

REACT JS PROJECT

CA 3

TOPIC: BUDGET APPLICATION



L OVELY
P ROFESSIONAL
U NIVERSITY

Transforming Education Transforming India

NAME: HARSH BANSAL

REGISTRATION NO. :- 12012999

SECTION: K20ND

ROLL NO. :- RK20NDA11

SEMESTER:- 6th

SCHOOL: - COMPUTER SCIENCE AND ENGINEERING

INTRODUCTION

ReactJS, often referred to simply as React, is an open-source JavaScript library for building user interfaces for web applications. Developed and maintained by Facebook, React is one of the most popular libraries for front-end development and has gained widespread adoption in the web development community. React follows a component-based architecture, where the UI is broken down into small, reusable components that can be combined to build complex user interfaces. These components are written using JSX (JavaScript XML), which is a syntax extension that allows developers to write HTML-like code within JavaScript. React provides a virtual DOM (Document Object Model) that allows for efficient updates and rendering of UI components, resulting in better performance and faster user experiences. React also supports a unidirectional data flow, making it easy to manage the state of UI components and maintain a clear flow of data throughout the application. React is often used in combination with other tools and libraries to build complete web applications, such as Redux for state management, React Router for routing, and Axios for handling API requests. React has a large and active community, which provides extensive documentation, tutorials, and support, making it a popular choice for building modern, interactive, and responsive web applications.

PROJECT DEFINITION

PROJECT OVERVIEW

BUDGET APP created using React js. This application helps in tracking expenses.

-We can ADD BUDGET which takes in Name and Budget amount to be spent on a stipulated time.

-We can ADD EXPENSES to the budget card which gives information about how much money we have spent and how much is remaining.

-It takes in input as Description, Amount, and Budget drop down which gives us the option where we have to add that expense in which Budget Card.

-We can view our expenses and delete them as per our need also we can delete the Budget Card if we want

SOFTWARE REQUIREMENTS

- VS CODE
- LAPTOP

SKILLS REQUIRED:-

- GITHUB
- REACT JS
- FIREBASE (HOSTING)
- HTML
- BOOTSTRAP

IMPLEMENTATION

IMPLEMENTATION OF THE PROJECT

SOURCE CODE:-

COMPONENT :- AddBudgetModal.js

```
import { Form, Modal, Button } from "react-bootstrap"
import { useRef } from "react"
import { useBudgets } from "../contexts/BudgetsContext"

export default function AddBudgetModal({ show, handleClose }) {
  const nameRef = useRef()
  const maxRef = useRef()
  const { addBudget } = useBudgets()
  function handleSubmit(e) {
    e.preventDefault()
    addBudget({
      name: nameRef.current.value,
      max: parseFloat(maxRef.current.value),
    })
    handleClose()
  }

  return (
    <Modal show={show} onHide={handleClose}>
      <Form onSubmit={handleSubmit}>
        <Modal.Header closeButton>
          <Modal.Title>New Budget</Modal.Title>
        </Modal.Header>
        <Modal.Body>
          <Form.Group className="mb-3" controlId="name">
            <Form.Label>Name</Form.Label>
            <Form.Control ref={nameRef} type="text" required />
          </Form.Group>
          <Form.Group className="mb-3" controlId="max">
```

```

        <Form.Label>Maximum Spending</Form.Label>
        <Form.Control
          ref={maxRef}
          type="number"
          required
          min={0}
          step={0.01}
        />
      </Form.Group>
      <div className="d-flex justify-content-end">
        <Button variant="primary" type="submit">
          Add
        </Button>
      </div>
    </Modal.Body>
  </Form>
</Modal>
)
}

```

COMPONENT :- BudgetCard

```

import { Button, Card, ProgressBar, Stack } from "react-bootstrap"
import { currencyFormatter } from "../utils"

export default function BudgetCard({
  name,
  amount,
  max,
  gray,
  hideButtons,
  onAddExpenseClick,
  onViewExpensesClick,
}) {
  const classNames = []
  if (amount > max) {
    classNames.push("bg-danger", "bg-opacity-10")
  } else if (gray) {

```

```

        classNames.push("bg-light")
    }

    return (
        <Card className={classNames.join(" ")}>
            <Card.Body>
                <Card.Title className="d-flex justify-content-between align-items-baseline fw-normal mb-3">
                    <div className="me-2">{name}</div>
                    <div className="d-flex align-items-baseline">
                        {currencyFormatter.format(amount)}
                        {max && (
                            <span className="text-muted fs-6 ms-1">
                                / {currencyFormatter.format(max)}
                            </span>
                        )}
                    </div>
                </Card.Title>
                {max && (
                    <ProgressBar
                        className="rounded-pill"
                        variant={getProgressBarVariant(amount, max)}
                        min={0}
                        max={max}
                        now={amount}
                    />
                )}
                {!hideButtons && (
                    <Stack direction="horizontal" gap="2" className="mt-4">
                        <Button
                            variant="outline-primary"
                            className="ms-auto"
                            onClick={onAddExpenseClick}
                        >
                            Add Expense
                        </Button>
                        <Button onClick={onViewExpensesClick} variant="outline-secondary">
                            View Expenses
                        </Button>
                    </Stack>
                )}
            </Card.Body>
        </Card>
    )
}

```

```

        </Card.Body>
      </Card>
    )
  }

function getProgressBarVariant(amount, max) {
  const ratio = amount / max
  if (ratio < 0.5) return "primary"
  if (ratio < 0.75) return "warning"
  return "danger"
}

```

COMPONENT:- ViewExpensesModal

```

import { Modal, Button, Stack } from "react-bootstrap"
import { UNCATEGORIZED_BUDGET_ID, useBudgets } from "../contexts/BudgetsContext"
import { currencyFormatter } from "../utils"

export default function ViewExpensesModal({ budgetId, handleClose }) {
  const { getBudgetExpenses, budgets, deleteBudget, deleteExpense } =
    useBudgets()

  const expenses = getBudgetExpenses(budgetId)
  const budget =
    UNCATEGORIZED_BUDGET_ID === budgetId
      ? { name: "Uncategorized", id: UNCATEGORIZED_BUDGET_ID }
      : budgets.find(b => b.id === budgetId)

  return (
    <Modal show={budgetId !== null} onHide={handleClose}>
      <Modal.Header closeButton>
        <Modal.Title>
          <Stack direction="horizontal" gap="2">
            <div>Expenses - {budget?.name}</div>
            {budgetId !== UNCATEGORIZED_BUDGET_ID && (
              <Button
                onClick={() => {
                  deleteBudget(budget)

```



```

        handleClose()
      }}
      variant="outline-danger"
    >
      Delete
    </Button>
  )}
</Stack>
</Modal.Title>
</Modal.Header>
<Modal.Body>
  <Stack direction="vertical" gap="3">
    {expenses.map(expense => (
      <Stack direction="horizontal" gap="2" key={expense.id}>
        <div className="me-auto fs-4">{expense.description}</div>
        <div className="fs-5">
          {currencyFormatter.format(expense.amount)}
        </div>
        <Button
          onClick={() => deleteExpense(expense)}
          size="sm"
          variant="outline-danger"
        >
          &times;
        </Button>
      </Stack>
    )}}
  </Stack>
</Modal.Body>
</Modal>
)
}

```

COMPONENT:- TotalBudgetCard

```

import { useBudgets } from "../contexts/BudgetsContext"
import BudgetCard from "../BudgetCard"

```

```

export default function TotalBudgetCard() {
  const { expenses, budgets } = useBudgets()
  const amount = expenses.reduce((total, expense) => total + expense.amount, 0)
  const max = budgets.reduce((total, budget) => total + budget.max, 0)
  if (max === 0) return null

  return <BudgetCard amount={amount} name="Total" gray max={max} hideButtons />
}

```

CONTEXT :- BudgetsContext

```

import React, { useContext, useState } from "react"
import { v4 as uuidV4 } from "uuid"
import useLocalStorage from "../hooks/useLocalStorage"

const BudgetsContext = React.createContext()

export const UNCATEGORIZED_BUDGET_ID = "Uncategorized"

export function useBudgets() {
  return useContext(BudgetsContext)
}

export const BudgetsProvider = ({ children }) => {
  const [budgets, setBudgets] = useLocalStorage("budgets", [])
  const [expenses, setExpenses] = useLocalStorage("expenses", [])

  function getBudgetExpenses(budgetId) {
    return expenses.filter(expense => expense.budgetId === budgetId)
  }

  function addExpense({ description, amount, budgetId }) {
    setExpenses(prevExpenses => {
      return [...prevExpenses, { id: uuidV4(), description, amount, budgetId }]
    })
  }

  function addBudget({ name, max }) {
    setBudgets(prevBudgets => {
      if (prevBudgets.find(budget => budget.name === name)) {
        return prevBudgets
      }
    })
  }
}

```

```

    }
    return [...prevBudgets, { id: uuidV4(), name, max }]
  })
}

function deleteBudget({ id }) {
  setExpenses(prevExpenses => {
    return prevExpenses.map(expense => {
      if (expense.budgetId !== id) return expense
      return { ...expense, budgetId: UNCATEGORIZED_BUDGET_ID }
    })
  })

  setBudgets(prevBudgets => {
    return prevBudgets.filter(budget => budget.id !== id)
  })
}

function deleteExpense({ id }) {
  setExpenses(prevExpenses => {
    return prevExpenses.filter(expense => expense.id !== id)
  })
}

return (
  <BudgetsContext.Provider
    value={{
      budgets,
      expenses,
      getBudgetExpenses,
      addExpense,
      addBudget,
      deleteBudget,
      deleteExpense,
    }}
  >
    {children}
  </BudgetsContext.Provider>
)
}

```

HOOKS:- UseLocalStorage

```
import { useState, useEffect } from "react"

export default function useLocalStorage(key, defaultValue) {
  const [value, setValue] = useState(() => {
    const jsonValue = localStorage.getItem(key)
    if (jsonValue !== null) return JSON.parse(jsonValue)

    if (typeof defaultValue === "function") {
      return defaultValue()
    } else {
      return defaultValue
    }
  })

  useEffect(() => {
    localStorage.setItem(key, JSON.stringify(value))
  }, [key, value])

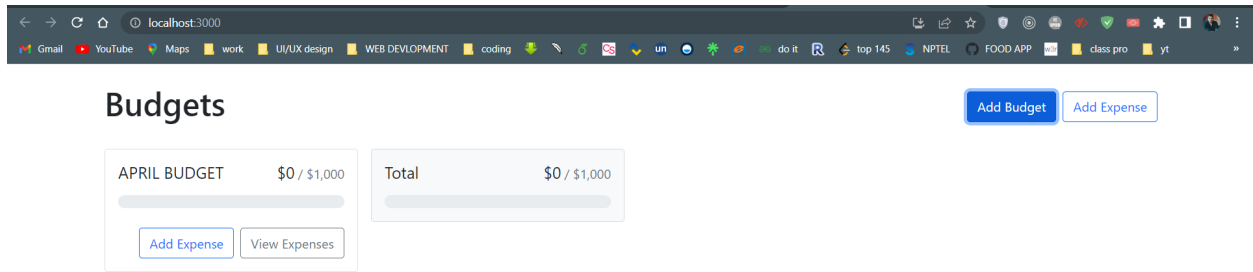
  return [value, setValue]
}
```

GITHUB LINK :- https://github.com/HarshBansal2k01/Budget_App

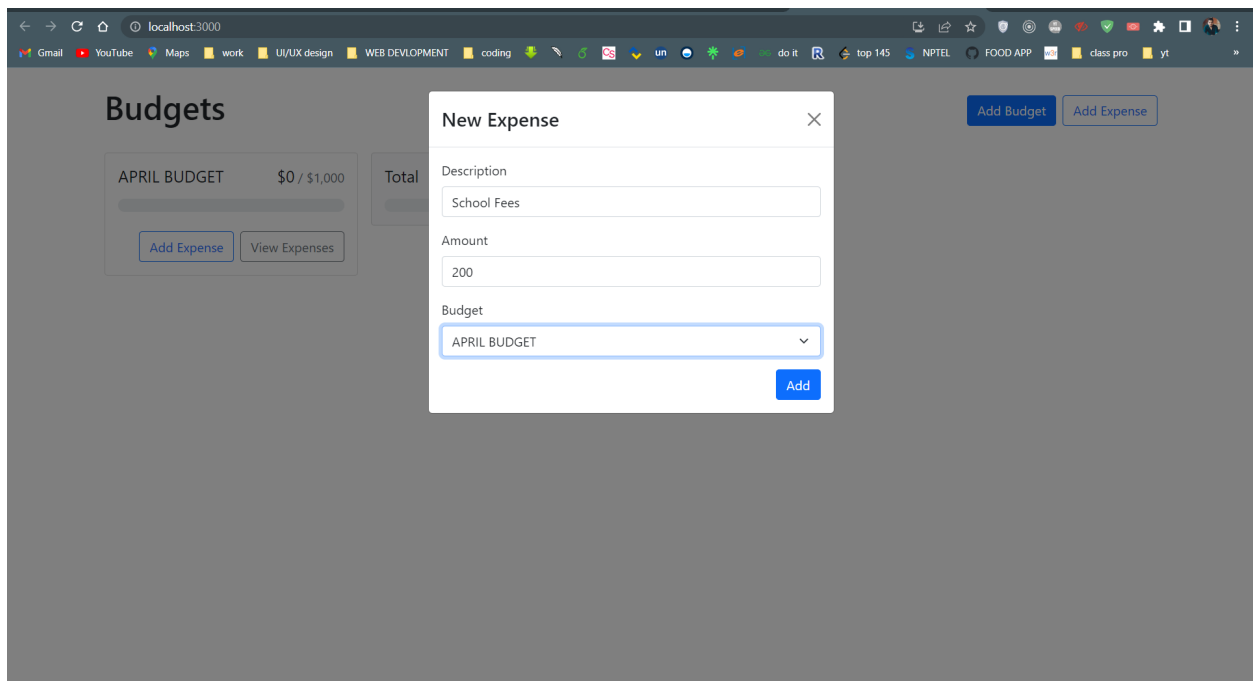
HOSTING :-

IMPLEMENTATION OUTPUT:-

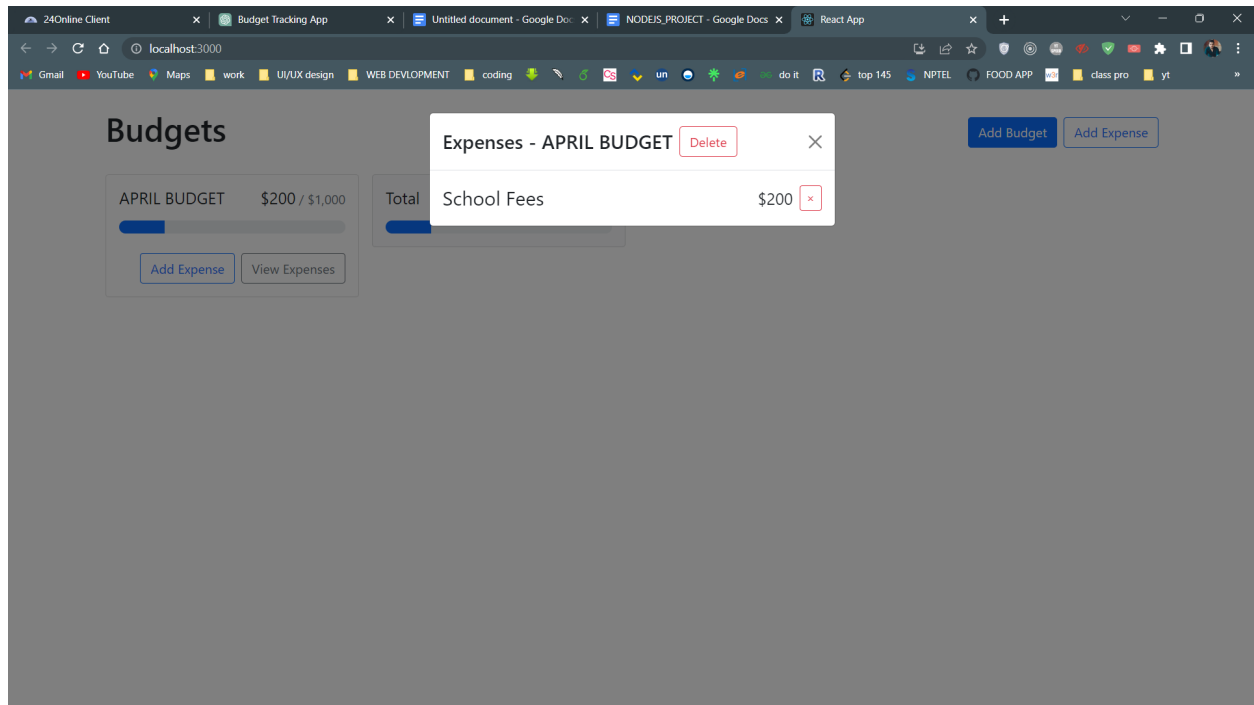
ADDING BUDGET CARD



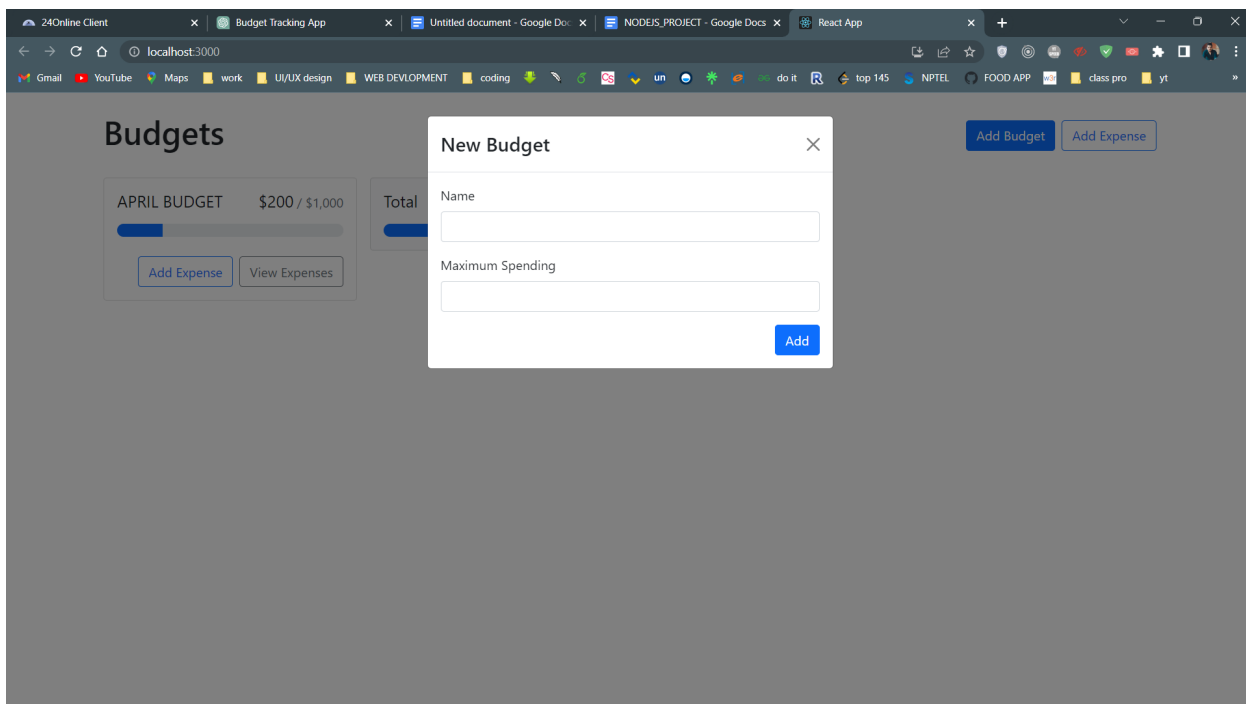
ADDING NEW EXPENSES TO APRIL BUDGET CARD



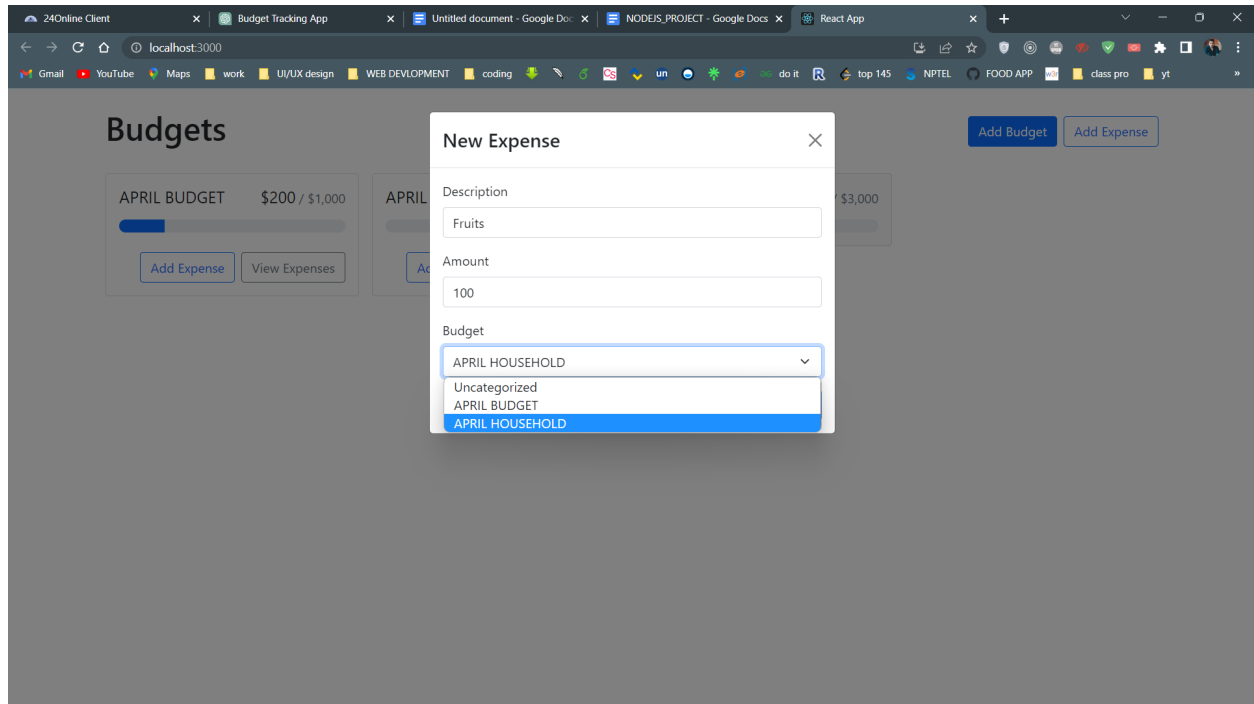
WE CAN DELETE & VIEW EXPENSES



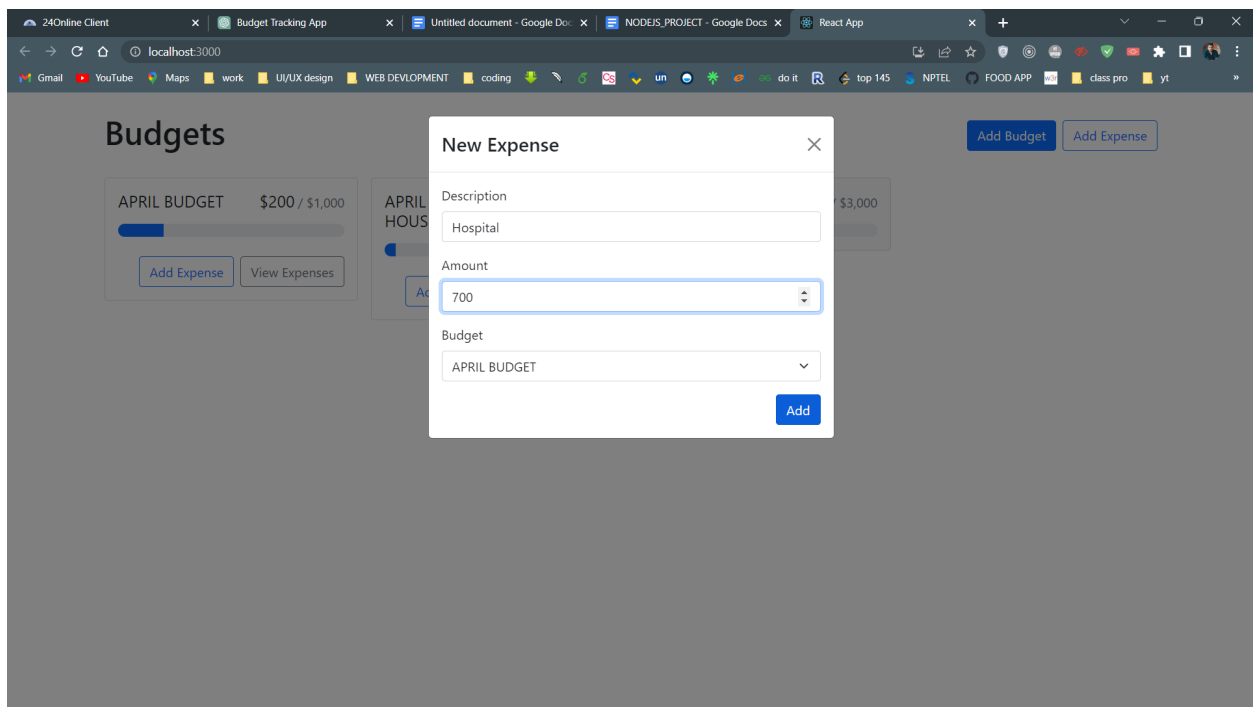
ADDING A NEW BUDGET CARD (APRIL HOUSEHOLD)



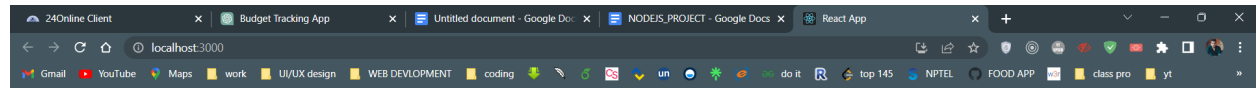
ADDING EXPENSE (FRUITS) TO APRIL HOUSEHOLD



ADDING EXPENSE (HOSPITAL) IN APRIL BUDGET



APRIL BUDGET SCALE BECOMES RED (CROSSED 75%)



Budgets

[Add Budget](#)[Add Expense](#)

APRIL BUDGET **\$900 / \$1,000**

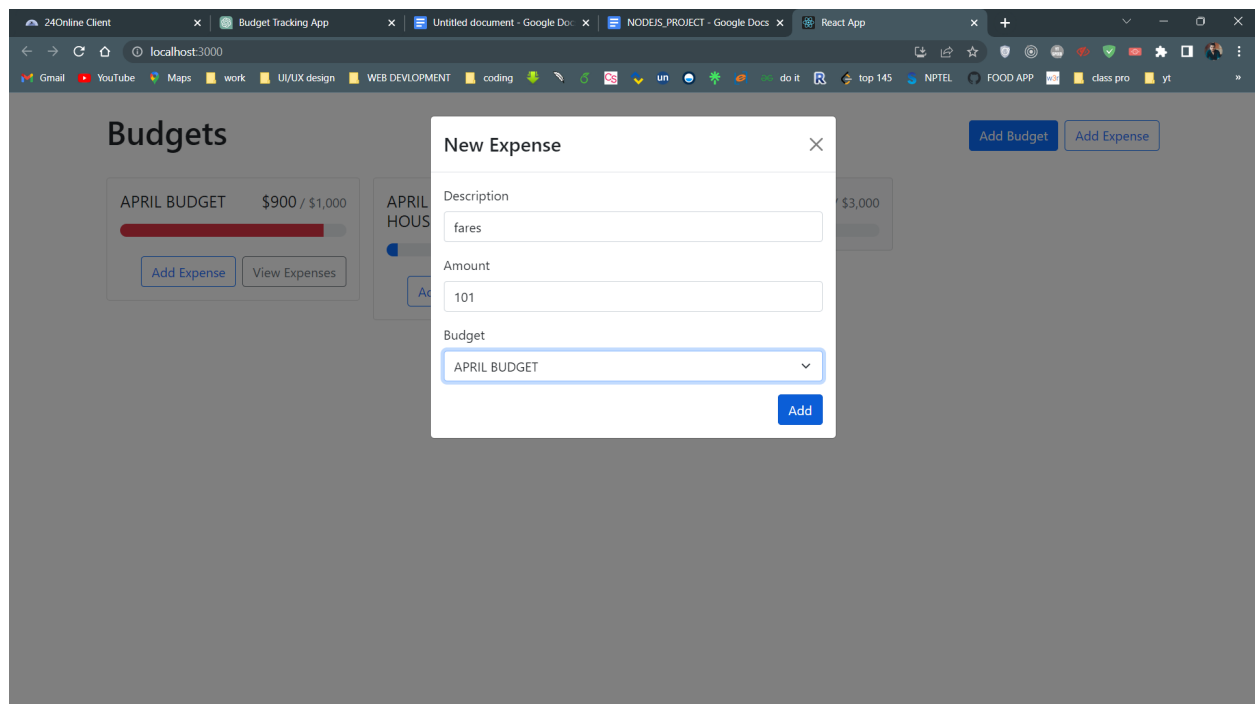
[Add Expense](#) [View Expenses](#)

APRIL HOUSEHOLD **\$100 / \$2,000**

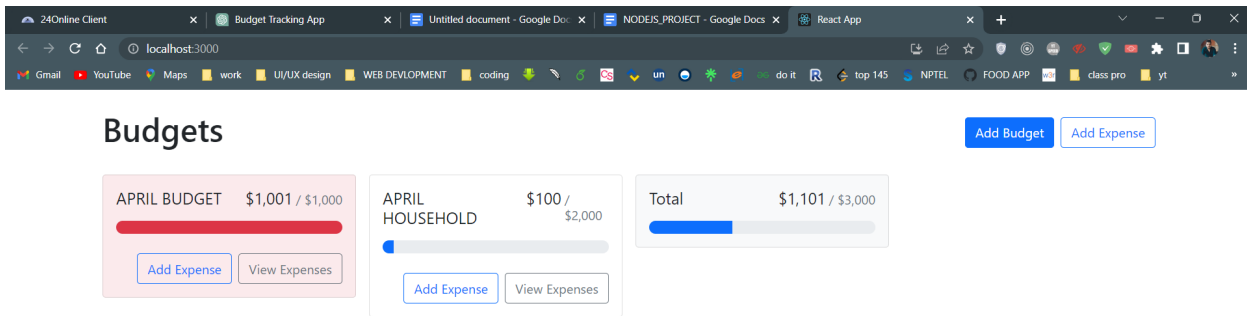
[Add Expense](#) [View Expenses](#)

Total **\$1,000 / \$3,000**

ADDING EXPENSE (FAIR) IN APRIL BUDGET



APRIL BUDGET CARD BECOMES RED AS IT CROSSES 100%



THE TOTAL BUDGET CROSSES 50% IT SCALE BECOMES YELLOW

