**SYNOPSIS**

**ON**

URBAN HARVEST

**submitted in partial fulfilment of the requirements for the award of degree of**

**BACHELOR OF ENGINEERING**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted by: Supervised By:**

**Harshit Sharma Dr. Isha Kansal**

**2010991265 Chitkara University**

**Aman**

**2010993672**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CHITKARA UNIVERSITY**

**CHANDIGARH-PATIALA NATIONAL HIGHWAY, RAJPURA, PUNJAB, INDIA**

**CONTENTS**

**Title Page No.**

1. Abstract
2. Methodology
3. Tools and Technologies
4. **Abstract**

* 1. **Problem Statement**

Ordering food for delivery can be a time-consuming and frustrating process, involving multiple steps and interactions with different parties. Customers often have to call restaurants to place orders, wait on hold, and provide information. Restaurants, in turn, have to manually process orders, manage bulk orders and handle customer inquiries and complaints. This process can be inefficient and error-prone, leading to delays, misunderstandings, and dissatisfied customers.

To address this problem, there is a need for a food delivery app that streamlines the ordering process and provides a convenient and user-friendly experience for both customers and restaurants.

* 1. **Introduction**

Urban Harvest was designed with a hope to emerge as a convenient and accessible way for individuals to seek the services of wide selection of restaurants and menus, place orders. The app is designed to provide a seamless and convenient ordering experience for users, while also helping restaurants to manage orders.

This app is intended to be a scalable and reliable solution for food ordering, with the ability to handle a large number of users and orders without degrading performance or availability. The app is also designed to be easy to use and maintain, with a modular architecture and well-documented codebase.

1. **Methodology**

The methodology section outlines the systematic approach that will be followed develop the Urban Harvest using the Front End development. It details the various stages of the development process, from initial planning and requirements gathering to deployment and maintenance.

**1. Requirements Gathering:**

* Defining the scope and goals of the our platform.
* Gathering requirements by understanding user needs, preferences, and expectations.
* Identify key features such as user registration, menu browsing, adding items to cart, form-submissions and more.

**2. System Design:**

Once the requirements are gathered, the system design phase begins. The design should prioritize user experience (UX) and accessibility while ensuring scalability, security, and maintainability of the system.

**3. Front-end Development:**

Using React.js, the front-end development phase involves translating the design mockups into interactive user interfaces. Components such as forms, tables, and dashboards are developed to facilitate user interactions. The front-end should be responsive, intuitive, and visually appealing, adhering to best practices in web development and accessibility standards.

**4. Testing:**

Throughout the development process, rigorous testing is conducted to ensure the quality and reliability of the Urban Harvest. Unit tests, integration tests, and end-to-end tests are performed to identify and fix any bugs or issues. These tests are conducted with real users to gather feedback and validate the system against the original requirements.

**5. Deployment:**

Once testing is complete and the Urban Harvest is deemed ready for production, it is deployed to a hosting environment using platforms such Vercel, Netlify. Deployment involves configuring servers, setting up environment variables, and deploying application code. Continuous integration and continuous deployment (CI/CD) pipelines

1. **Tools and Technologies:**

**1. React:** A JavaScript library for building user interfaces, providing a modern and

responsive frontend for Tiffin Tray.

**2. Redux:** A predictable state container for managing application state and ensuring

data consistency across the frontend components.

**3. TailwindCSS:** A popular CSS framework for implementing consistent and

visually appealing user interface components.

**4.Firebase:** A standard for securely logging and signing in, used for implementing user authentication and authorization.

**5.Git and Github:** Platform for version control and hosting the website on Vercel, providing scalability, reliability, and security for the deployed application.