 120
      },
      "_id": "67c50ce33633593a527a3b47",
      "organization_registration": "ORG123456789",
      "date_established": "2010-06-15T00:00:00.000Z",
      "psic": "12345",
      "target_members": "Small Businesses, Farmers",
      "annual_production": [
        {
          "product": "Corn",
          "type": "Crop",
          "quantity": 5000,
          "unit": "kg",
          "market_value": 10000,
          "_id": "67c50ce33633593a527a3b48"
        },
        {
          "product": "Wheat",
          "type": "Crop",
          "quantity": 3000,
          "unit": "kg",
          "market_value": 7500,
          "_id": "67c50ce33633593a527a3b49"
        }
      ]
    }
  }
]
```

```
"name": "TESTING PLEASE WORK",
"shortname": "TPW",
"description": "This is a test CBO",
"address": "Payatas, Quezon City",
"representation": "Main",
"operationDetails": {
  "organization_registration": "Cooperative",
  "date_established": "2025-03-20",
  "psic": "asd",
  "target_members": "Farmers",
  "number_of_members": {
    "male": 15,
    "female": 2
  },
  "annual_production": [
    {
      "product": "Cabbage",
      "type": "Vegetable",
      "quantity": 100,
      "unit": "kg",
      "market_value": 5000
    }
  ],
  "production_scope": "Manila, Quezon City",
  "sales_scope": "Manila, Quezon City",
  "total_assets": "155",
  "total_liabilities": "155",
  "annual_gross_income": "1554",
  "procurement_experience": [
    {
      "method": "Small Value Procurement",
      "participation_count": 0,
      "contracts_won": 0,
    }
  ]
}
```

<pre> }], "production_scope": "National", "sales_scope": "Local", "total_assets": 5000000, "total_liabilities": 2000000, "annual_gross_income": 1000000, "procurement_experience": [{ "method": "Government", "participation_count": 10, "contracts_won": 7, "successful_implementations": 5, "_id": "67c50ce33633593a527a3b4a" }, { "method": "Private", "participation_count": 8, "contracts_won": 6, "successful_implementations": 4, "_id": "67c50ce33633593a527a3b4b" }], "sponsor_agency": "XYZ Funding Agency", "other_sponsor_agency": "ABC Sponsorship Org", "__v": 0 }, "createdAt": "2025-03-03T01:58:59.538Z", "updatedAt": "2025-03-03T01:58:59.538Z", "__v": 0 }] </pre>	<pre> "successful_implementations": 0 }], "sponsor_agency": "EPAHP", "other_sponsor_agency": "" } } </pre>
--	---

Functioning RegistrationForm.js:

```
import React, { useState } from "react";
import BasicInfoStep from "../registrationSteps/BasicInfoStep";
import OperationStep from "../registrationSteps/OperationStep";
import { registerCBO } from "../utils/api"; // Import API function

const RegistrationForm = () => {
  const [step, setStep] = useState(1);
  const [loading, setLoading] = useState(false); // Loading state
  const [formData, setFormData] = useState({
    name: "",
    shortname: "",
    description: "",
    address: "",
    representation: "",
    operationDetails: { // Update key to camelCase to match backend
      organization_registration: "",
      date_established: "",
      psic: "",
      target_members: "",
      number_of_members: { male: 0, female: 0 },
      annual_production: [],
      production_scope: "",
      sales_scope: "",
      total_assets: 0,
      total_liabilities: 0,
      annual_gross_income: 0,
      procurement_experience: [],
      sponsor_agency: "",
    },
  });
```

```
    other_sponsor_agency: "",
  }
});

const validateStep = () => {
  if (step === 1) {
    if (!formData.name || !formData.shortname || !formData.address || !formData.representation) {
      alert("Please fill in all required fields");
      return false;
    }
  }
  return true;
};

const nextStep = () => {
  if (validateStep()) setStep(step + 1);
};

const prevStep = () => setStep(step - 1);

const handleSubmit = async (e) => {
  e.preventDefault();
  setLoading(true);

  console.log("Submitting CBO Data:", JSON.stringify(formData, null, 2)); // Debugging

  try {
    const response = await registerCBO(formData); // API call
    console.log("Server Response:", response);
  }
}
```

```
alert("Registration Successful!");
setFormData({
  name: "",
  shortname: "",
  description: "",
  address: "",
  representation: "",
  operationDetails: {
    organization_registration: "",
    date_established: "",
    psic: "",
    target_members: "",
    number_of_members: { male: 0, female: 0 },
    annual_production: [],
    production_scope: "",
    sales_scope: "",
    total_assets: 0,
    total_liabilities: 0,
    annual_gross_income: 0,
    procurement_experience: [],
    sponsor_agency: "",
    other_sponsor_agency: "",
  },
});

setStep(1);
} catch (error) {
```

```

        console.error("Error registering CBO:", error);
        alert("Registration Failed. Check console for details.");
    } finally {
        setLoading(false);
    }
};

return (
    <form onSubmit={handleSubmit}>
        {step === 1 && <BasicInfoStep formData={formData} setFormData={setFormData} />}
        {step === 2 && (
            <OperationStep
                formData={formData.operationDetails}
                setFormData={(newOperationData) =>
                    setFormData({ ...formData, operationDetails: newOperationData })
                }
            />
        )}
    </form>
    <div>
        {step > 1 && <button type="button" onClick={prevStep}>Back</button>}
        {step < 2 && <button type="button" onClick={nextStep}>Next</button>}
        {step === 2 && (
            <button type="submit" disabled={loading}>
                {loading ? "Submitting..." : "Submit"}
            </button>
        )}
    </div>
);

```

```
};
```

```
export default RegistrationForm;
```

```
import React, { useState } from "react";
import BasicInfoStep from "../registrationSteps/BasicInfoStep";
import OperationStep from "../registrationSteps/OperationStep";
import { registerCBO } from "../utils/api"; // Import API function
```

```
const RegistrationForm = () => {
  const [step, setStep] = useState(1);
  const [loading, setLoading] = useState(false); // Loading state
  const [customOrg, setCustomOrg] = useState("");
  const [formData, setFormData] = useState({
    name: "",
    shortname: "",
    description: "",
    address: "",
    representation: "",
    operationDetails: {
      organization_registration: "",
      date_established: "",
      psic: "",
      target_members: "",
      number_of_members: { male: 0, female: 0 },
    },
  });
```

```

    annual_production: [],
    production_scope: "",
    sales_scope: "",
    total_assets: 0,
    total_liabilities: 0,
    annual_gross_income: 0,
    procurement_experience: [],
    sponsor_agency: "",
    other_sponsor_agency: "",
  }
});

const validateStep = () => {
  if (step === 1) {
    if (!formData.name || !formData.shortname || !formData.address || !formData.representation) {
      alert("Please fill in all required fields");
      return false;
    }
  }
  return true;
};

const nextStep = () => {
  if (validateStep()) setStep(step + 1);
};

const prevStep = () => setStep(step - 1);

const handleSubmit = async (e) => {

```



```
e.preventDefault();
setLoading(true);

// Ensure "Others" input is stored correctly before submission
const finalData = {
  ...formData,
  operationDetails: {
    ...formData.operationDetails,
    organization_registration:
      formData.operationDetails.organization_registration === "Others"
        ? customOrg
        : formData.operationDetails.organization_registration,
  },
};

try {
  const response = await registerCBO(finalData); // API call
  console.log("Server Response:", response);

  alert("Registration Successful!");
  setFormData({
    name: "",
    shortname: "",
    description: "",
    address: "",
    representation: "",
    operationDetails: {
      organization_registration: "",
      date_established: "",
      psic: "",
    },
  });
}
```

```

        target_members: "",
        number_of_members: { male: 0, female: 0 },
        annual_production: [],
        production_scope: "",
        sales_scope: "",
        total_assets: 0,
        total_liabilities: 0,
        annual_gross_income: 0,
        procurement_experience: [],
        sponsor_agency: "",
        other_sponsor_agency: "",
    },
});

    setStep(1);
} catch (error) {
    console.error("Error registering CBO:", error);
    alert("Registration Failed. Check console for details.");
} finally {
    setLoading(false);
}
};

return (
    <form onSubmit={handleSubmit}>
        {step === 1 && <BasicInfoStep formData={formData} setFormData={setFormData} />}
        {step === 2 && (
<OperationStep

```

```

    formData={formData.operationDetails}
    setFormData={ (newOperationData) =>
      setFormData({ ...formData, operationDetails: newOperationData })
    }
    customOrg={customOrg} // Pass customOrg to OperationStep
    setCustomOrg={setCustomOrg} // Pass setter to update input
  />
)}

<div>
  {step > 1 && <button type="button" onClick={prevStep}>Back</button>}
  {step < 2 && <button type="button" onClick={nextStep}>Next</button>}
  {step === 2 && (
    <button type="submit" disabled={loading}>
      {loading ? "Submitting..." : "Submit"}
    </button>
  )}
</div>
</form>
);
};

export default RegistrationForm;

```

Last handlers:

```

const handleOrgChange = (e) => {
  const { value } = e.target;

```

```
if (value === "Others") {  
  // Keep the selected value as "Others" but don't store it in the final data yet  
  setFormData({ ...formData, organization_registration: "Others" });  
  setCustomOrg(""); // Reset custom input field  
} else {  
  // Set the selected organization type and clear custom input  
  setFormData({ ...formData, organization_registration: value });  
  setCustomOrg("");  
}  
};  
  
const handleCustomOrgChange = (e) => {  
  setCustomOrg(e.target.value);  
};
```