

CS3CO40(P): Software Engineering Laboratory	Experiment no- 1
Experiment : To Identify different types of software's with their examples. Also Identify the problem and solution statement for a suggested system of relevance.	

1. Objective (s):

TO IDENTIFY DIFFERENT TYPES OF SOFTWARES WITH THEIR EXAMPLES. ALSO IDENTIFY THE PROBLEM AND SOLUTION STATEMENT FOR A SUGGESTED SYSTEM OF RELEVANCE.

2. Theory:

Software is defined as a collection of programs, data, and documentation that provides the functionality necessary to meet specific user requirements and perform tasks effectively. Software engineering focuses on the systematic, disciplined, and quantifiable approach to the development, operation, and maintenance of software systems.

We study various existing software used in daily life as below.

SNo.	Software Name	Launch Year	Technology Used	Features
1.	Instagram	2010	Frontend - React ,Redux, React Native,TypeScript. Backend - Python,Django, Nginx,PostgreSQL.	<ul style="list-style-type: none"> • A platform to connect with people. • A platform to upload videos and watch them. • Chatting, voice calls, video calls.
2	Whatsapp	2009	Frontend - JavaScript, React, Html/Css Backend - C++, Java, SQLite, XMPP	<ul style="list-style-type: none"> • Chatting, voice calls, video calls. • Business payments • Forming communities • Creating groups • Screen sharing
3	Youtube	2005	Frontend - Html/Css, JavaScript, React, PWA Backend - Python, C++, Java, BigTable, MySQL, Docker,TensorFlow etc	<ul style="list-style-type: none"> • Video uploading • Video streaming • Community posts • Youtube shorts videos
4	Flipkart	2007	Frontend - Html/Css, JavaScript, React,Redux, Angular, Bootstrap Backend - Java,Python, Node.js,Spring,MySQL etc	<ul style="list-style-type: none"> • Buying and selling products. • Flipkart UPI for bill payments and recharges. • Games • Flight and movie tickets

CS3CO40(P): Software Engineering Laboratory	Experiment no- 1
Experiment : To Identify different types of software's with their examples. Also Identify the problem and solution statement for a suggested system of relevance.	

5	Amazon	1994	Frontend - Html/Css, JavaScript, React, Angular, TypeScript, AWS Amplify Backend - Java, Python, Node.js, C++, spring, Django, DynamoDB, RDS ElasticSearch, AWS, Docker, Jenkins etc.	<ul style="list-style-type: none"> • Buying and selling products. • Amazon pay services for bill payments and recharges. • Daily quizzes and games. • Hotel, flight, train tickets booking.
6	Snapchat	2011	Frontend - JavaScript, React, Redux, Swift, Kotlin. Backend - Python, Java, C++, Django, Flask, MySQL, Docker, AWS, GCP etc	<ul style="list-style-type: none"> • Share images and videos. • Chat and create groups • Maintaining streaks • Short videos for entertainment.
7	Zomato	2008	Frontend- Html/Css, JS, React, Redux, React native. Backend - Python, Java, Node.js, Django, Flask, MongoDB, CDN, Docker, AWS etc	<ul style="list-style-type: none"> • Order food online • Zomato dining for booking tables in restaurants • Explore restaurants around the country
8	Swiggy	2014	Frontend - Html/Css, React, Redux, JavaScript, Webpack Backend - Node.js, Express.js, Java, Springboot, Python, MySQL, MongoDB etc	<ul style="list-style-type: none"> • Order food online • Swiggy dineout provides grocery delivery • Book restaurants tables via swiggy dineout
9	Netflix	2007	Frontend-Html/Css, React Js, Redux, Node.js Backend - Java, Spring boot, python, Golang, Redis, MySQL, AWS, Docker.	<ul style="list-style-type: none"> • Provides movies and web streaming services • Provides movies and series renting services
10	Gmail	2004	Frontend - Html/Css, Angular, Js, Web comp. Backend - Java, C++, python, BigTable, GCP, Borg, Kubernetes.	<ul style="list-style-type: none"> • Send and receive emails. • Chatting and communication.

Table: 1 Details of existing software

CS3CO40(P): Software Engineering Laboratory	Experiment no- 1
Experiment : To Identify different types of software's with their examples. Also Identify the problem and solution statement for a suggested system of relevance.	

3. Attendance Management System

Problem Statement: An attendance management system is a tool designed to track and manage student or employee attendance efficiently. The problem it addresses is the difficulty and time-consuming nature of manually recording attendance. In schools and workplaces, keeping accurate records of who is present and who is absent can be challenging and prone to errors. This system automates the process by allowing users to check in and out digitally, reducing the chances of mistakes and saving time. It can generate reports, track patterns, and provide reminders for absences. By using this system, teachers and managers can easily monitor attendance, manage leave requests, and ensure better accuracy in records, ultimately improving efficiency and accountability in tracking attendance. To solve the problem of managing attendance manually, we propose an automated Attendance Management System. This system allows users to record their attendance easily through digital methods, like checking in with a smartphone or using a card. Instead of writing down attendance by hand, the system automatically logs who is present and who is absent, which reduces mistakes and saves time. It can also create reports and send alerts for absences or lateness, making it simple to track and manage attendance. The system stores all data securely and can be accessed from anywhere, providing convenience and flexibility. With this solution, schools and workplaces can handle attendance more accurately and efficiently, making the process faster and more reliable for everyone involved.