

```
import * as React from 'react';
import Box from '@mui/material/Box';
import Stepper from '@mui/material/Stepper';
import Step from '@mui/material/Step';
import StepLabel from '@mui/material/StepLabel';
import StepContent from '@mui/material/StepContent';
import Button from '@mui/material/Button';
import Paper from '@mui/material/Paper';
import Typography from '@mui/material/Typography';

const steps = [
  {
    label: 'Select campaign settings',
    description: `For each ad campaign that you create, you can control how
much
        you're willing to spend on clicks and conversions, which networks
        and geographical locations you want your ads to show on, and
more.` ,
  },
  {
    label: 'Create an ad group',
    description:
      'An ad group contains one or more ads which target a shared set of
keywords.',
  },
  {
    label: 'Create an ad',
    description: `Try out different ad text to see what brings in the most
customers,
        and learn how to enhance your ads using features like ad
extensions.
        If you run into any problems with your ads, find out how to tell if
        they're running and how to resolve approval issues.` ,
  },
];
```

```

export default function VerticalLinearStepper() {
  const [activeStep, setActiveStep] = React.useState(0);

  const handleNext = () => {
    setActiveStep((prevActiveStep) => prevActiveStep + 1);
  };

  const handleBack = () => {
    setActiveStep((prevActiveStep) => prevActiveStep - 1);
  };

  const handleReset = () => {
    setActiveStep(0);
  };

  return (
    <Box sx={{ maxWidth: 400 }}>
      <Stepper activeStep={activeStep} orientation="vertical">
        {steps.map((step, index) => (
          <Step key={step.label}>
            <StepLabel
              optional={
                index === 2 ? (
                  <Typography variant="caption">Last step</Typography>
                ) : null
              }
            >
              {step.label}
            </StepLabel>
            <StepContent>
              <Typography>{step.description}</Typography>
              <Box sx={{ mb: 2 }}>
                <div>
                  <Button
                    variant="contained"
                    onClick={handleNext}

```

```

        sx={{ mt: 1, mr: 1 }}
      >
        {index === steps.length - 1 ? 'Finish' : 'Continue'}
      </Button>
      <Button
        disabled={index === 0}
        onClick={handleBack}
        sx={{ mt: 1, mr: 1 }}
      >
        Back
      </Button>
    </div>
  </Box>
</StepContent>
</Step>
  )}
</Stepper>
{activeStep === steps.length && (
  <Paper square elevation={0} sx={{ p: 3 }}>
    <Typography>All steps completed - you're finished</
Typography>
    <Button onClick={handleReset} sx={{ mt: 1, mr: 1 }}>
      Reset
    </Button>
  </Paper>
  )}
</Box>
);
}

```