Project Report On

**“TODO App”**

Submitted for the partial fulfillment of the requirement for the degree of

**Bachelor of Technology**

in

**COMPUTER SCIENCE & ENGINEERING**

**By**

**Anjali Rani** Reg. No: 1701287291

**Vikash Malakar** Reg. No: 1701287289

**Rakesh kumar** Reg. No: 1701287258

**Adarsh Raj**  Reg No: 1701287262

**Supervised/Guided By**

Prof. Vikash Singh



**GANDHI INSTITUTE FOR TECHNOLOGICAL ADVANCEMENT, BHUBANESWAR**

**NOVEMBER 2020**



**Department of Computer Science & Engineering Gandhi Institute for Technological Advancement, Bhubaneswar**

**Ref no: Date: 25.11.2020**

This is to certify that the project work “TODO App” submitted by:

1. Miss. Anjali Rani Reg No. 1701287291
2. Mr. Vikash Malakar Reg No. 1701287289
3. Mr. Rakesh Kumar Reg No. 1701287285
4. Adarsh Raj Reg. No.1701287262

is an authentic work carried out by them at GITA under my guidance. The matter embodied in this work has not been submitted earlier for the award of any degree or to the best of my knowledge and belief .

**Prof. (Dr.) Tarini Prasad Panigrahy Prof. Vikash Singh**

**(H.O.D Dept. of CSE) (Supervisor)**



**Department of Computer Science & Engineering Gandhi Institute for Technological Advancement, Bhubaneswar**

**ACKNOWLEDGEMENT**

*I express my deepest sense of gratitude to our beloved professor (Late)****Prof.(Dr.)Manoranjan Pradhan****, Ex-Head of Department, Computer Science & Engineering for his initiative and constant inspiration.*

*I also express and gratitude* ***to Prof. Vikash Singh****, project supervisor for his guidance and constant support.*

*I also take this opportunity to thank* **Prof. (Dr.) Tarini Prasad Panigrahy,** *Head of Department, Computer Science & Engineering, for his constant support and timely advice.*

*Lastly, words run to express my gratitude to all the faculties of the CSE Dept. and friends for their support and co-operation, constructive criticism and valuable suggestion during preparation of this project report.*

*Thanking All…..*

(Full signature of the student) (Full signature of the student)

**Anjali RANI Vikash Malakar**

Reg.No: 1701287291 Reg. No: 1701287289

Email: anjalirani289@gmail.com Email: [vikashmalakar275@gmail.com](mailto:shristi.divyasuman17@gmail.com)

(Full signature of the student) (Full signature of the student)

**Rakesh Kumar Adarsh Raj**

Reg. No. 1701287285 Reg. No. 1701287262

Email: [rakeshrai.kumar121@gmail.com](mailto:rakeshrai.kumar121@gmail.com) Email: adarshraj5860@gmail.com

## ABSTRACT

**Title:** TODO App

## Scope & Objectives :

This paper reports on the results of studies of task management to support the design of a task list manager. We examined the media used to record and organize to-dos and tracked how tasks are completed over time. Our work shows that, contrary to popular wisdom, people are not poor at prioritizing. Rather, they have well-honed strategies for tackling particular task management challenges. By illustrating what factors influence task completion and how representations function to support task management, we hope to provide a strong foundation for the design of a personal to-do list manager. We also present some preliminary efforts in this direction.

# TABLE OF CONTENTS

**Chapter 1** : Introduction

Objective of the system Justification and need for the system Advantages of the system

**Chapter 2 :** DEVELOPMENT OF THE SYSTEM

Hardware, Software requirements

System requirements and System specifications

**Chapter 3 :** IMPLEMENTATION & CODING

(coding Should be submitted in CD/DVD)

**Chapter 4 :** TESTING & TESTING RESULTS

**Chapter 5 :** CONCLUSION

## Chapter : 1 INTRODUCTION

TODO App is software in the category of Task Management, Project Management, Productivity, “Getting Things Done” (GTD), Scheduling, and Collaboration.

TODO App is general-purpose, Windows-based software, which can be used for simple “honey do” home lists or to manage complex multi-user projects for business. In addition to tracking the status of tasks, TODO App includes a powerful system for logging and reporting time spent on tasks. Combined with powerful reporting mechanisms, this makes TODO App an effective tool for client billing.

#### OBJECTIVE AND SCOPE OF THE PROJECT

**Objective:**

• Capture the person’s daily tasks.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

• Plan and execute simple actions.

• Prioritize, manage, and reason about tasks.

• Learn to improve by being told, observing the user, asking questions and

reflection.

• Record notes & action items and ideas.

• Answer questions and offer advice and assist in planning and problem

Solving As part of this initiative, an initial effort at our laboratory was undertaken to understand natural practices of task management and types and quantities of tasks taken on by people similar to prospective users of DARPA

technology namely busy professionals and managers. In particular we sought to discover what kinds of task management demands might be supported by a TLM.

## Scope:

To start with, it is better to mention that **a typical TODO App (also called as a list of things, a list of errands, things-to-do list, etc.) is just a list of tasks for a certain period of time, a day or a week**.

Initially, TODO App was created for the purpose to be a well-structured work plan helps every person to see his current load and execute all the necessary tasks. Now it becomes a common practice almost for everyone, and it solves many problems concerning the high workload.

To-do list has its following functions:

* Organization of a proper work with an [accurate planning](https://hygger.io/blog/three-ways-create-project-calendar/" \t "_blank) of every task with its short description
* Simplify the workflow and a timely execution of necessary work scope
* Improvement of every employee’s [productivity](https://hygger.io/blog/4-effective-recommendations-how-to-be-productive-in-project-management/" \t "_blank) and time management skills

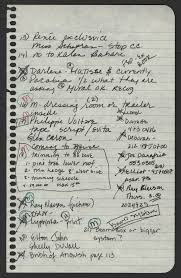
#### JUSTIFICATION AND NEED FOR THE SYSTEM

There are a number of best sellers and tools available on how to organize one’s time and prioritize work . The market for these resources seems to thrive on the fact that many people worry about whether they are prioritizing and meeting their many obligations effectively . Personal information management (PIM; organization, note taking, reminding and calendaring) has also been examined in the HCI literature, but mainly focusing on the problem of organizing documents, files and notes for the purpose of reminding and efficient retrieval, rather than task management . There are a number of studies on how people use calendars in practice (for far more than just event scheduling). But this Project focuses only on a single resource that mainly serves time management needs. And the many readings available on cognition, planning and task execution in the classical psychological literature have little to say about task management and planning in work practice. Distributed cognition analysis comes closest to the kind of analysis we seek here showing that external resources are critical in performing complex tasks,. And task management from an articulation standpoint goes beyond simply organizing physical and virtual collections and putting events in a calendar. Recent work has begun to tackle the challenge of general task management at it plays out in Email . Given that many PC users are overwhelmed by the number of tasks handled through email, this trend is not surprising. This work suggests that any successful task management resource must be closely integrated with email functions. But since email is unlikely to be the entire story, we felt we needed to look at task management practices more broadly.

Some key properties of effective to-dos quickly emerged, a selection of which are discussed in the following:

To-dos are made expending minimal effort, so most of them do not describe the task, they typically are only elaborated enough to provide a salient cue. For example one to-do was some text on a pad of paper; ‘Joe the attorney.’ The explanation was, “A reminder to send him mail. I think I was supposed to ask him about this lawsuit. I can't remember.” Interestingly, to-do text is often not grammatical as in, “Send Mother’s Day” or even “Beth blah blah”. The cue is so minimal that it is only effective for a limited period of time while the task stays in memory. In some cases an item such as a book or a pile serves as enough of a cue to recall the task, without creating a note. Only a minority of to-do reminders appear in lists.

## OVERVIEW:

The online platform/ created for taking online test has following stages :

1. Login
2. Test
3. Subject Selection
4. Login:-

Here there are two login windows present. One for the candidate and another for the recruiter. In this, first we have to make a profile and choose which profile we have to make whether a candidate profile or a recruiter profile .Then, a sign up page will be provided in order to sign in to the account. So, there are two login id’s present but only one login page will be there to login. Hence it is more secure and reliable.

1. Test:-

Test page is the most creative and important page in this project. It consists of 2 modules namely:

* 1. Profile Selection
  2. Resume Selection
  3. Feedback page

1. Subject selection:-

From the given choices the candidate can select his desired company to apply for and the company can also select the candidates or shortlist the candidates on the basis of their skills present in their resume. Here the company can also post the job opportunities offered by their company and the candidate can also select their desired company to apply and can send their resume on the company’s portal

## Advantages of TODO App :

One of the fundamental tools for [time management](http://en.wikipedia.org/wiki/Time_management" \t "_blank" \o "Time management) is that list of things you need to get done. It consolidates all your tasks in one place. From there you can prioritize them and tackle the important ones first.

1] A to-do app doesn’t forget

Your brain is not the most efficient memory tool and will only trust systems that it knows works. Good memory recall is as simple as finding those things that will jog your brain at the time it needs to remember. Having a written list helps us remember when things have do be done so we do not miss anything.

2] A to-do app helps you set priorities

Making a to-do app is an important first step but prioritizing that list ensures that you focus on the most important items rather than giving in to the temptation of working on less important items because they may stand out more or because they are easier to do. Once you have a list of the things you need to complete, set priorities and decide which jobs should be done first.

3] A to-do app lets you coordinate similar tasks

A to-do app helps us to avoid repetition of labour. For example, if we have to deliver a document at an office and collect a document from another office which is on the same block, both these tasks can be done together.

A load of time is lost in the starting, stopping and changing of different levels or types of activity. Save time by performing like tasks together. Make all your outgoing phone calls at the same time; organize your errands into a single run; reply to e-mail; etc. You will find this a more efficient use of your time.

4] A to-do app tracks your progress

Using a to-do app enables you to mark off the tasks you have completed. At the end of the day, when you look at the list, it will give you a sense of accomplishment and satisfaction. It might also have the effect of waking you up if nothing has been marked completed

.

5] A to-do app makes it easy to carry-over tasks

If anything remains incomplete at the end of the day, it can be carried over to tomorrow’s list. This is an easy way of preparing a to-do app for the next day; by examining the to-do app of today and carrying forward any task that is incomplete

#### 6] Brings Structure To The Day

There are two ways to go about one’s day – either be at sea and just wing it with whatever comes or go about with a list of things to be done. I have realized that my peace of mind is higher when I have a plan rather than swimming around accomplishing anything thrown at my direction. Having a to-do app is like having GPS to reach a destination rather than walking down a road not knowing your destination but just setting on paths shouted out by onlookers.

7] Reduces Anxiety

Remember Sheldon in Big Bang Theory who has a knack of doing things a certain way and cannot bear the idea of tasks remaining incomplete? [Take a look at his obsession for yourself in this video clip](https://www.youtube.com/watch?v=r60lVSiF1AA). While he might be the epitome of [zeigarnik effect](http://www.rightattitudes.com/2017/03/14/zeigarnik-effect/), the impact is for real. As discovered by Lithuanian psychologist, Bluma Zeigarnik, waiters remembered even complex orders till the time they were incomplete while failing to recall any detail once an order had been delivered. Probing further, she was able to conclude that our brain focusses much more on incomplete tasks. Incomplete tasks have a way of popping up in our head at unexpected times or to continue brewing in your sub conscious, compounding and leading to anxiety. A daily to-do app helps on cutting down that anxiety.

**Chapter : 2**

**DEVELOPMENT OF THE SYSTEM**

**Proposed:**

The main objective of this to-do app is By keeping such a **list**, you make sure that your **tasks** are written down all in one place so you don't forget anything important. And by prioritizing **tasks**, you plan the order in which you'll do them, so that you can tell what needs your immediate attention, and what you can leave until later.

## Scope:

**Scope** is a simple **to-do list** that will help you in your daily routine. You can easily create **lists** and **tasks**, manage them and receive reminders on time. In the next versions will be added synchronization, creating your own smart **lists**, voice notes, support for Siri and 3D Touch, a group of **lists** and much more.

## SYSTEM REQUIREMENTS

**Software Requirement :**

#### Operating System

Window10 and others

#### Front End

* **Back End**
* **Browser**

HTML, CSS, JavaScript

Node.JS, MongoDB

Internet Explorer and Others

## Hardware Requirement :

#### Client Side:

**Browser**

Any Browser

#### Processor

Intel Core i3 and others

#### Ram

4 GB

**Server Side:**

#### Processor

Intel Core i3 amd others

#### Hard disk space

200GB

## User Requirements :

* Every user should be:

Comfortably work with computer.

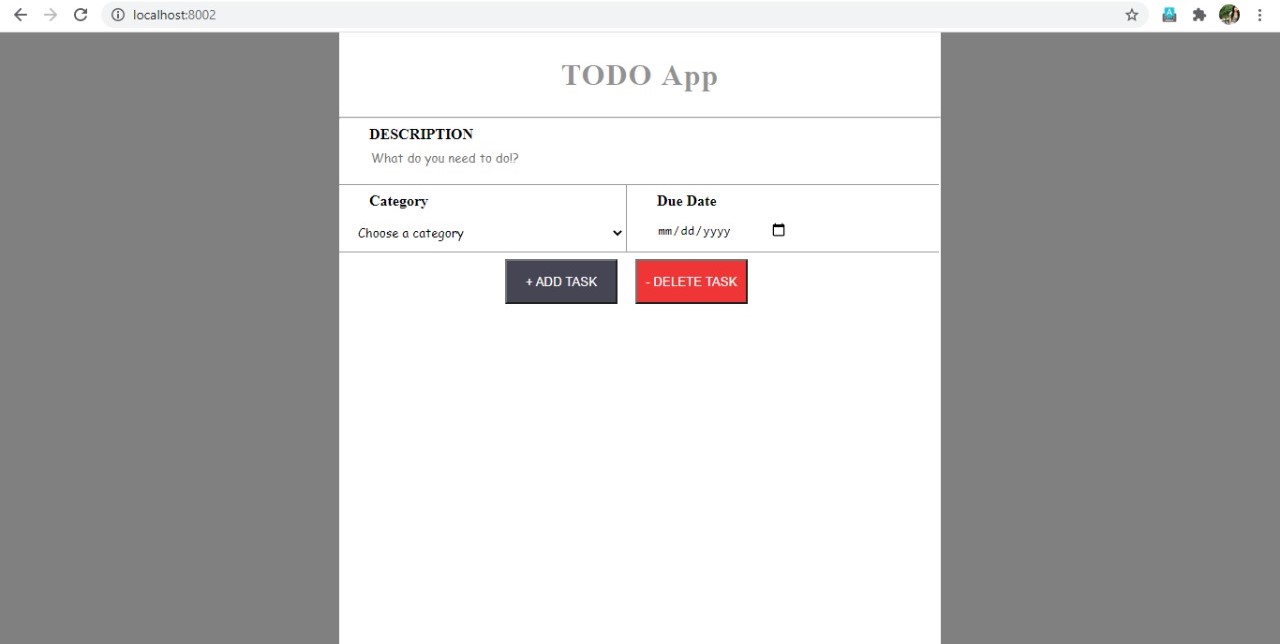
## Constraints :

* Graphical user interface is only in English.
* All users will be authorized to use the services.
* Limited to HTTPS.
* This system is working for single server.

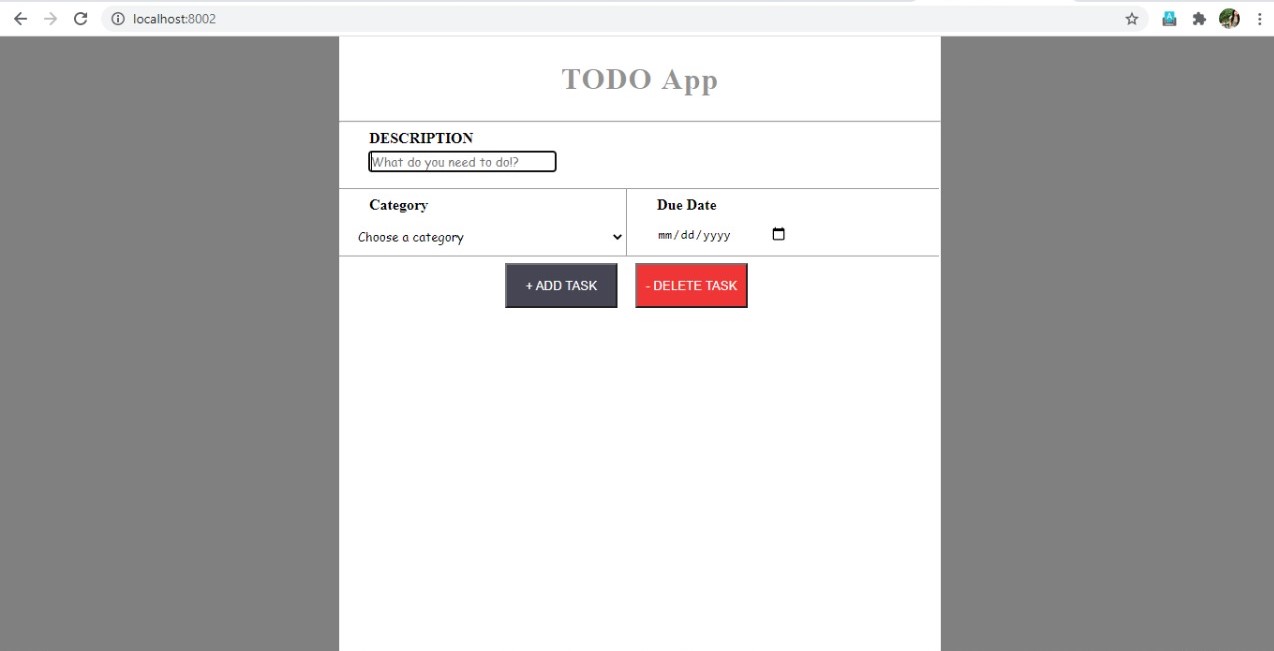
# Chapter : 3

**Implementation & Coding**

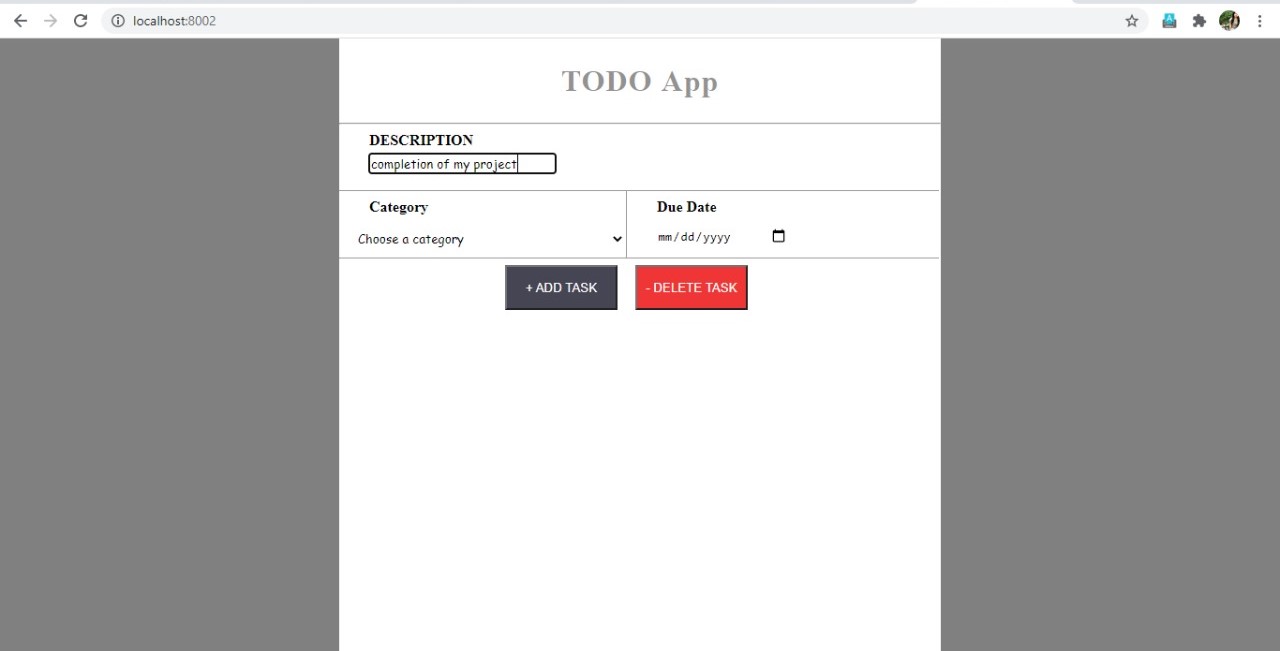
**Home Page**



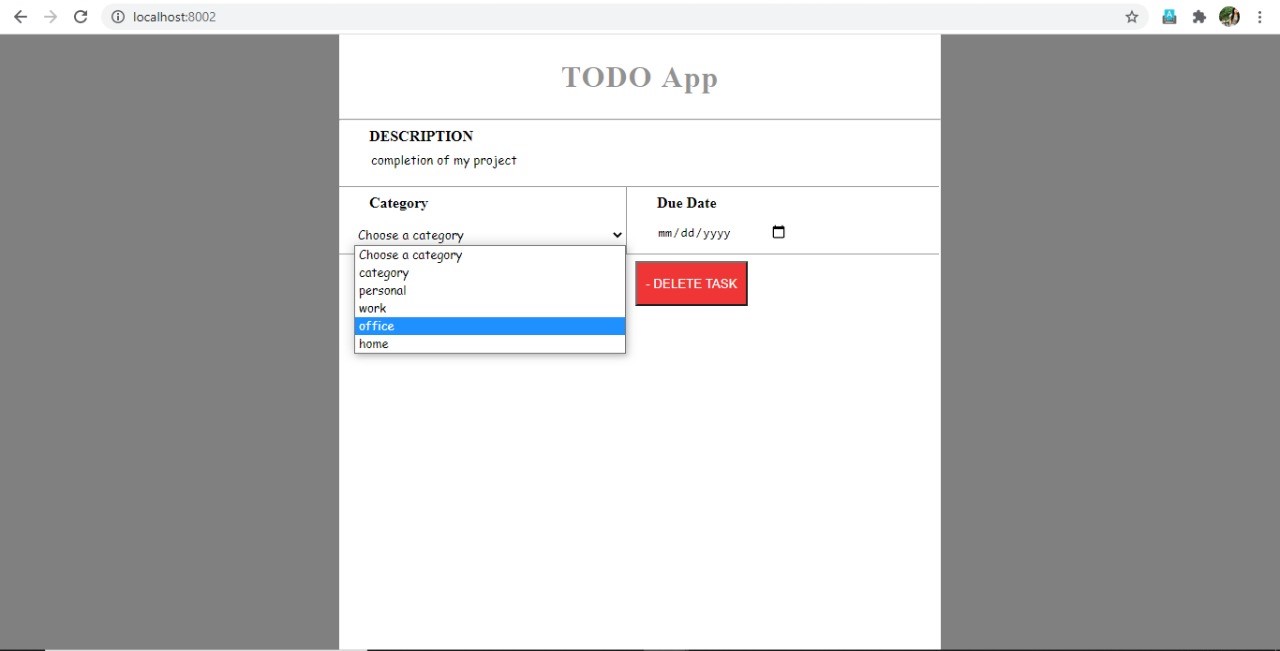
**Select Description Box**



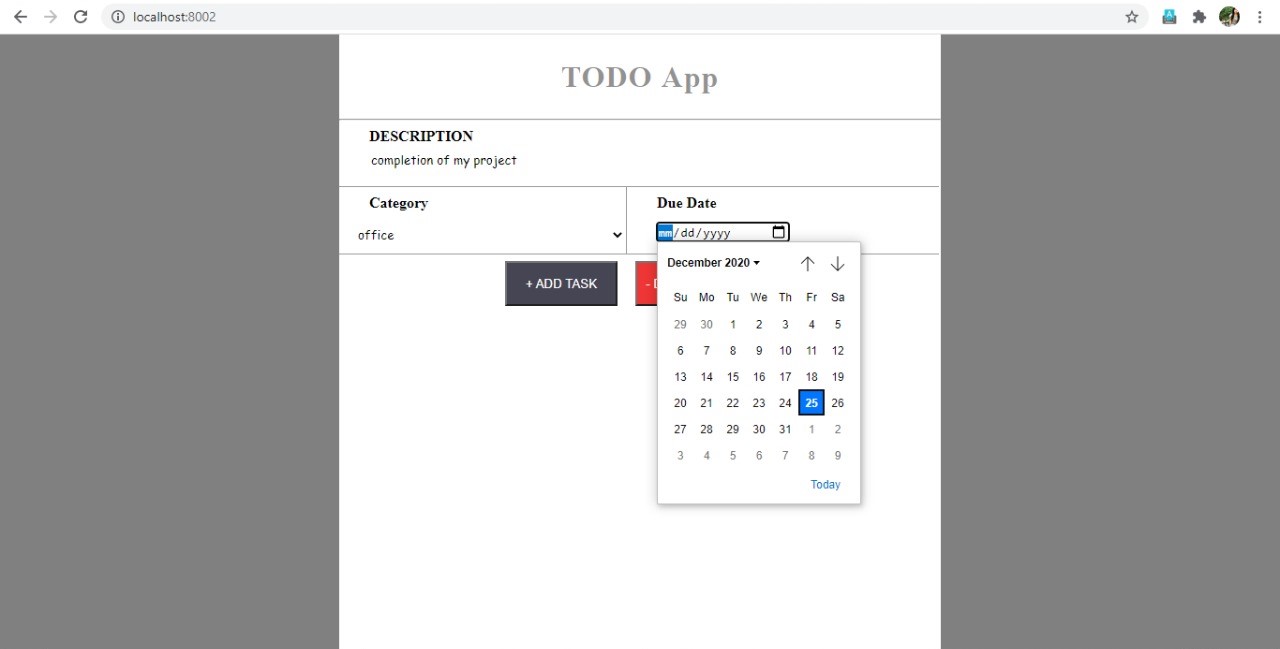
**Adding Description**



**Choose Category**



**Choose Date**

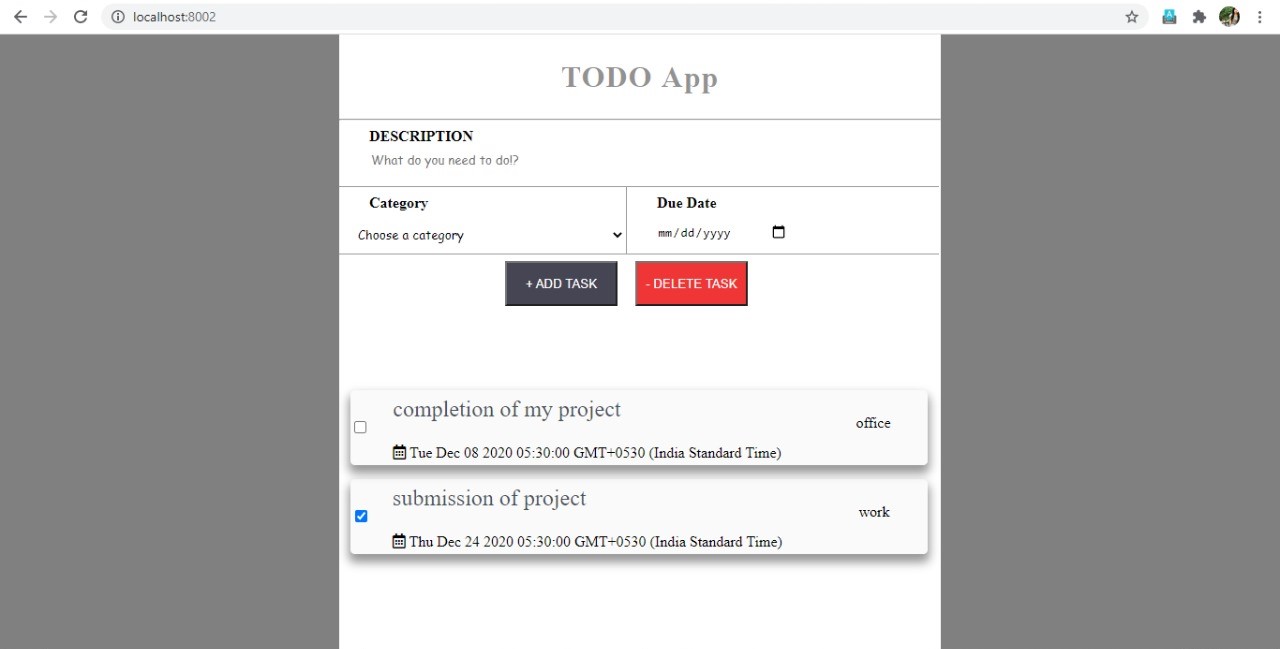


# Add Task

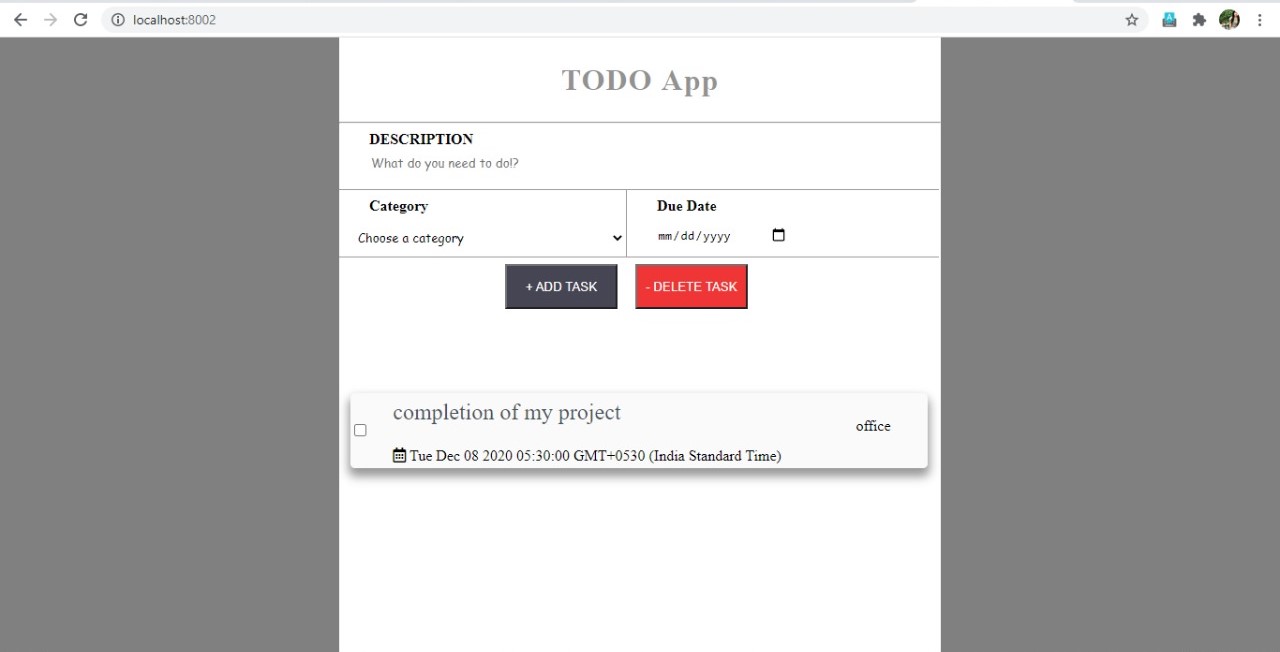
# 6

# 7

**Select Task**



**Delete Task**



### Login Page (Frontend) :

<label class="form-check-label">

<asp: Checkbox ID="chk\_remember" runat="server" CssClass="form- check-input remembermecustom" />

Remember Password</label>

</div>

</div>

<asp: Button ID="btn\_login" runat="server" Text="Log In" CssClass="btn btn- primary btn-block" OnClick="btn\_login\_Click" />

<div class="text-center">

<a class="d-block small mt-3" href="register.aspx">Register an Account</a>

<a class="d-block small" href="fpass.aspx">Forgot Password?</a>

</div>

</form>

</div>

</div>

</div>

</body>

</html>

from django.contrib.admin.sites import AdminSite

class MyAdminSite(AdminSite):

Pass

myadmin = MyAdminSite(name="myadmin")

"""

Django settings for ProHyre project.

Generated by 'django-admin startproject' using Django 2.2.1.

For more information on this file, see https://docs.djangoproject.com/en/2.2/topics/settings/

For the full list of settings and their values, see https://docs.djangoproject.com/en/2.2/ref/settings/ """

import os

# Build paths inside the project like this: os.path.join(BASE\_DIR, ...) BASE\_DIR = os.path.dirname(os.path.dirname(os.path.abspath( file ))) SETTINGS\_PATH = os.path.dirname(os.path.dirname( file ))

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/2.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret! SECRET\_KEY = '1)%(q(b3\_l y^\_i&e(=gjf\_9g(1im(v(u+fei@h@-&rp#38='

# SECURITY WARNING: don't run with debug turned on in production! DEBUG = True

ALLOWED\_HOSTS = ['localhost', 'divya063.pythonanywhere.com', '\*']

# Application definition INSTALLED\_APPS = [

'app',

'resume', 'django.contrib.admin', 'django.contrib.auth', 'django.contrib.contenttypes', 'django.contrib.sessions', 'django.contrib.messages', 'django.contrib.staticfiles',

]

MIDDLEWARE = [

'django.middleware.security.SecurityMiddleware', 'django.contrib.sessions.middleware.SessionMiddleware', 'django.middleware.common.CommonMiddleware', 'django.middleware.csrf.CsrfViewMiddleware', 'django.contrib.auth.middleware.AuthenticationMiddleware', 'django.contrib.messages.middleware.MessageMiddleware', 'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'ProHyre.urls' TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates', 'DIRS': [],

'APP\_DIRS': True, 'OPTIONS': {

'context\_processors': [ 'django.template.context\_processors.debug', 'django.template.context\_processors.request', 'django.contrib.auth.context\_processors.auth', 'django.contrib.messages.context\_processors.messages',

],

},

},

]

WSGI\_APPLICATION = 'ProHyre.wsgi.application'

# Database

# https://docs.djangoproject.com/en/2.2/ref/settings/#databases

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.sqlite3', 'NAME': os.path.join(BASE\_DIR, 'db.sqlite3'),

}

}

# Password validation

# https://docs.djangoproject.com/en/2.2/ref/settings/#auth-password-validators

AUTH\_PASSWORD\_VALIDATORS = [

{

'NAME':

'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

},

]

# Internationalization

# https://docs.djangoproject.com/en/2.2/topics/i18n/ LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'UTC'

USE\_I18N = True USE\_L10N = True USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/2.2/howto/static-files/ AUTH\_USER\_MODEL = 'app.User'

STATIC\_ROOT = os.path.join(os.path.dirname(BASE\_DIR), 'static')

STATIC\_URL = '/static/' LOGIN\_URL = 'login'

LOGOUT\_URL = 'logout' LOGIN\_REDIRECT\_URL = 'home' LOGOUT\_REDIRECT\_URL = 'home1'

### Login Page (Backend) :

"""ProHyre URL

Configuration

The `urlpatterns` list routes URLs to views. For more information please see: https://docs.djangoproject.com/en/2.2/topics/http/urls/

Examples:

Function views

* 1. Add an import: from my\_app import views
  2. Add a URL to urlpatterns: path('', views.home, name='home') Class-based views

1. Add an import: from other\_app.views import Home
2. Add a URL to urlpatterns: path('', Home.as\_view(), name='home') Including another URLconf
3. Import the include() function: from django.urls import include, path
4. Add a URL to urlpatterns: path('blog/', include('blog.urls')) """

from django.contrib import admin from django.urls import include, path from ProHyre.admin import myadmin

from django.views.generic.base import TemplateView from django.conf.urls import handler404, handler500 from app import views

**Chapter: 4**

**Software \_Testing :**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive. A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

The following are the Testing Objectives:

* Testing is a process of executing a program with the intent of finding an error.
* A good test has a high probability of finding an as yet undiscovered error.
* A successful test is one that uncovers an as yet undiscovered error.

#### Design of test cases &scenario:

The objective is to design tests that systematically uncover different classes of errors and do so with a minimum amount of time and

effort. Testing cannot show the absence of defects, it can only show that software defects are present.

#### Integration Testing:

Integration Testing Modules integrated by moving down the program design hierarchy. Can use depth first or breadth first top down integration verifies major control and decision points early in design process. Top-level structure tested most. Depth first implementation allows a complete function to be implemented, tested and demonstrated and does depth first implementation of critical functions early. Top down integration forced (to some extent) by some development tools in programs with graphical user interfaces. Begin construction and testing with atomic modules (lowest level modules).Bottom up integration testing as its name implies begins construction and testing with atomic modules. Because modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated.

#### Validation Testing:

Validation testing is aims to demonstrate that the software functions in a manner that can be reasonably expected by the customer. This tests conformance the software to the Software Requirements Specification.

#### Alpha and Beta Testing

Alpha testing it’s an acceptance testing conducted by the developed environment.

Beta Testing it’s an acceptance testing conducted by the multiple customers in the customer environment.

**System Testing:**

Software is only one component of a system. Software will be incorporated with other system components and system integration and validation test performance. Validation aims to demonstrate that the software functions in a manner that can be reasonably expected by the user. An experiment has done for checking the consistency for the user requirements regarding the username and password which should be validated.

## Chapter:5

**Conclusion :**

Ensure consistency in the recruitment and selection of candidates without any discrimination.

* Open communication between the candidates and the recruiters/ interviewers
* An effective feedback system that enables two-way free communication
* Avoids typical evaluations of candidates that may be filled with ambiguity, superficial statements, and generalizations.
* A follow up system which ensures conversations don’t slip through the cracks. A system that will send a reminder if a recruiter doesn’t follow up within a stipulated time frame.
* Pre-interview and post-interview feedbacks.