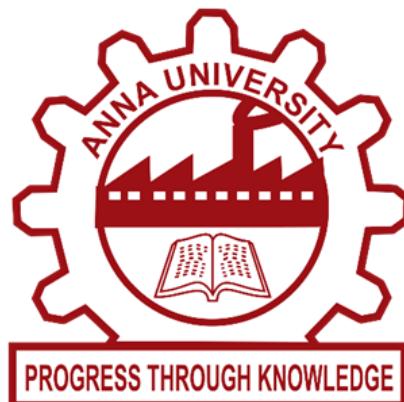


CS6308 - JAVA PROGRAMMING

PROJECT TITLE : TODO PLANNER

HEMASHIRISHA N - 2019103525

KANISHKAA R - 2019103534



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,
COLLEGE OF ENGINEERING GUINDY CAMPUS,
ANNA UNIVERSITY,CHENNAI-25**

CONTENTS

| TOPIC | PAGE NO |
|--------------------------------|-----------|
| 1. ABSTRACT | 3 |
| 2. INTRODUCTION | 4 |
| 3. ARCHITECTURE DIAGRAM | 10 |
| 4. MODULES DESIGN | 11 |
| 5. MODULE 01 | 12 |
| 6. MODULE 02 | 19 |
| 7. MODULE 03 | 25 |
| 8. MODULE 04 | 32 |
| 9. MODULE 05 | 35 |
| 10. RESULTS | 36 |

1. ABSTRACT

Time management can be seen as something which is obvious to anyone but studies show us that most people find it more difficult to manage time than anything else in the world, including money. The reason behind it could be Planning Fallacy, lack of understanding of the tasks at hand or ignorance of clock ticking away and there could be many more reasons pertaining to individual level.

Time management has become a very crucial part of all our lives owing to the ever increasing number of disturbances and distractions around us. It has become very easy to lose focus and hence fall behind in the invisible race. Improper time management leads to build up of tasks later and hence increased pressure , poor results and decreased productivity.

First step to efficiently manage time is to set goals with achievable deadlines. It is important to have a clear idea of the tasks, their priorities and what should be done to complete the tasks. But it's almost impossible for people to remember all the tasks and deadlines. It decreases the productivity as people use most of their energy and memory in remembering tasks and deadline rather than focusing and completing the task at hand.

And this is where the TODO Planner comes to help. TODO Planner provides a medium to help organise tasks, set priorities and deadlines, and schedule the days accordingly. It is just one click away from accessing anytime and anywhere.

2. INTRODUCTION

TODO-Planner has features that help the users in scheduling and organising their work. Using the TODO planner allows you to schedule each event, appointment, errand, and task, so that you know what to expect and don't run out of time. Set specific due dates and deadlines for everything to help you stay on track.

The features of the TODO-Planner comprises of:

- **TODO List:**
 - Todo Today: Tasks for the day
 - Todo Weekly: Tasks for the next 7 days
 - Todo Monthly: Tasks for the next 30 days
 - Others: Previously incomplete tasks and tasks with deadline after 30 days
- **Custom TODO:**
 - Many tasks include many other subtasks to be completed
 - Custom tasks can be created.
 - Subtasks can be added to each custom task
- **Calendar:**
 - Provides summary of tasks and events for the selected day
 - On selecting a day on the calendar, the tasks which were scheduled for that day are displayed
 - Events for that particular day can also be added
- **Stats:**
 - Home page of the user contains the statistics of their activity for the last 15 days on the webapp.

For the tasks, the user will be able to assign priorities and deadlines which will help them in scheduling their day accordingly.

Technologies used to develop the webapp:

Front end: **HTML, CSS, JavaScript**

Backend: **JSP, SERVLETS, JDBC**

Database: **MySQL**

FRONT END:-

1. HTML

The HyperText Markup Language is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

2. CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

3. JAVASCRIPT

JavaScript often abbreviated JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. Over 97% of websites use JavaScript on the client side for web page behavior often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices.

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

BACKEND:-

1. JSP

It stands for Java Server Pages. It is a server side technology. It is used for creating web application. It is used to create dynamic web content. In this JSP tags are used to insert JAVA code into HTML pages. It is an advanced version of Servlet Technology. It is a Web based technology helps us to create dynamic and platform independent web pages. In this, Java code can be inserted in

HTML/ XML pages or both. JSP is first converted into servlet by JSP container before processing the client's request.

JSP allows Java code and certain predefined actions to be interleaved with static web markup content, such as HTML. The resulting page is compiled and executed on the server to deliver a document. The compiled pages, as well as any dependent Java libraries, contain Java bytecode rather than machine code. Like any other .jar or Java program, code must be executed within a Java virtual machine (JVM) that interacts with the server's host operating system to provide an abstract, platform-neutral environment.

JSPs are usually used to deliver HTML and XML documents, but through the use of OutputStream, they can deliver other types of data as well.

2. SERVLETS

Servlet technology is used to create a web application (resides at server side and generates a dynamic web page). Servlet technology is robust and scalable because of java language. Before Servlet, CGI (Common Gateway Interface) scripting language was common as a server-side programming language. There are many interfaces and classes in the Servlet API such as Servlet, GenericServlet, HttpServlet, HttpServletRequest, HttpServletResponse, etc.

Servlets provide a component-based, platform-independent method for building Webbased applications, without the performance limitations of CGI programs. Servlets have access to the entire family of Java APIs, including the JDBC API to

access enterprise databases.

3. JDBC

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database, and is oriented toward relational databases. A JDBC-to-ODBC bridge enables connections to any ODBC-accessible data source in the Java virtual machine (JVM) host environment.

JDBC allows multiple implementations to exist and be used by the same application. The API provides a mechanism for dynamically loading the correct Java packages and registering them with the JDBC Driver Manager. The Driver Manager is used as a connection factory for creating JDBC connections.

JDBC connections support creating and executing statements. These may be update statements such as SQL's CREATE, INSERT, UPDATE and DELETE, or they may be query statements such as SELECT. Additionally, stored procedures may be invoked through a JDBC connection.

DATABASE:-

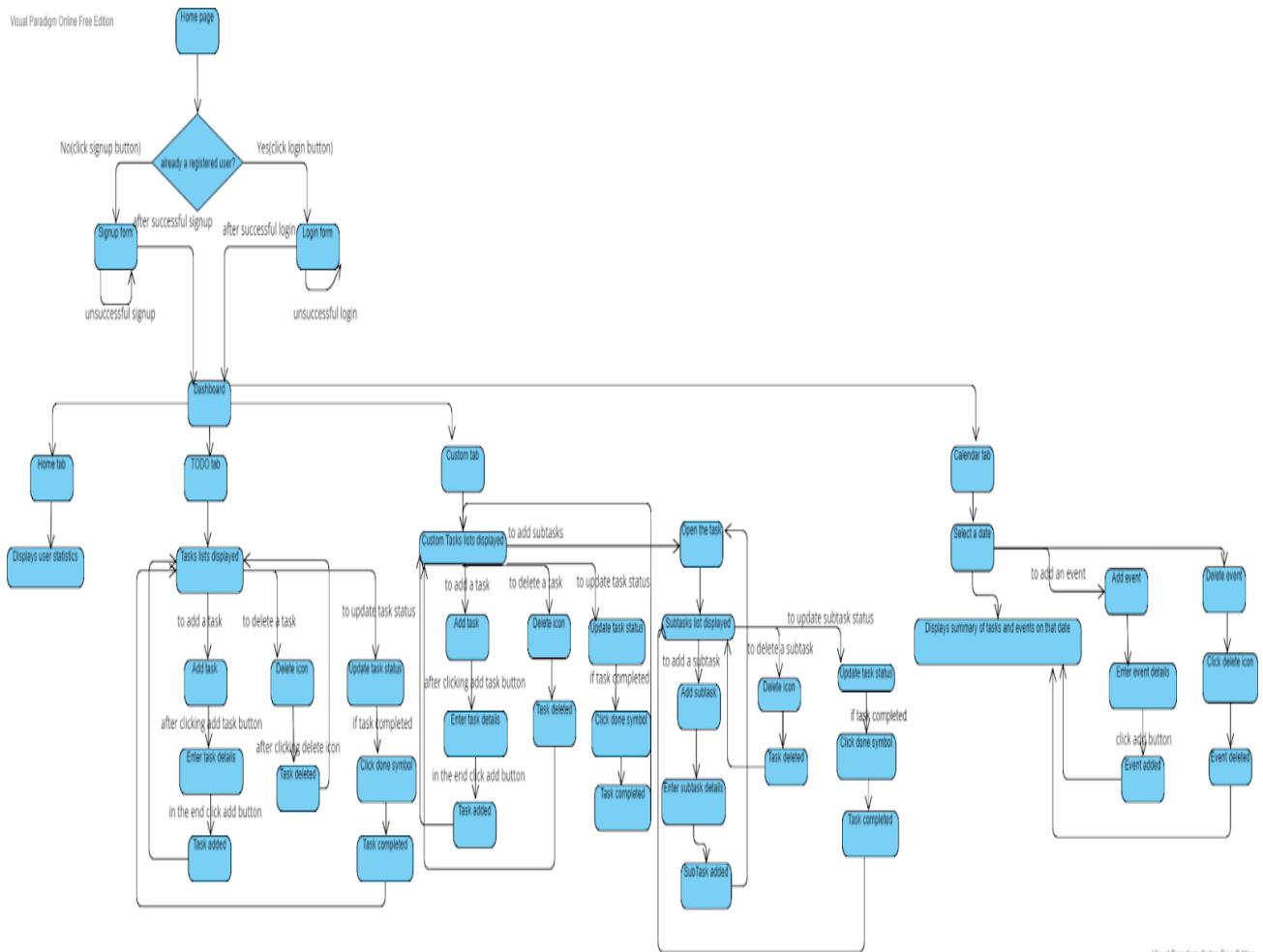
1. MYSQL

MySQL is currently the most popular database management system software used for managing the relational database. It is open-source database software, which is supported by Oracle Company. It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise applications. MySQL is a Relational Database Management System (RDBMS) software.

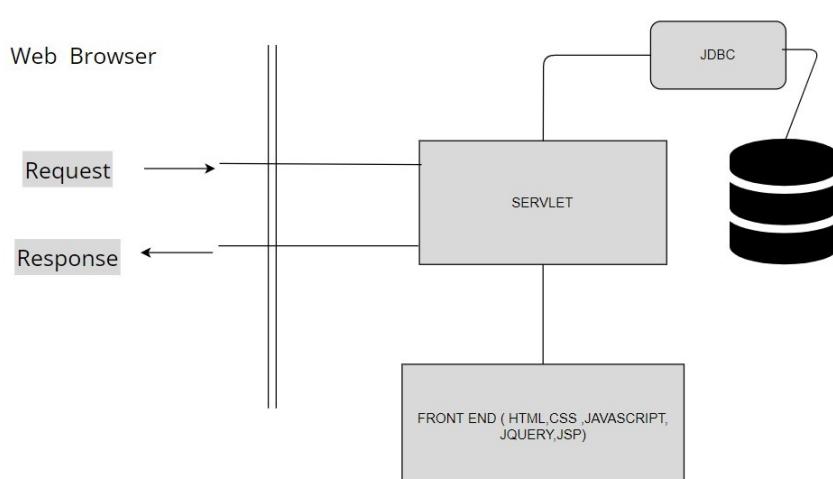
MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python.

MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook, Flickr, MediaWiki, Twitter, and YouTube.

3. ARCHITECTURE DIAGRAM



SYSTEM ARCHITECTURE DESIGN:-



4. MODULES DESIGN

MODULE 1: *HOME PAGE*

- Design of landing page
- Design login and signup forms and link them to landing page
- Create and set session variables for users while signing up and logging in
- Add backend logic to login and signup forms

MODULE 2: *TODO TASKS*

- Design common task list display
- Add backend logic for adding, deleting and updating tasks
- Sort tasks into 4 criteria - Today, Weekly, Monthly, Others

MODULE 3: *CUSTOM TASKS*

- Design display for custom tasks and subtasks
- Add backend logic for adding, deleting and updating custom tasks
- Add backend logic for adding, deleting and updating subtasks for a particular custom task

MODULE 4: *CALENDAR*

- Design calendar display and task,event lists.
- Add backend logic for adding, deleting and updating events

MODULE 5: *USER STATS*

- Design display for graph
- Add logic to get the percentage of completed tasks of every day for the last 15 days.

5. MODULE 1

HOME PAGE

It is this page the user lands in when the URL for the webapp is typed. The home page consists of forms for user login and signup. It also contains information which gives the user an idea what the webapp is about.

When the user signs up, in the backend, it's checked if the username is already present. If it's present, it displays an error message saying the username is taken. If the username is available, the user's data is inserted into the database and is redirected to the user's home page.

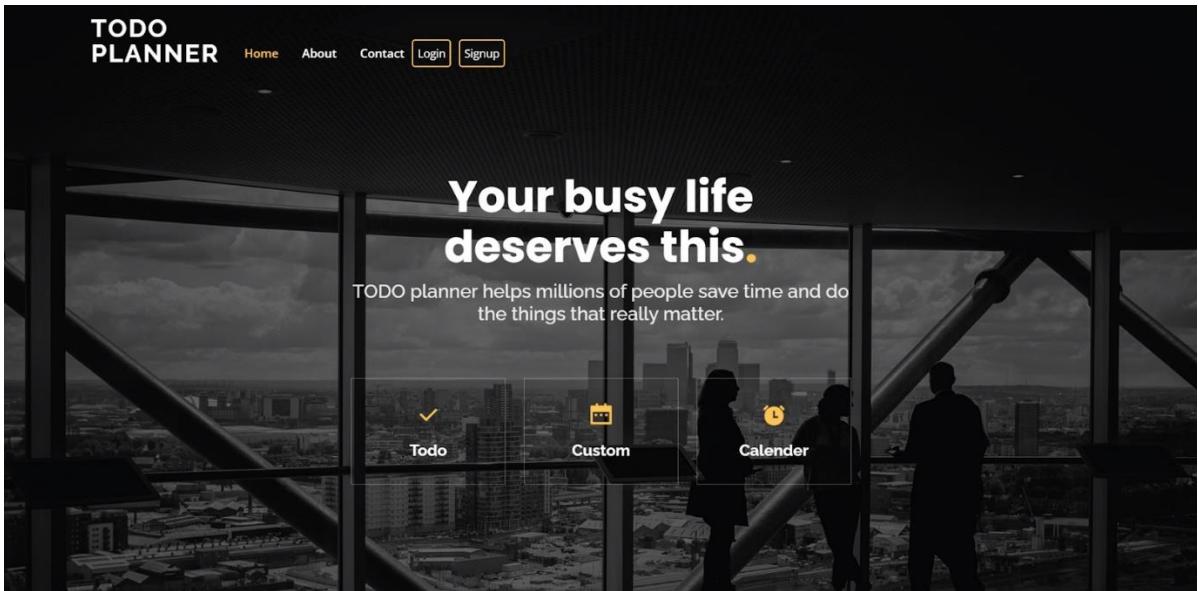
When the user logs in, in the backend, it's checked if the username and password are correct. If wrong, an error message saying invalid login is displayed. If correct, the user is redirected to the user's home page.

On a user's successful signup or login, the session variables to uniquely identify the user are set to allow the user to navigate through the pages meant for the user without having to login to access every page.

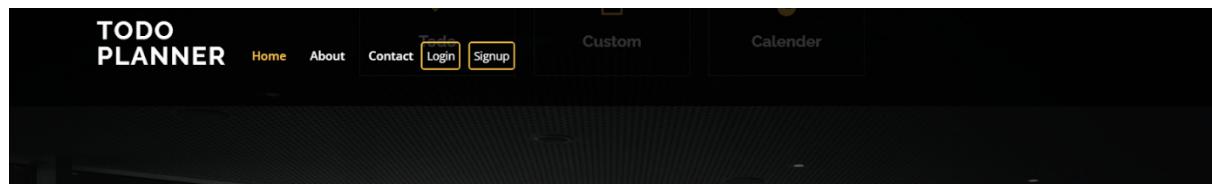
- Landing page:

Tech stack used: HTML, CSS JavaScript

The top page:



About page:



The best way to organise your life.

- ✓ Organize all your to-do's in lists and projects..
- ✓ Color tag them to set priorities and categories.
- ✓ Boost your productivity with notes, subtasks and attachments.
- ✓ Set due dates to keep track of your progress..

Eat deadlines for breakfast

Your work day consists of meetings and action items. Having a combined view accessible from your browser is the best way to stay productive.

About page:

The screenshot shows the 'About' section of the Todo Planner website. At the top, there's a dark header bar with the 'TODO PLANNER' logo and navigation links for 'Home', 'About', 'Contact', 'Login', and 'Signup'. Below the header, there's a large dark banner with a grid pattern. On the left side of the banner, the 'TODO PLANNER' logo and navigation links are repeated. To the right of the grid, there are four yellow icon boxes with text: 1. 'Eat deadlines for breakfast' (with a notepad icon) - 'Your work day consists of meetings and action items. Having a combined view accessible from your browser is the best way to stay productive.' 2. 'Run your day care-free' (with a hexagon icon) - 'Create reminders with natural language and never forget a thing. Easily add reminders to make sure nothing important slips away.' 3. 'Plan your day in seconds.' (with a square icon) - 'Don't let your tasks overwhelm you. With our ultimate daily planner, you can quickly run through your daily to-do's and prioritize them so you're focused only on the things that matter.' 4. 'Never miss a thing' (with a shield icon) - 'Control your day, week and month with calendar events and tasks in a single view. It's the best way to manage your time and achieve your ambitious goals.' A small upward arrow icon is in the top right corner of the banner.

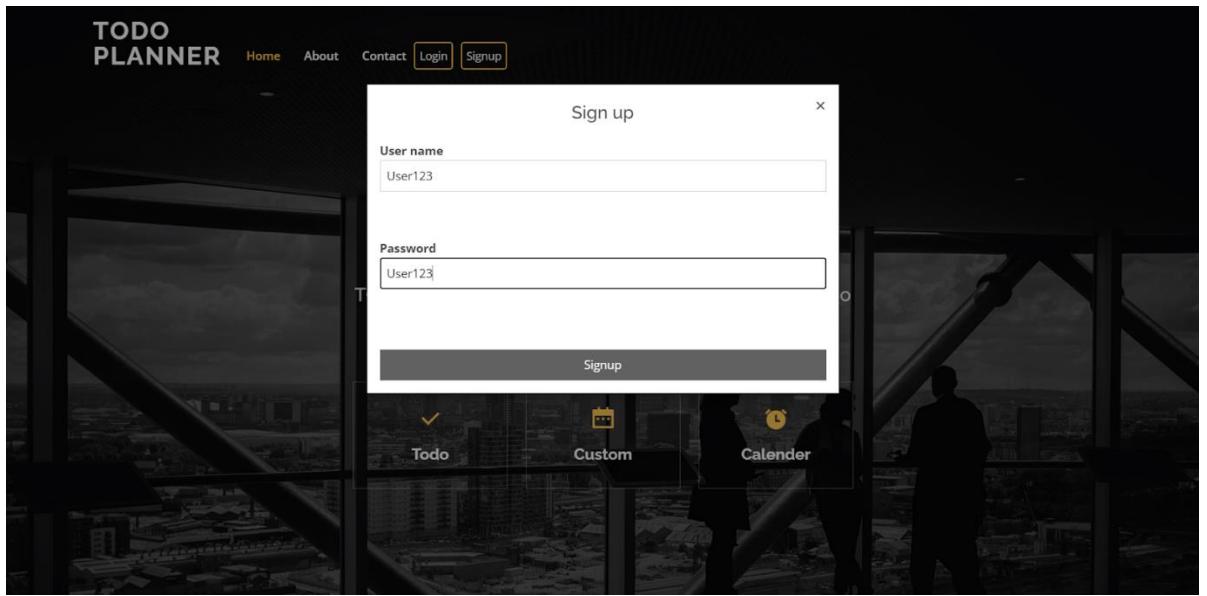
Contact page:

The screenshot shows the 'Contact' page of the Todo Planner website. At the top, there's a dark header bar with the 'TODO PLANNER' logo and navigation links for 'Home', 'About', 'Contact', 'Login', and 'Signup'. Below the header, there's a map of the Guindy area in Chennai, showing various landmarks like ITC Grand Chola, Madras Race Club, and Gandhi Mandapam. To the left of the map, there are three contact information sections: 1. 'Location:' with an info pin icon, followed by the address '12, Sardar Patel Rd, Anna University, Guindy, Chennai, Tamil Nadu 600025'. 2. 'Email:' with an envelope icon, followed by the email 'todo@gmail.com'. 3. 'Call:' with a phone icon, followed by the phone number '+91 9876543210'. To the right of these sections are input fields for 'Your Name', 'Your Email', 'Subject', and a large 'Message' area, all enclosed in a light gray form. A 'Send Message' button is at the bottom right of the form. At the very bottom of the page, there's a black footer bar with the text '© Copyright TODO. All Rights Reserved' and 'Designed by CEGIANS'.

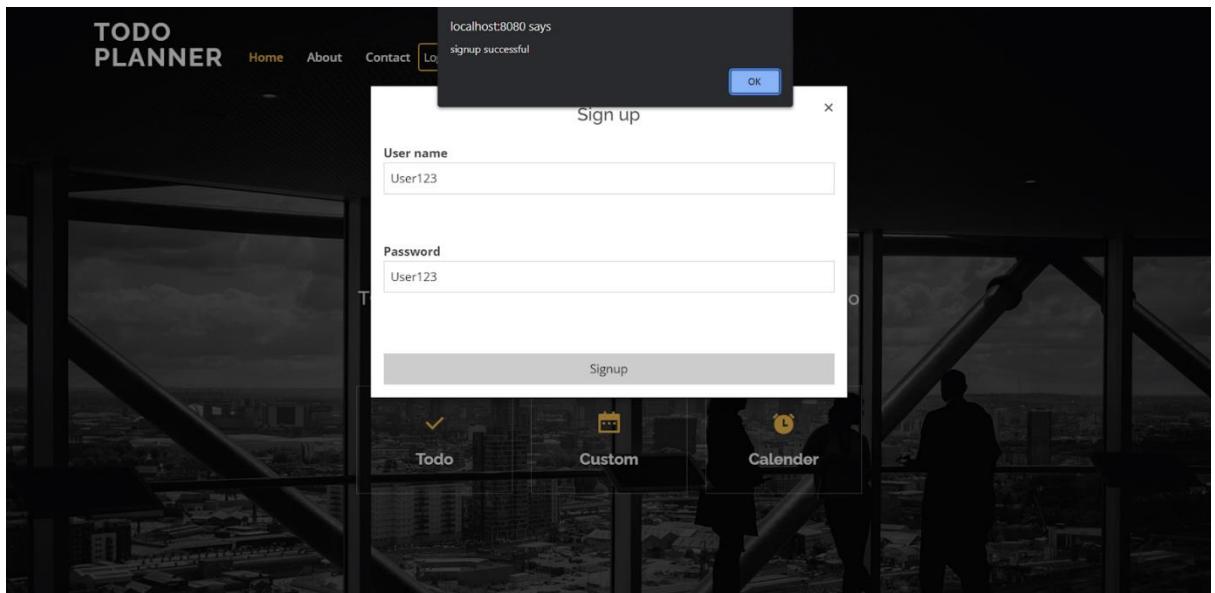
- Sign up form:

Tech stack used: HTML, CSS, JavaScript, jQuery, AJAX, Servlets, Java, MySQL

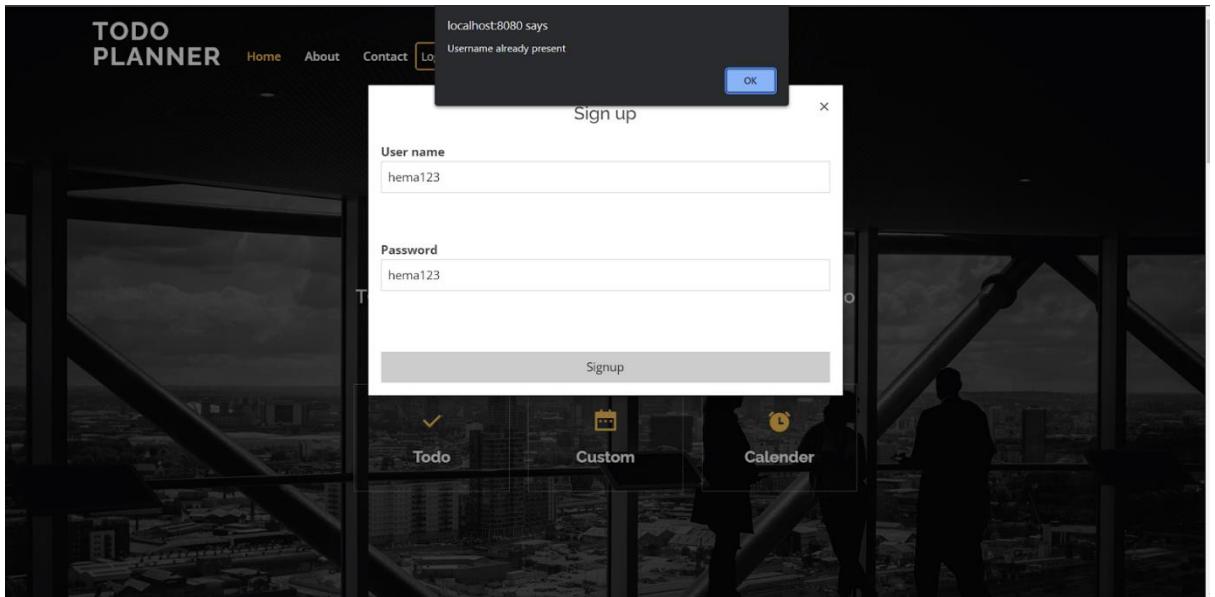
Signup form:



Successful signup:



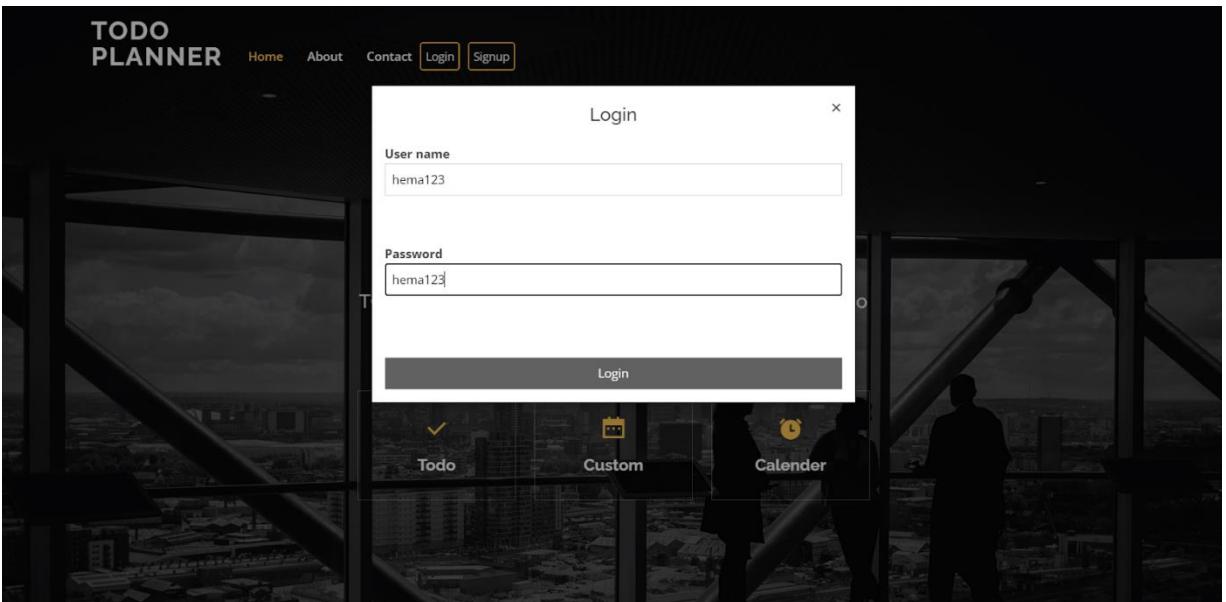
Unsuccessful signup:



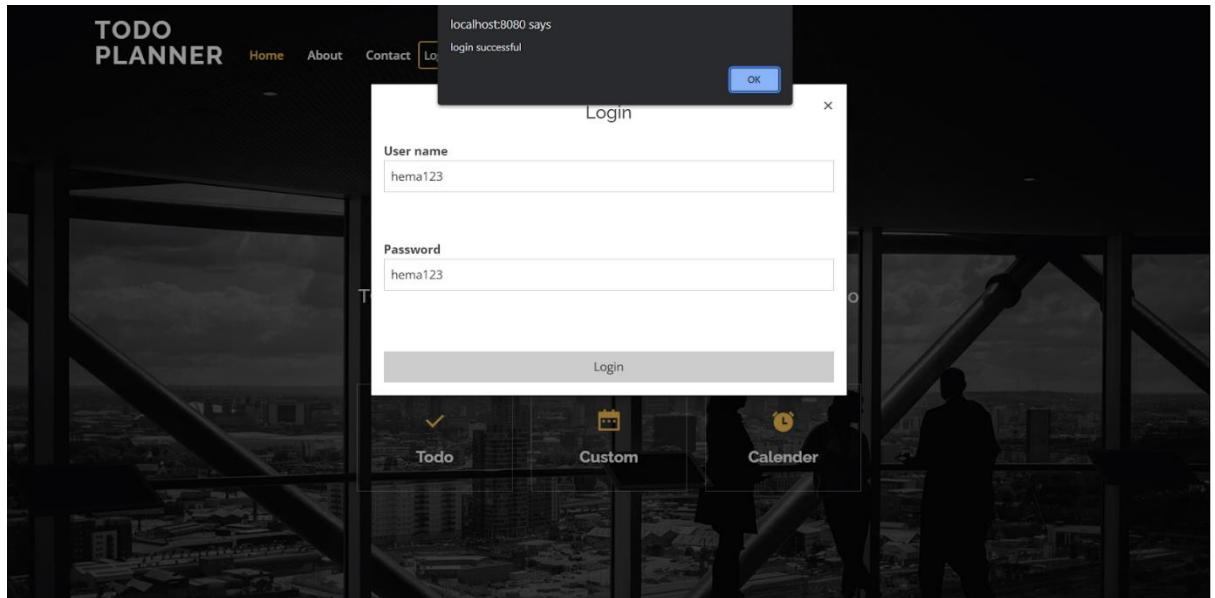
- Login form:

Tech stack used: HTML, CSS JavaScript, jQuery, AJAX, Servlets, Java, MySQL

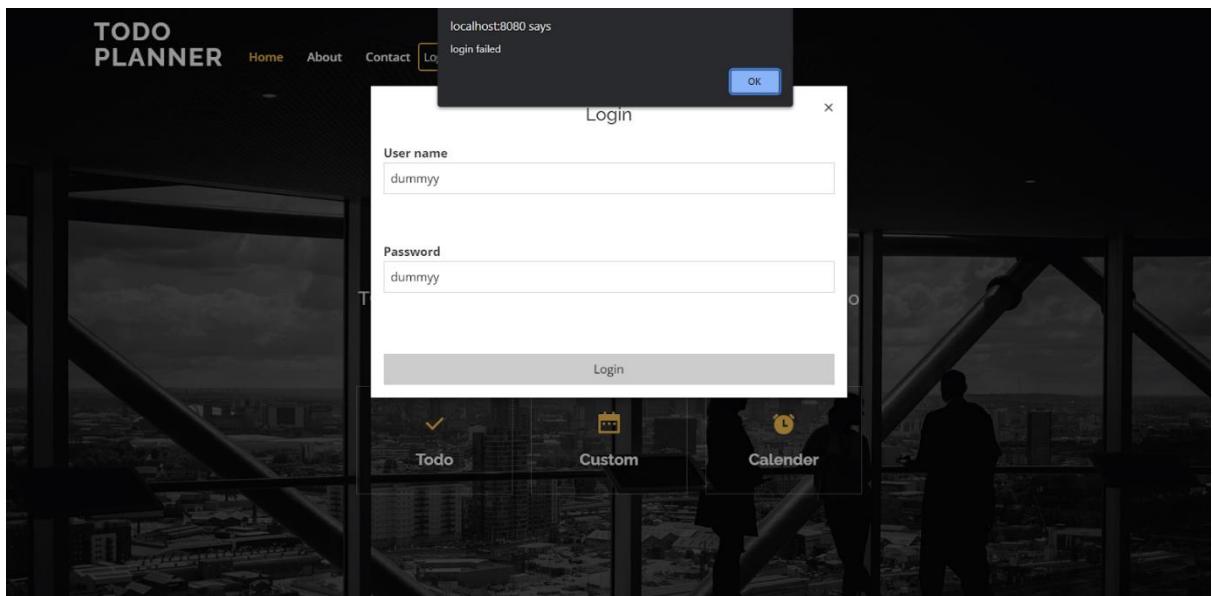
Login form:



Successful login:



Unsuccessful login:



- User class:

The User class is present in the bean package with attributes user_name, user_id and user_psw and contains corresponding public getter and setter methods. The user_id is the primary key and is labelled auto increment.

```
1 package bean;
2
3 public class User {
4     String user_name;
5     int user_id;
6     String user_psw;
7     public int getUser_id() {
8         return user_id;
9     }
10    public void setUser_id(int i) {
11        user_id = i;
12    }
13    public String getUser_name() {
14        return user_name;
15    }
16    public void setUser_name(String s) {
17        user_name=s;
18    }
19    public String getUser_psw() {
20        return user_psw;
21    }
22    public void setUser_psw(String s) {
23        user_psw=s;
24    }
25 }
```

6. MODULE 2

TODO TASKS

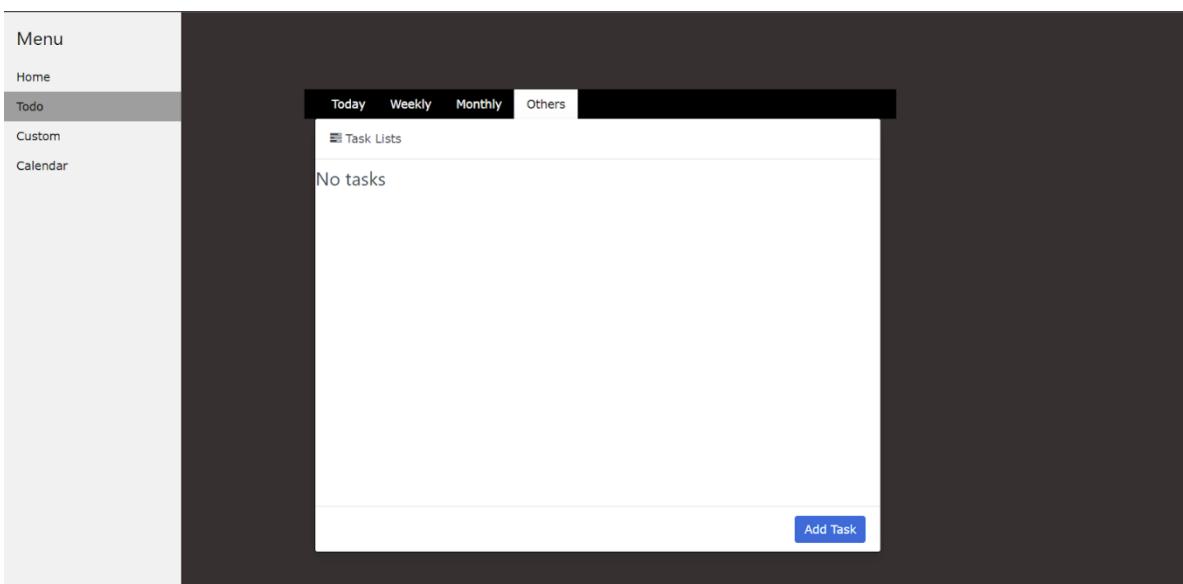
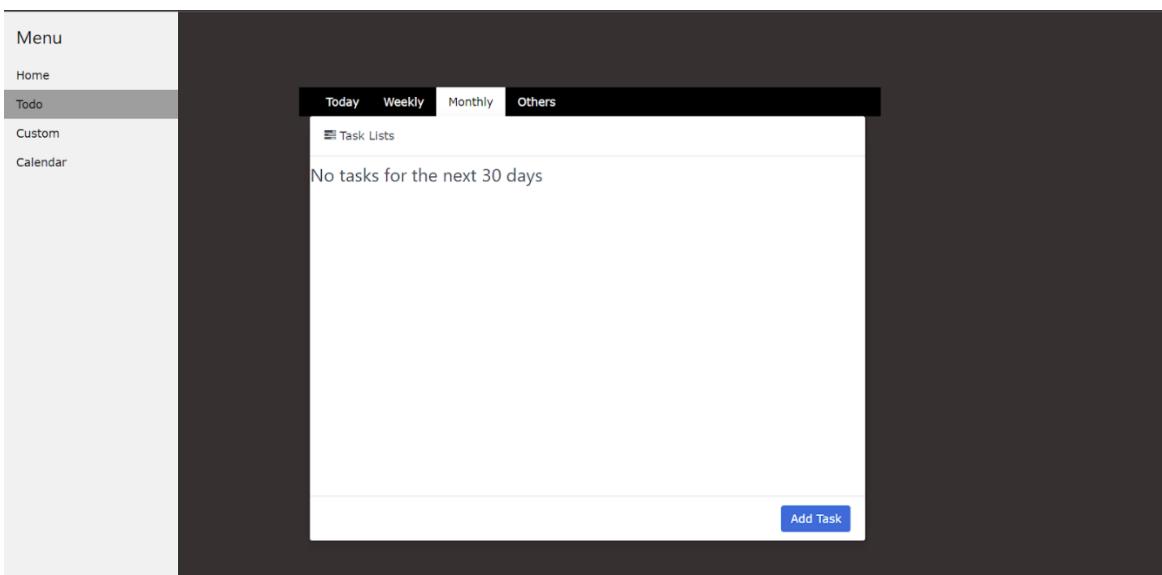
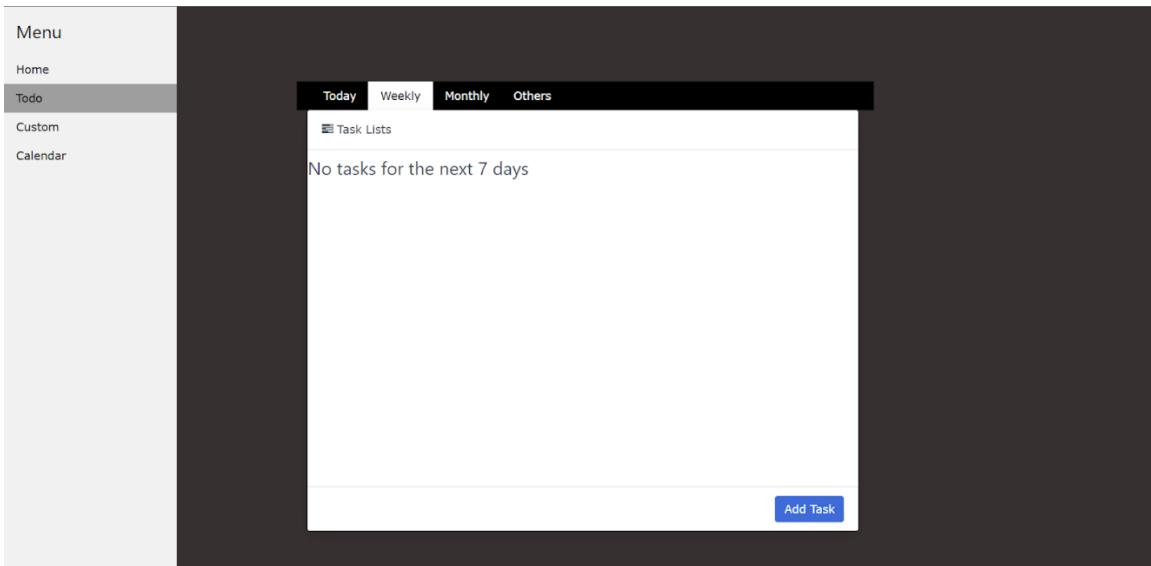
This page will be accessible only after the user successfully signs up or logs in. When the user signs up, the tasks list will be empty. The user will be able to add tasks along with priority and deadline.

The tasks added will be sorted based on the deadlines. The tasks will be sorted into 4 categories namely Today (tasks having deadline today), Weekly (Tasks having deadline in the next 7 days), Monthly (Tasks having deadline in the next 30 days), Others (Tasks incomplete within the deadline and those Tasks with deadline after 30 days).

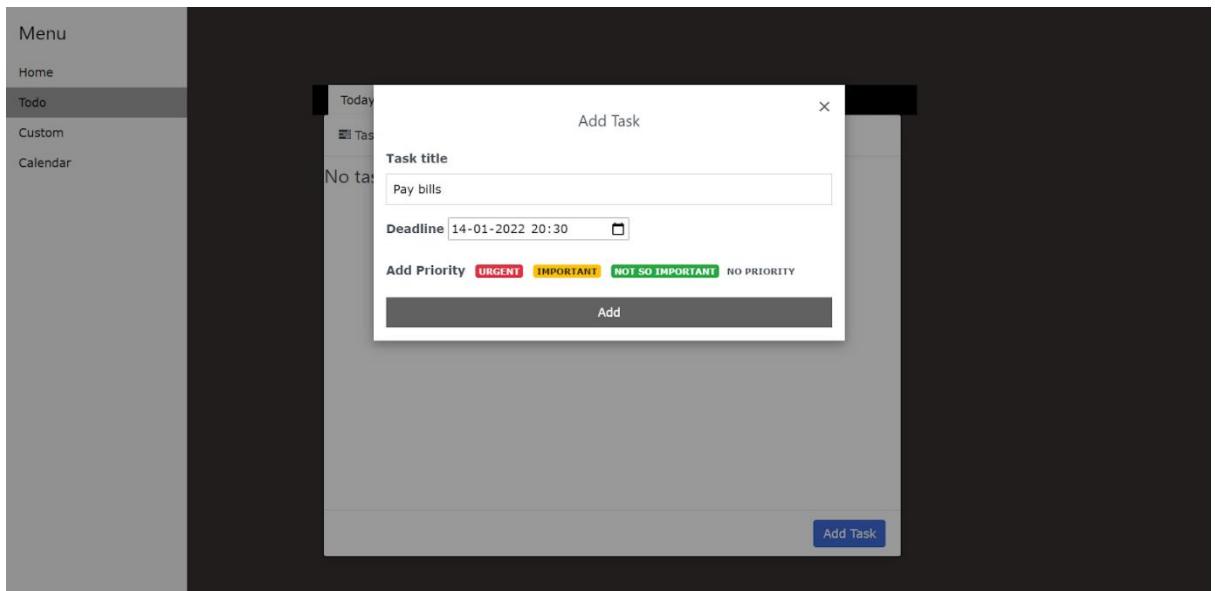
- *Task list:*

Technologies used: HTML, CSS, JavaScript, jQuery, AJAX, Servlets, JSP, Java, MySQL

When the tasks list are empty:

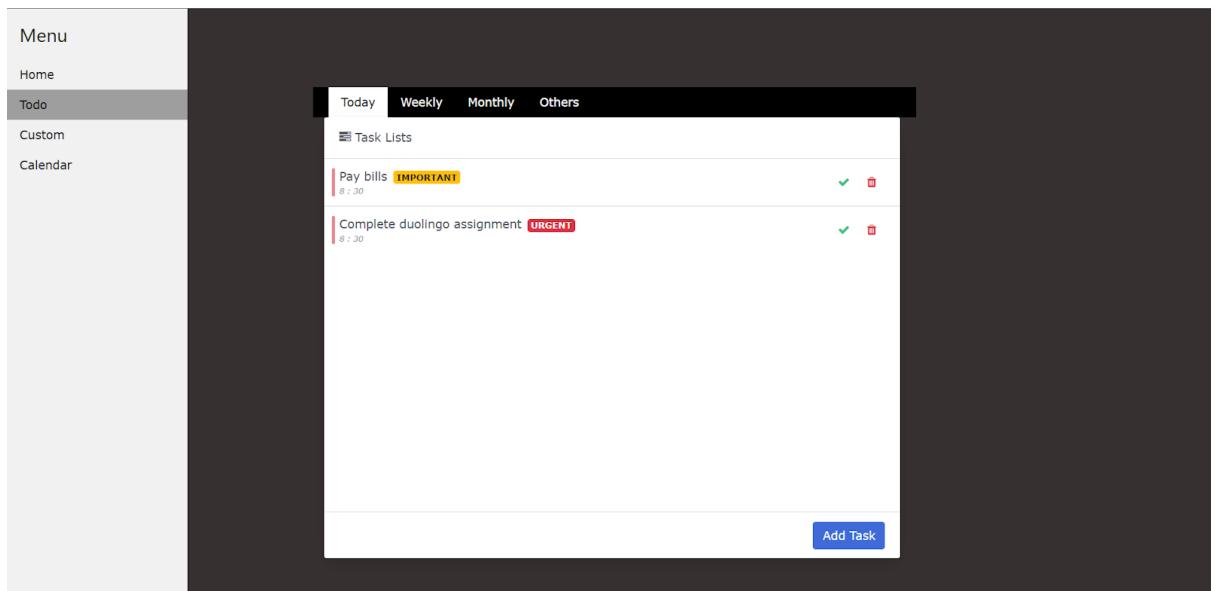


Adding task:



Tasks when not empty:

TODO TODAY:



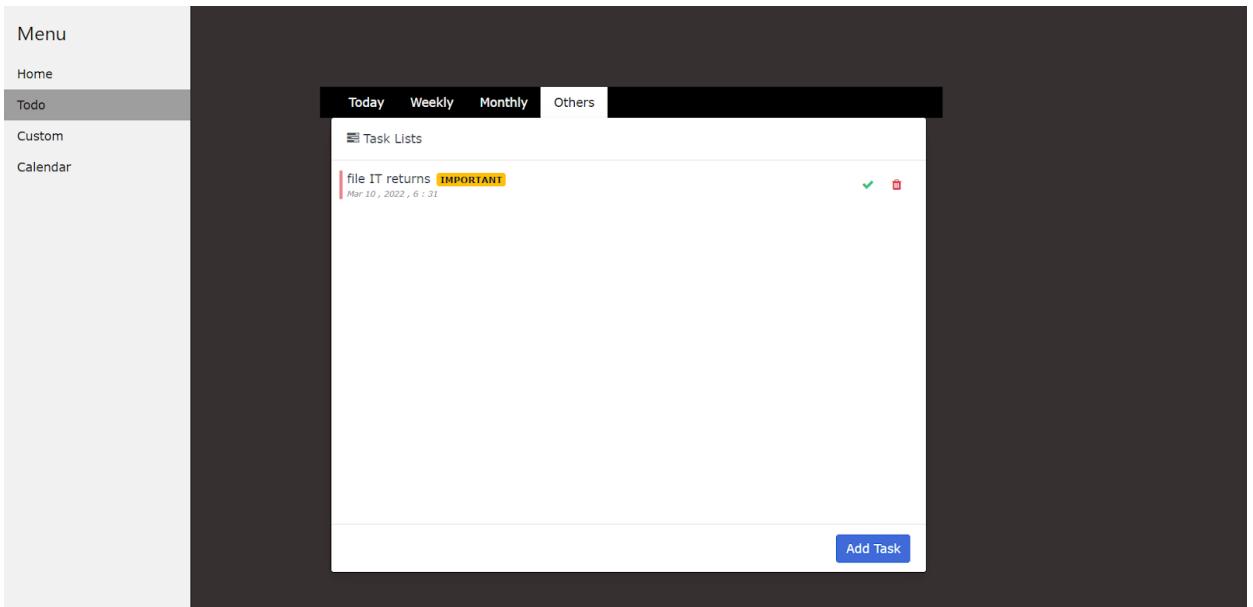
TODO WEEKLY:

A screenshot of a digital todo application interface. On the left is a vertical menu bar with options: Menu, Home, Todo (which is selected and highlighted in grey), Custom, and Calendar. The main area is a dark-themed window titled "Task Lists". At the top of this window is a navigation bar with tabs: Today, Weekly (selected), Monthly, and Others. Below the navigation is a list of tasks. The first task is "go to temple" with a due date of "Jan 18, 2022, 5 : 30". The second task is "Buy drafter" with a status indicator "IMPORTANT" and a due date of "Jan 16, 2022, 2 : 30". At the bottom right of the task list is a blue "Add Task" button.

TODO MONTHLY:

A screenshot of a digital todo application interface, similar to the weekly view but showing tasks for a month. The left menu bar includes: Menu, Home, Todo (selected), Custom, and Calendar. The main area shows a "Task Lists" window with a "Monthly" tab selected in the navigation bar. A single task is listed: "Complete game level 15" with a due date of "Feb 10, 2022, 6 : 32". A blue "Add Task" button is located at the bottom right.

TODO OTHERS:



- **Task class:**

The Task class is present in the bean package with attributes `user_id`, `task_name`, `task_id`, `deadline`, `priority`, and `completed` and contains corresponding public getter and setter methods.

It has a foreign key relationship with the User table in the database. The `completed(boolean)` attribute tells us whether the task has been completed or not, and `deadline (Date)`, `task_name(String)`, `task_id(int)`, `priority(String)` describes the Task. The `task_id` is the primary key and is labelled auto increment.

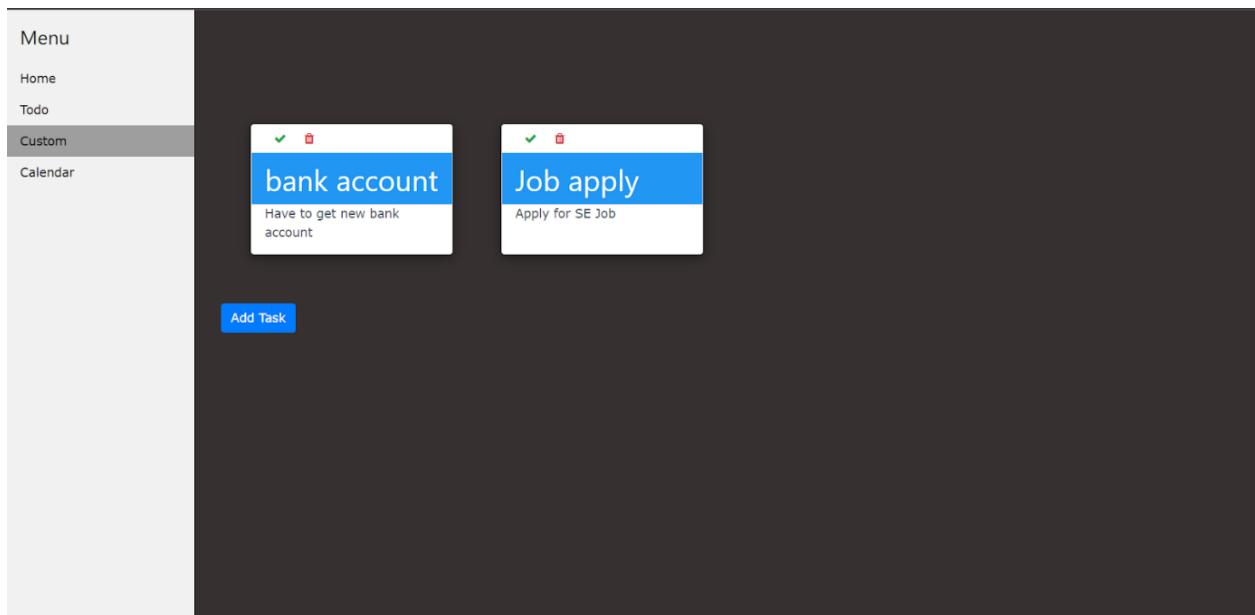
```
1 package bean;
2 import java.util.Date;
3 public class Task {
4     int task_id;
5     int user_id;
6     String task_name;
7     Date deadline;
8     String priority;
9     boolean completed;
10    public void setTask_id(int s){
11        task_id = s;
12    }
13    public void setUser_id(int s){
14        user_id = s;
15    }
16    public void setTask_name(String s){
17        task_name = s;
18    }
19    public void setDeadline(Date s){
20        deadline = s;
21    }
22    public void setPriority(String s){priority = s;}
23    public void setCompleted(boolean s){completed = s;}
24    public int getTask_id(){
25        return task_id;
26    }
27    public int getUser_id(){
28        return user_id;
29    }
30    public String getTask_name(){
31        return task_name;
32    }
33    public Date getDeadline(){
34        return deadline;
35    }
36    public String getPriority(){
37        return priority;
38    }
39    public boolean getCompleted(){
40        return completed;
41    }
42 }
```

7. MODULE 3

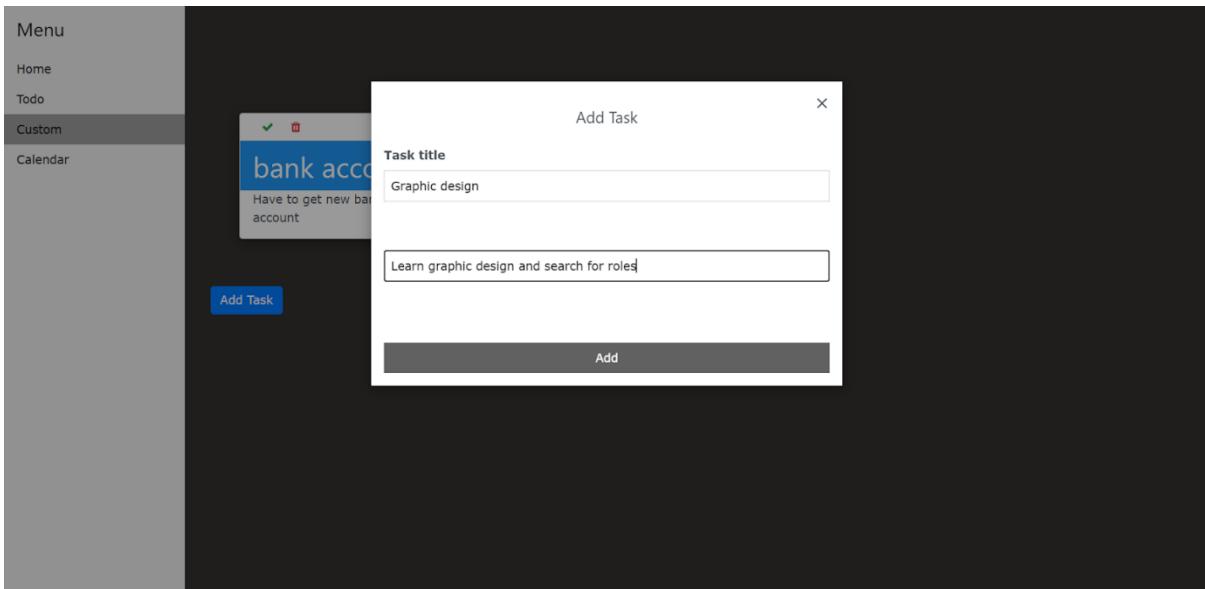
CUSTOM TASKS

This module will be accessible only for successfully signed or logged users. The user will be able to add custom tasks along with subtasks that constitute the custom task. Each subtask can be given a description and deadline. Completing a custom task completes all the subtasks in it. And similarly, deleting a custom task deletes all the subtasks in it.

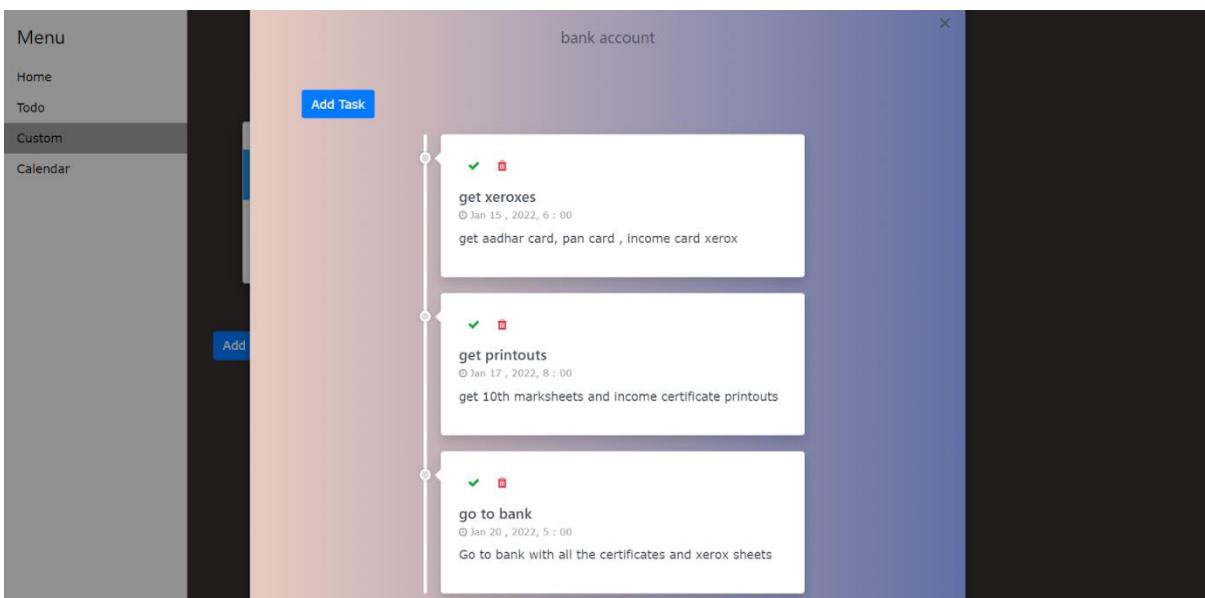
- Custom tasks list:



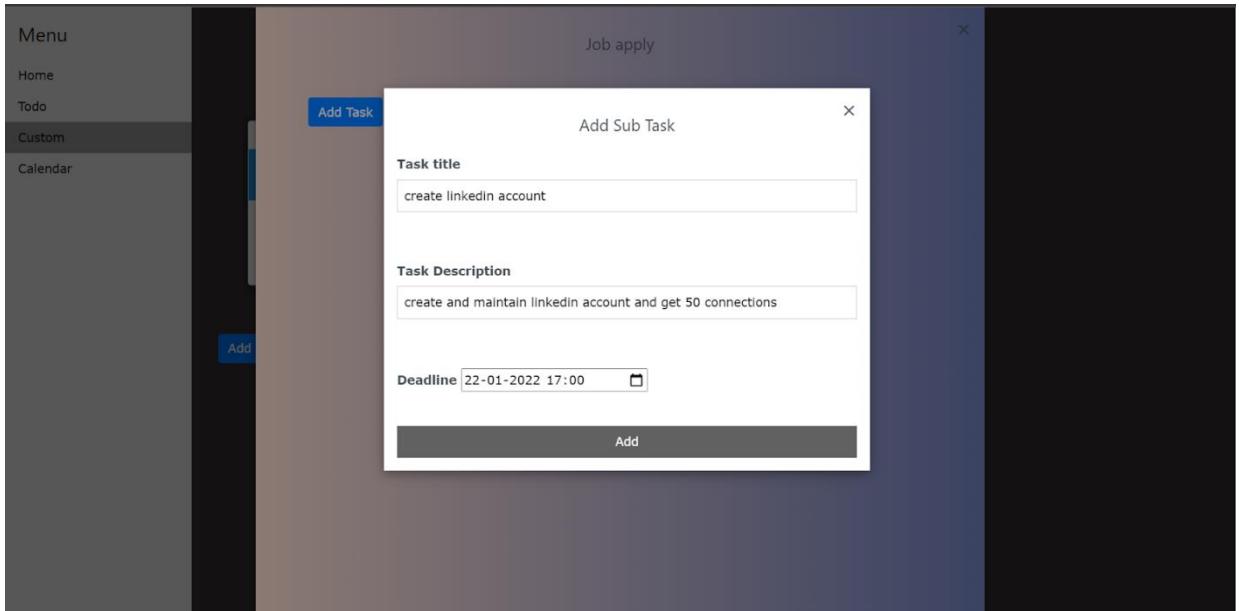
- Adding custom task:



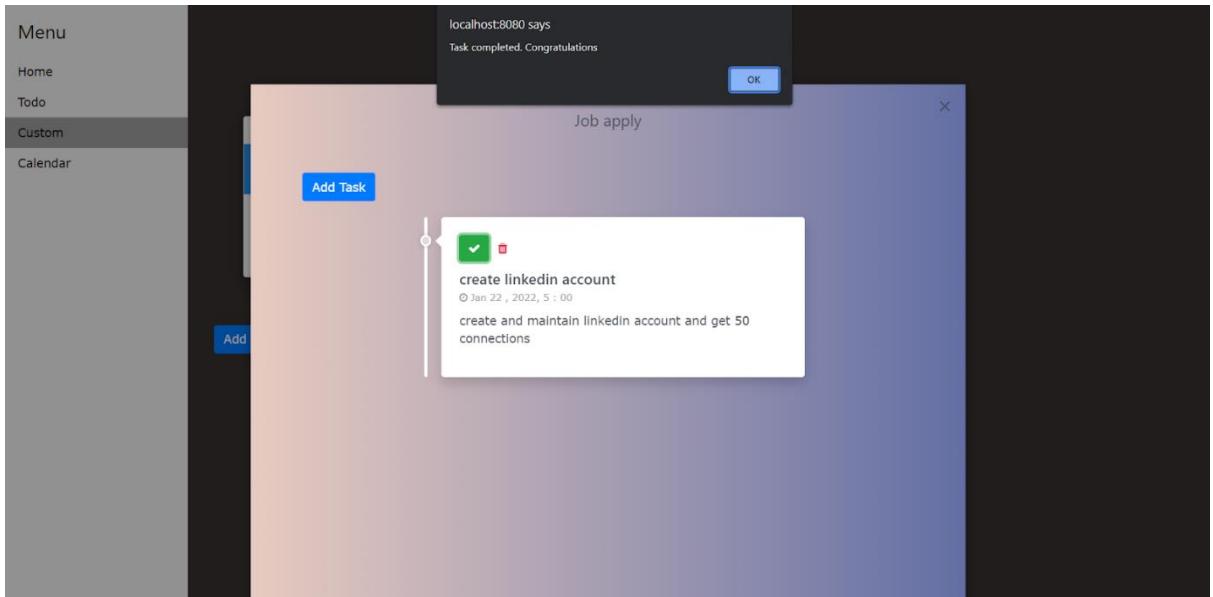
- Displaying subtasks:



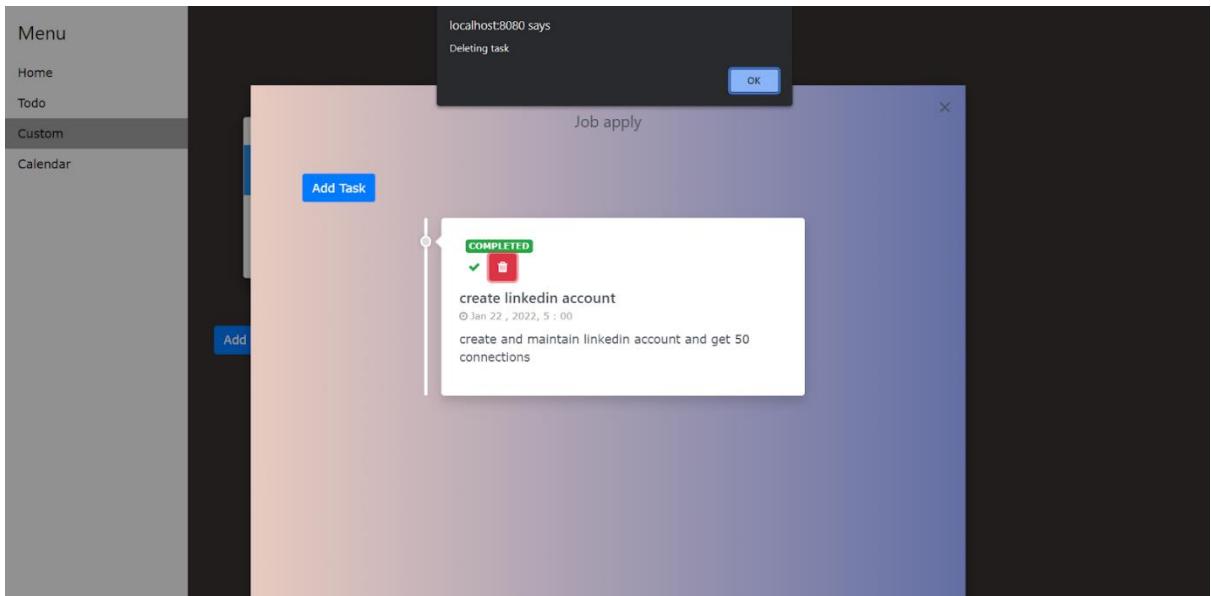
- Adding subtask:



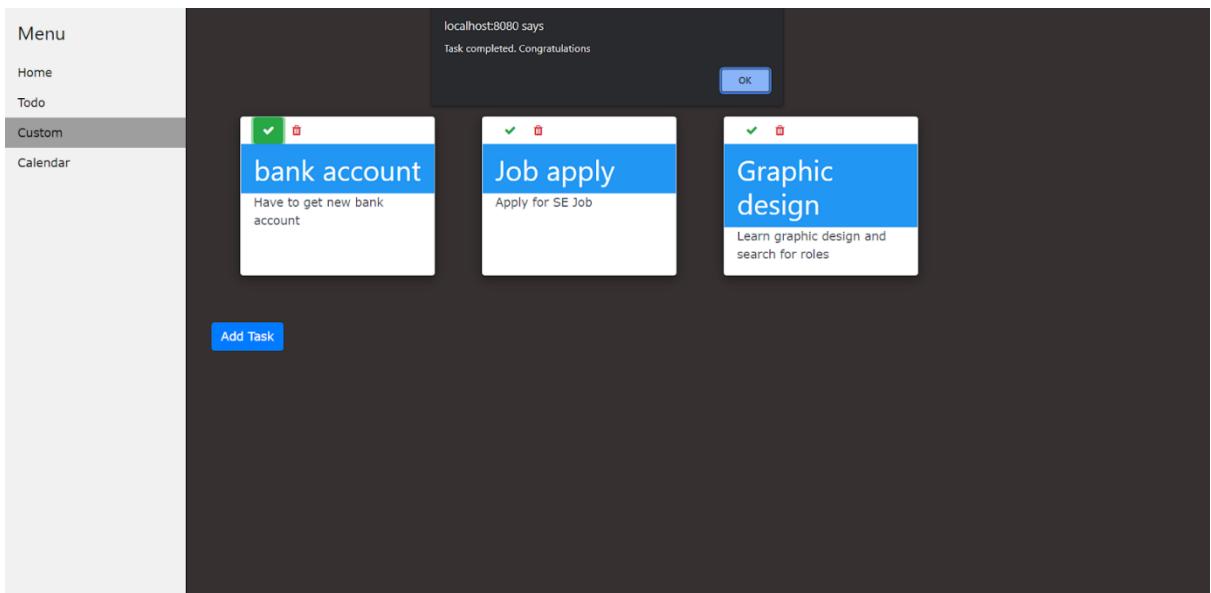
- Completing subtask:

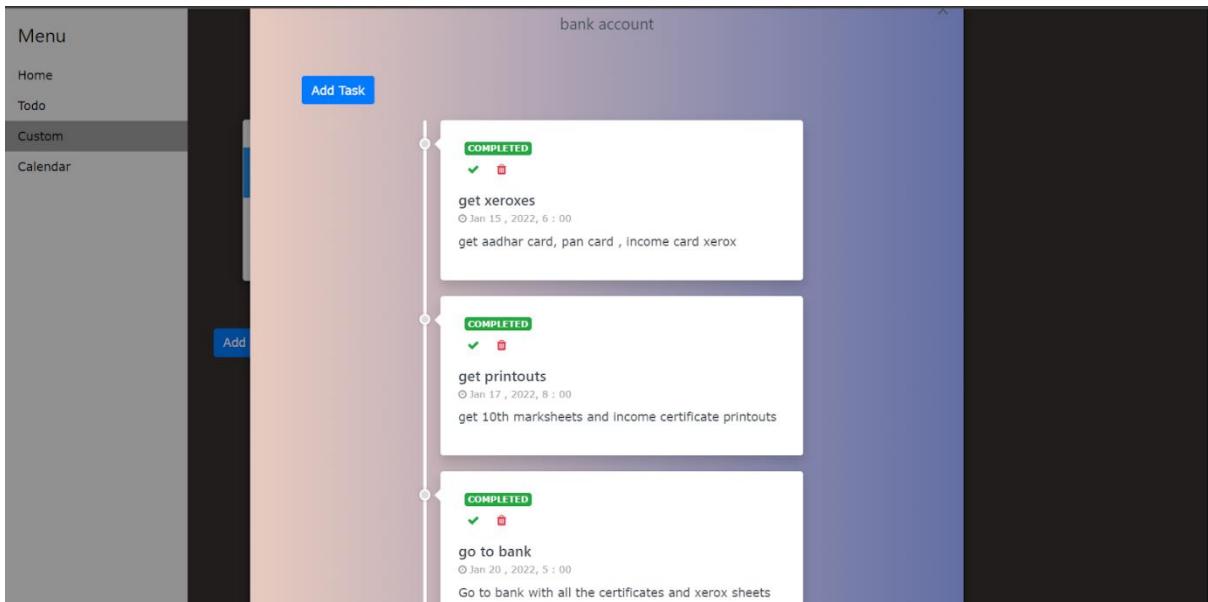


- Deleting subtask:



- Completing Custom task:





- **CustomTask class:**

CustomTask class has attributes

custom_task_id(int), user_id(int), task_name(String), desc(String), and completed(boolean) and respective public getter and setter methods.

It has a foreign key relationship with the User table in the database and user_id being the referencing attribute. custom_task_id is the primary key and is labelled auto increment.

```
1 package bean;
2 public class CustomTask {
3     private int custom_task_id;
4     private int user_id;
5     private String task_name;
6     private String desc;
7     private boolean completed;
8     public int getCustom_task_id() {
9         return custom_task_id;
10    }
11    public int getUser_id() {
12        return user_id;
13    }
14    public String getTask_name() {
15        return task_name;
16    }
17    public boolean getCompleted() {
18        return completed;
19    }
20    public String getDesc() {
21        return desc;
22    }
23    public void setCustom_task_id(int i) {
24        custom_task_id=i;
25    }
26    public void setUser_id(int i) {
27        user_id=i;;
28    }
29    public void setTask_name(String i) {
30        task_name = i;
31    }
32    public void setDesc(String i) {
33        desc = i;
34    }
35    public void setCompleted(boolean i) {
36        completed = i;
37    }
38 }
39 }
```

- SubTask class:

SubTask class has attributes custom_task_id(int), sub_task_id(int), task_name(String), desc(String), deadline(Date), and completed(boolean) and respective public getter and setter methods.

It has a foreign key relationship with the CustomTask table in the database and custom_task_id being the referencing attribute. sub_task_id is the primary key and is labelled auto increment.

```

1 package bean;
2 import java.util.*;
3 public class SubTask {
4     private int sub_task_id;
5     private int custom_task_id;
6     private Date deadline;
7     private boolean completed;
8     private String task_name;
9     private String desc;
10    public int getCustom_task_id() {
11        return custom_task_id;
12    }
13    public int getSub_task_id() {
14        return sub_task_id;
15    }
16    public String getTask_name() {
17        return task_name;
18    }
19    public String getDesc() {
20        return desc;
21    }
22    public Date getDeadline() {
23        return deadline;
24    }
25    public boolean getCompleted() {
26        return completed;
27    }
28    public void setCustom_task_id(int i) {
29        custom_task_id=i;
30    }
31    public void setSub_task_id(int i) {
32        sub_task_id=i;
33    }
34    public void setTask_name(String i) {task_name = i;}
35    public void setDesc(String i) {desc = i;}
36    public void setDeadline(Date d) {deadline = d;}
37    public void setCompleted(boolean i) {completed = i;}
38 }
39

```

8. MODULE 4

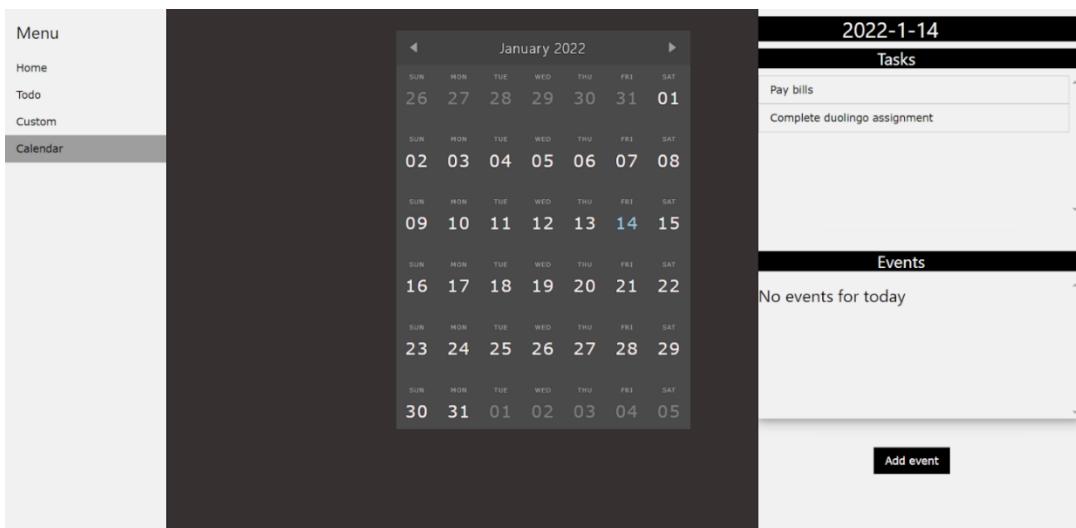
CALENDAR

Even this module is accessible only on successful sign up or login. This module provides the calendar feature where on selecting a date will display all the tasks that had/has a deadline on that day.

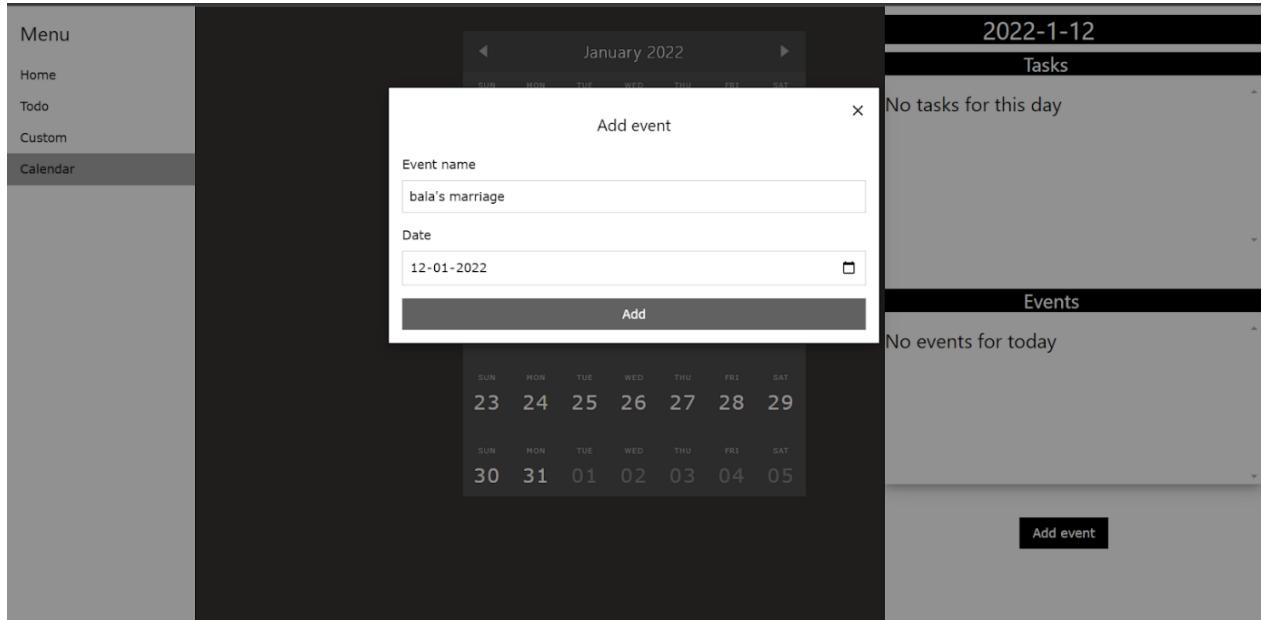
It also provides another feature called Events. Users will be able to add Events for a particular day and these Events will be displayed alongside the Tasks on selecting a day in the calendar. The current date will be displayed in light blue colour while the date selected will be displayed in dark blue colour.

- Calendar display:

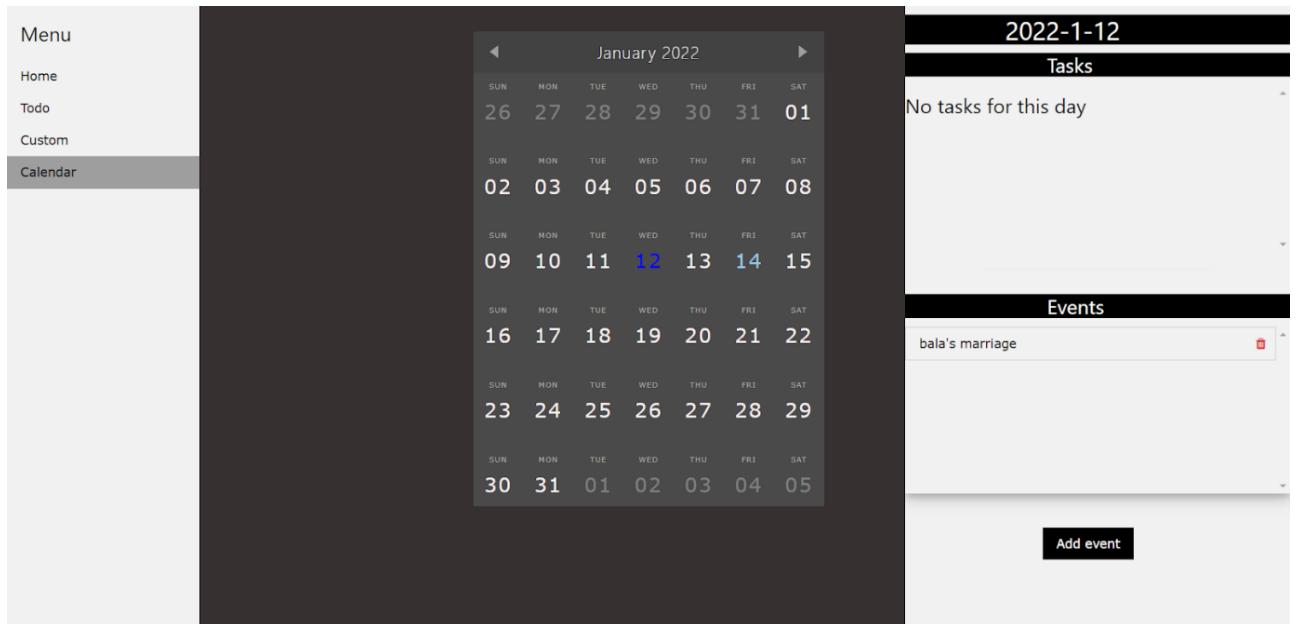
Technologies used: HTML, CSS, JavaScript, jQuery, AJAX, JSP, Servlet, Java, MySQL



- Add events:



- Display events:



- CalenderEvent class:
CalenderEvent class has attributes `user_id(int)`,

event_id(int), event_name(String), and event_date(Date), and respective public getter and setter methods.

It has a foreign key relationship with the User table in the database and user_id being the referencing attribute. event_id is the primary key and is labelled auto increment

```
1 package bean;
2 import java.util.*;
3
4 public class CalenderEvent {
5     private int user_id;
6     private int event_id;
7     private String event_name;
8     private Date event_date;
9     public int getUser_id() {
10         return user_id;
11     }
12     public int getEvent_id() {
13         return event_id;
14     }
15     public String getEvent_name() {
16         return event_name;
17     }
18     public Date getEvent_date() {
19         return event_date;
20     }
21     public void setEvent_id(int i) {
22         event_id = i;
23     }
24     public void setUser_id(int i) {
25         user_id = i;
26     }
27     public void setEvent_name(String s) {
28         event_name = s;
29     }
30     public void setEvent_date(Date d) {
31         event_date = d;
32     }
33
34 }
```

9. MODULE 5

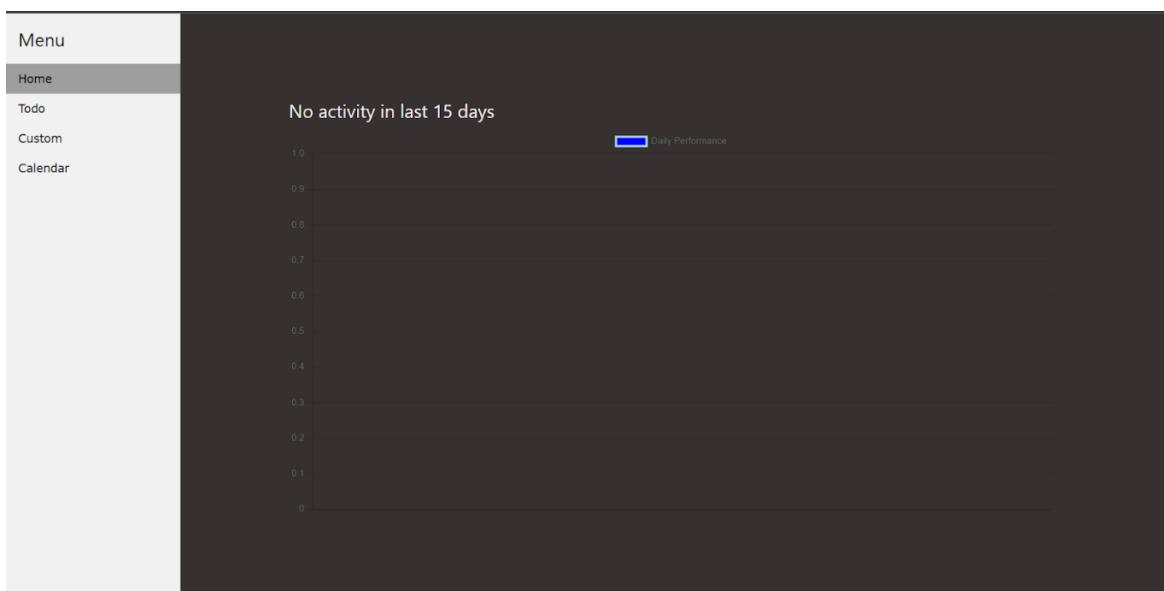
USER STATS

This module is the first page that the user views when he/she successfully signs up or logs in. This page displays the summary of the user's activity in the webapp for the last 15 days. The summary is represented in graphs where the x-axis shows the dates when the user was active, and the y-axis has the percentage of tasks completed for the respective date on x-axis.

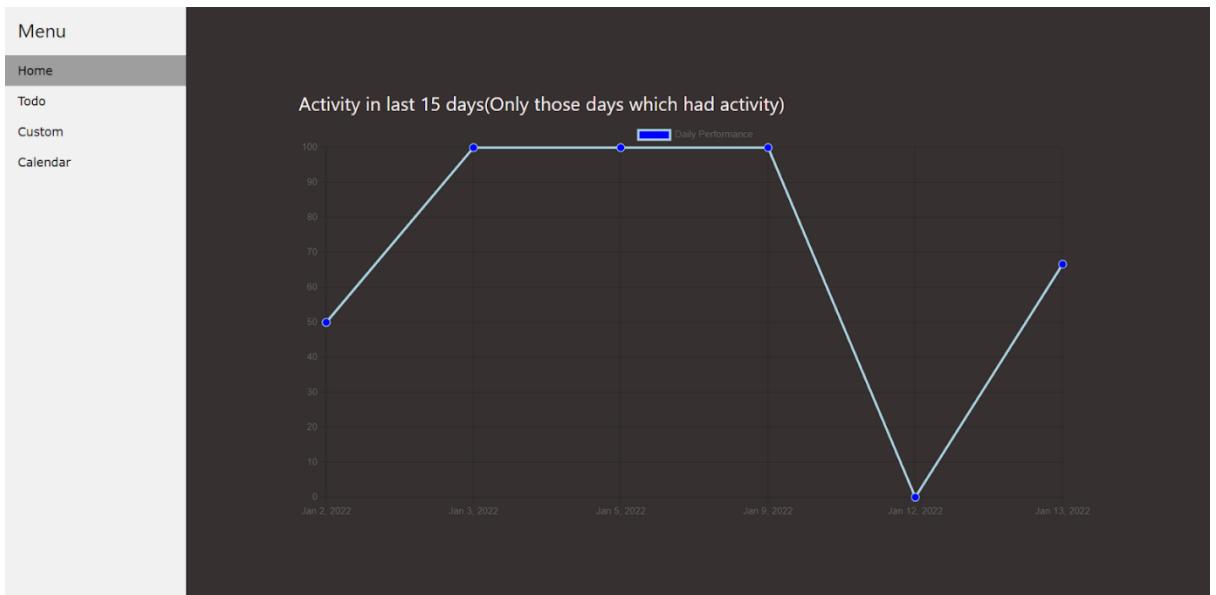
The percentage of completed tasks is calculated by:

$$\frac{100 * \text{No of tasks completed on a particular day}}{\text{No of tasks scheduled for that particular day}}$$

- User stats when no activity by the user for the last 15 days:



- User stats when has shown some activity on 6 dates:

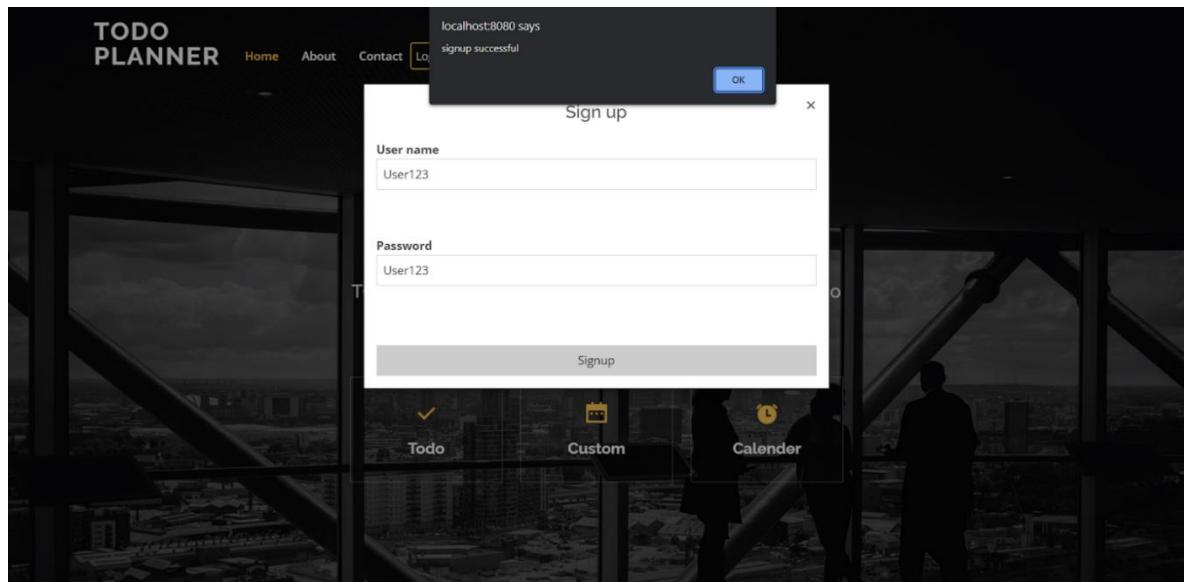


10. RESULTS

FLOW FROM USER SIGNUP TO NAVIGATING THROUGH THE WEBAPP

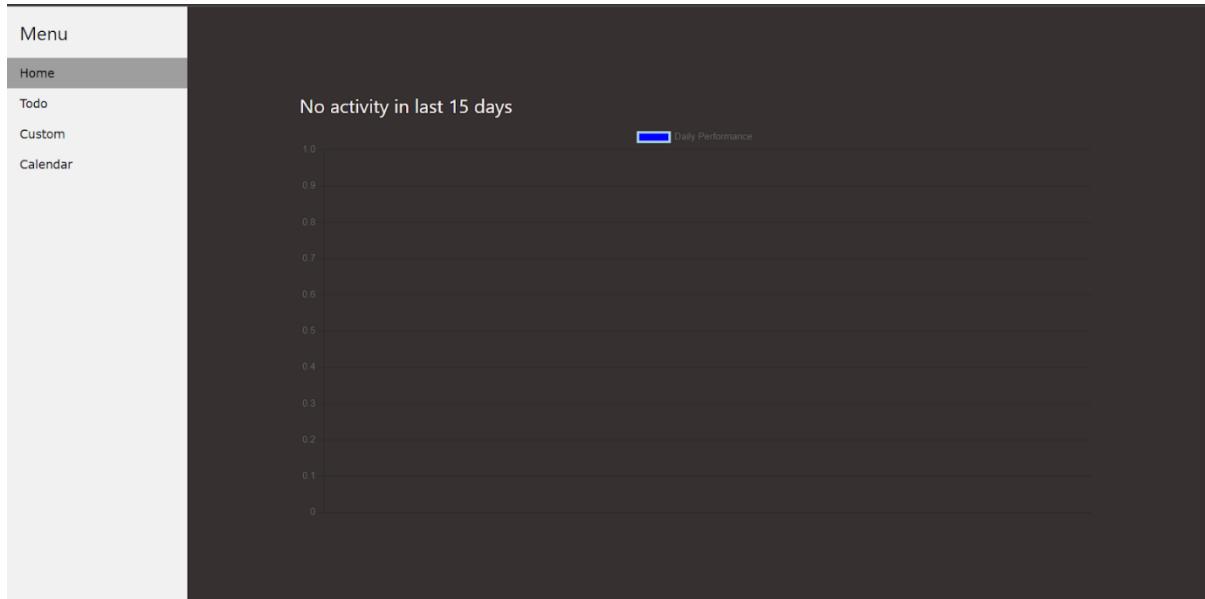
1. Signup

Users must enter a user name and password to sign up. The user name must not be already present for the signup to be successful



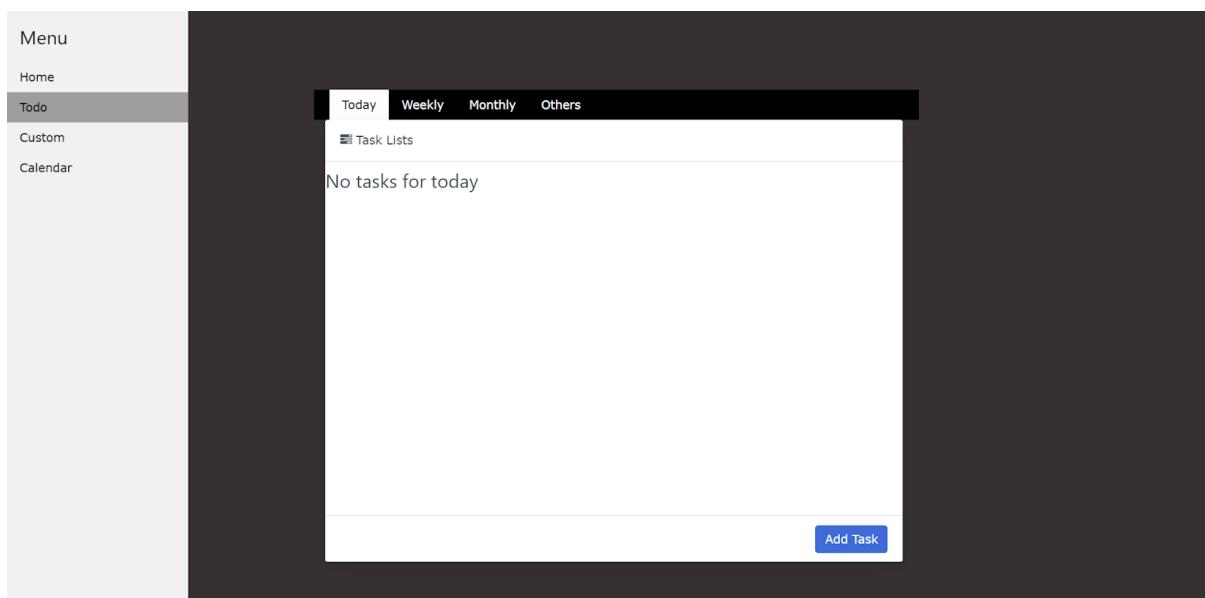
2. Home page(User stats)

This is the first page the user sees after logging in. Since the user has logged in for the first time, the user stats will be empty.



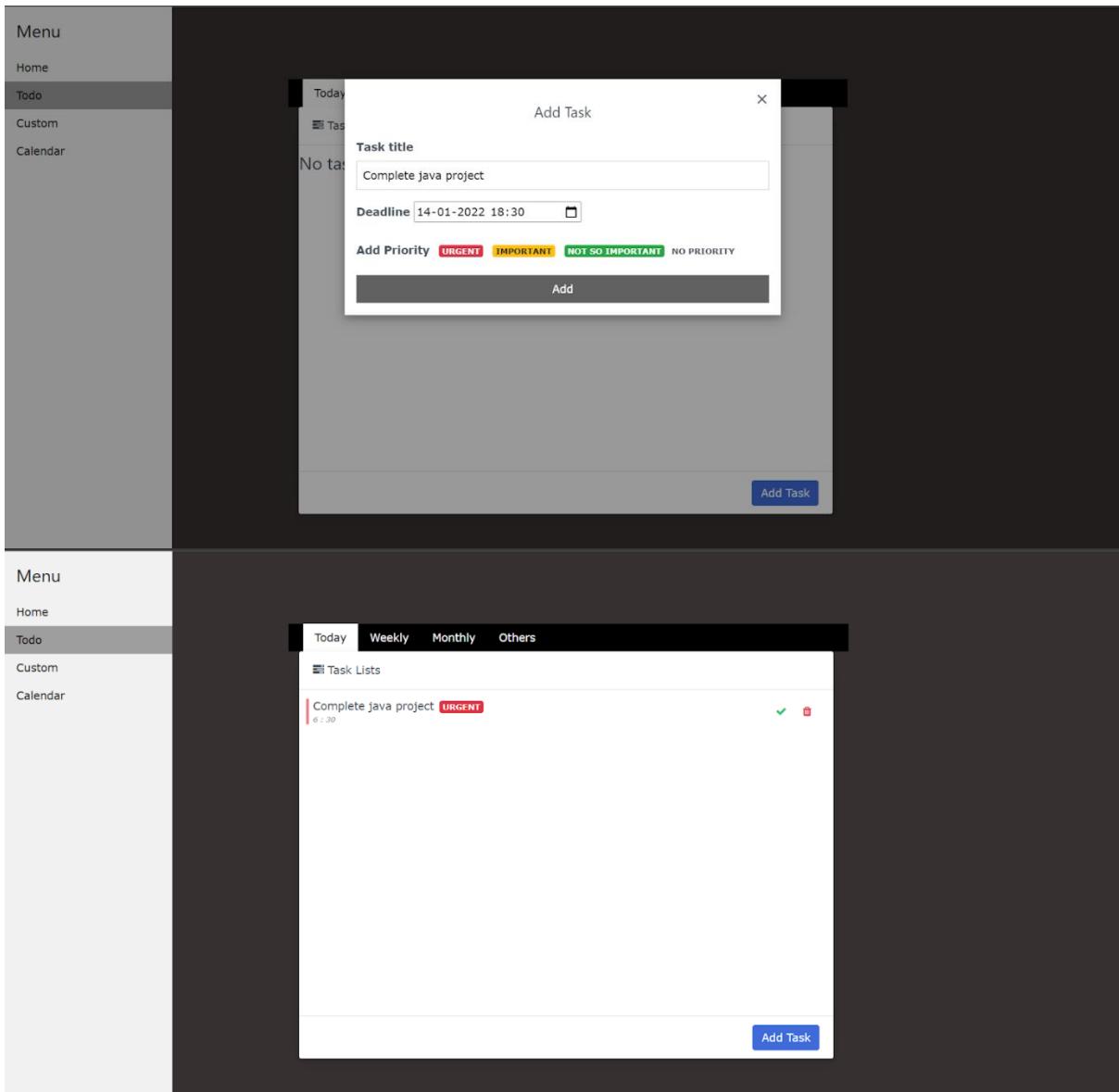
3. Todo lists

The tasks sorted into Today, Weekly, Monthly and Others are listed on this page. The user can assign priority and deadline while adding the tasks. On clicking the complete button(a tick) the task is labelled as completed. On clicking the delete button , the task is deleted from the database.



The image displays three vertically stacked screenshots of a dark-themed todo application interface. Each screenshot shows a sidebar on the left with a "Menu" section containing "Home", "Todo" (which is highlighted with a grey background), "Custom", and "Calendar". The main content area features a navigation bar at the top with tabs: "Today", "Weekly" (which is selected and highlighted with a white background), "Monthly", and "Others". Below the tabs is a section titled "Task Lists" with a message: "No tasks for the next 7 days". At the bottom right of this section is a blue "Add Task" button. The middle and bottom screenshots show the same layout but with different timeframes: the middle one shows "No tasks for the next 30 days" and the bottom one shows "No tasks".

Adding a task:



After adding few more tasks:

Menu

Home

Todo

Custom

Calendar

Today Weekly Monthly Others

Task Lists

| | | | |
|-----------------------|------------------|---|---|
| Complete java project | URGENT | ✓ | ✗ |
| Complete CD model | NOT SO IMPORTANT | ✓ | ✗ |
| Pay current bill | | ✓ | ✗ |

Add Task

Menu

Home

Todo

Custom

Calendar

Today Weekly Monthly Others

Task Lists

| | | | |
|--------------------|-----------|---|---|
| Present CN seminar | IMPORTANT | ✓ | ✗ |
|--------------------|-----------|---|---|

Add Task

Menu

Home

Todo

Custom

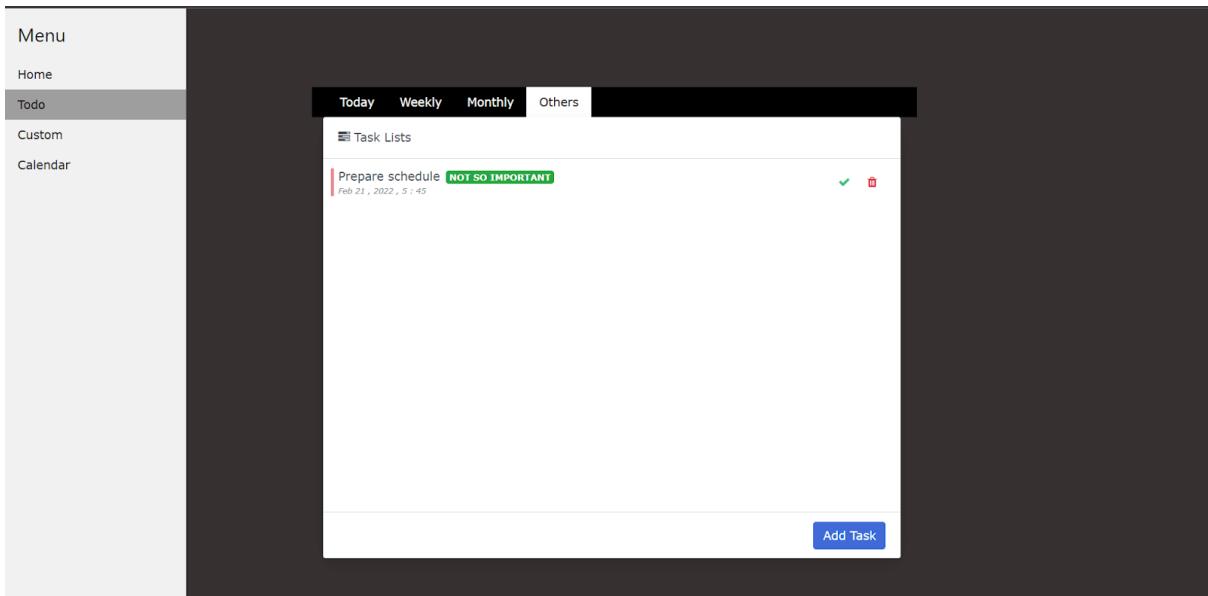
Calendar

Today Weekly **Monthly** Others

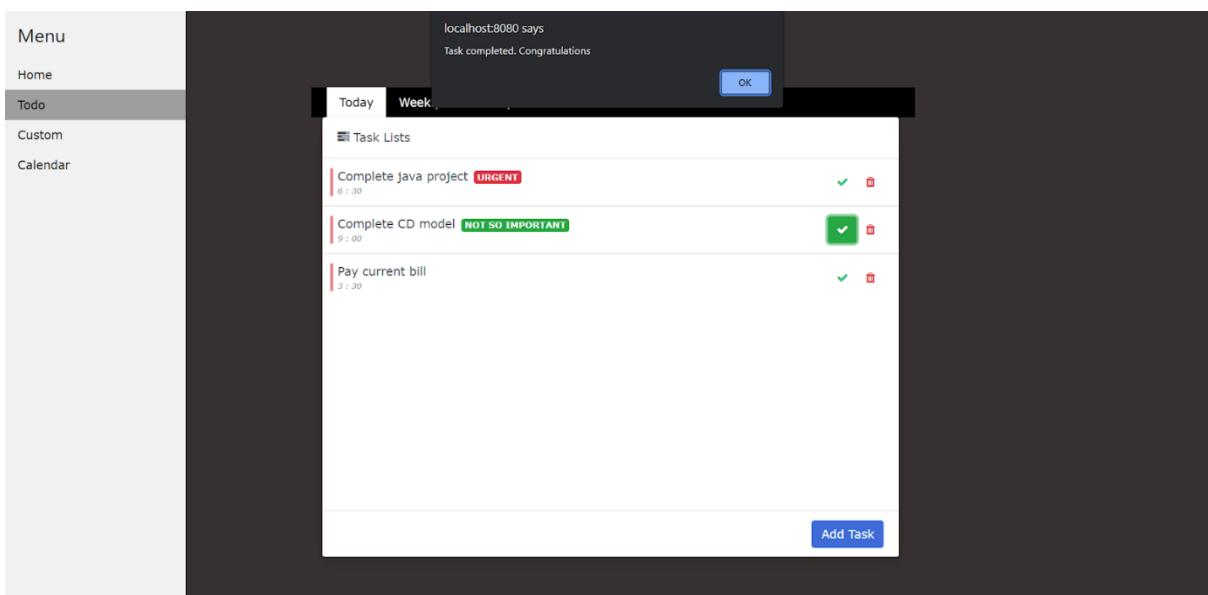
Task Lists

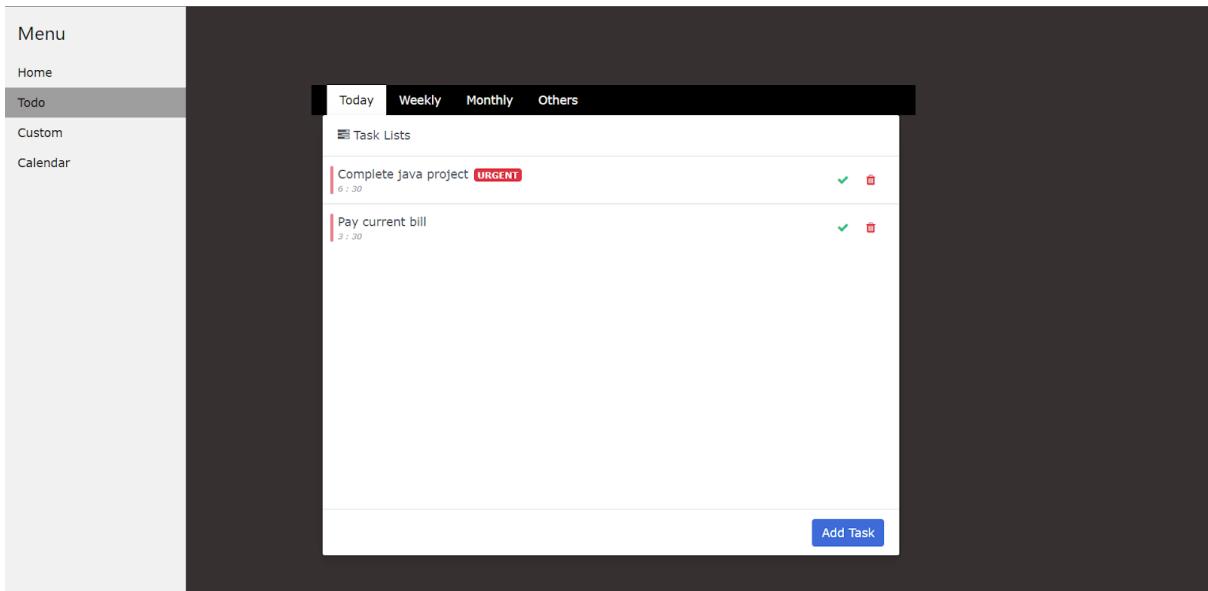
| | | | |
|----------------------------|-----------|---|---|
| Complete Java assignment | IMPORTANT | ✓ | ✗ |
| Complete leancode syllabus | | ✓ | ✗ |

Add Task

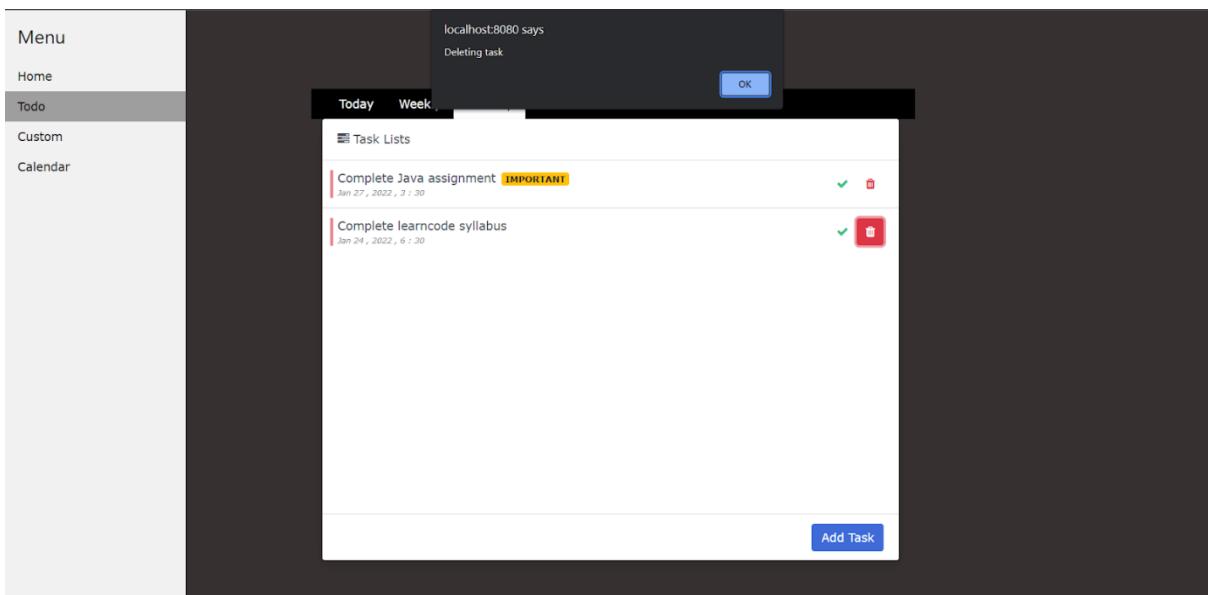


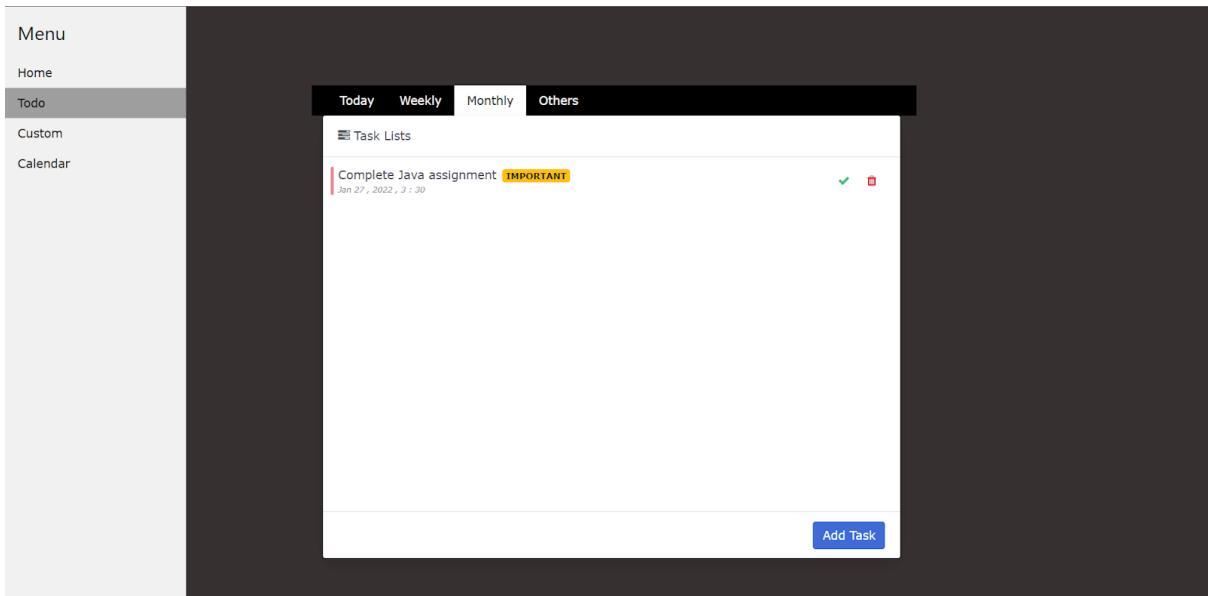
Completing a task:





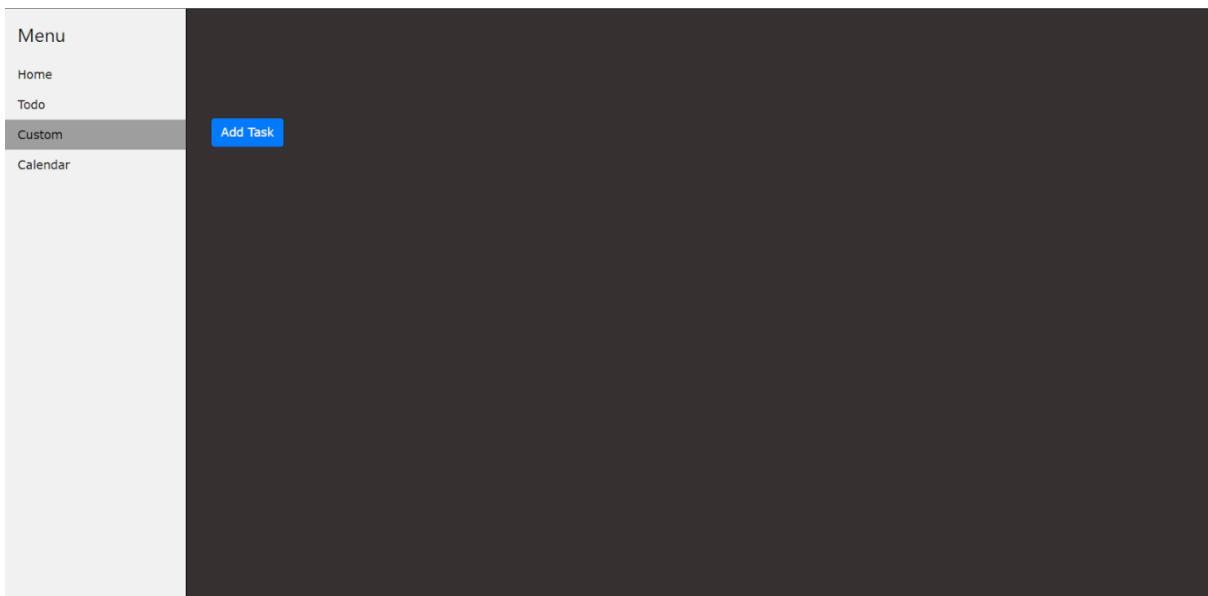
Deleting a task:



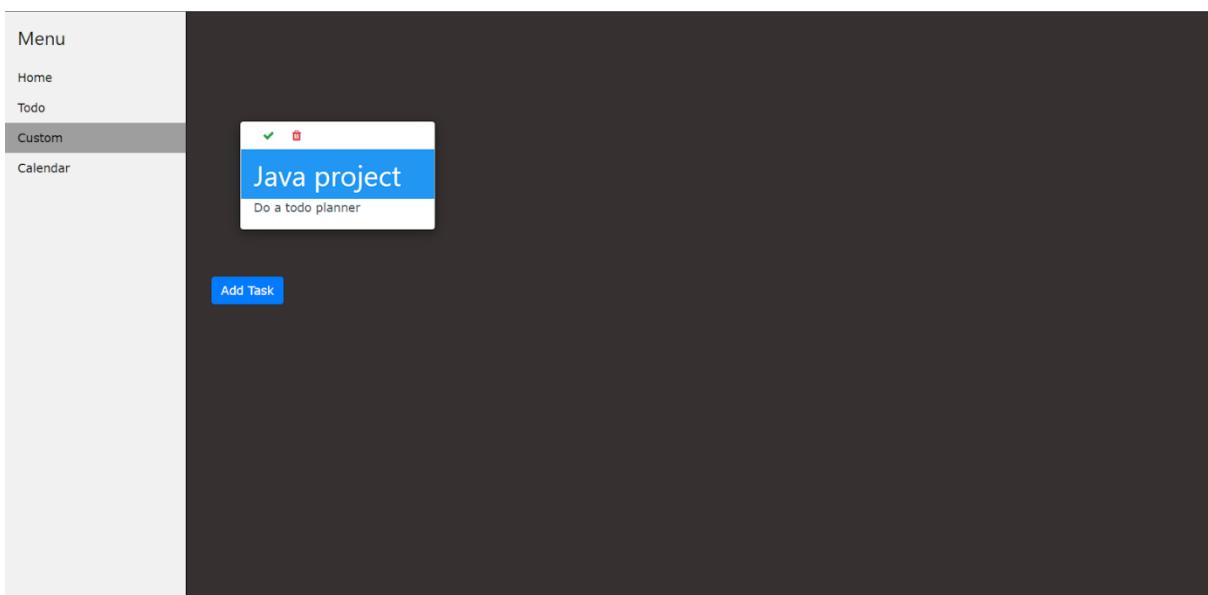
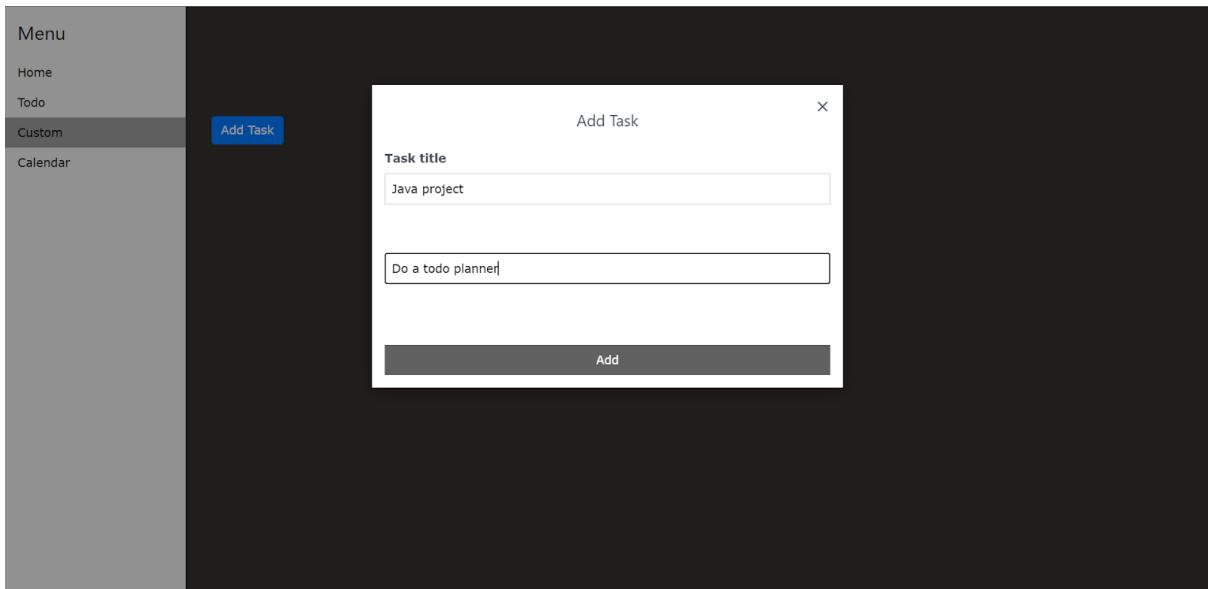


4. Custom tasks

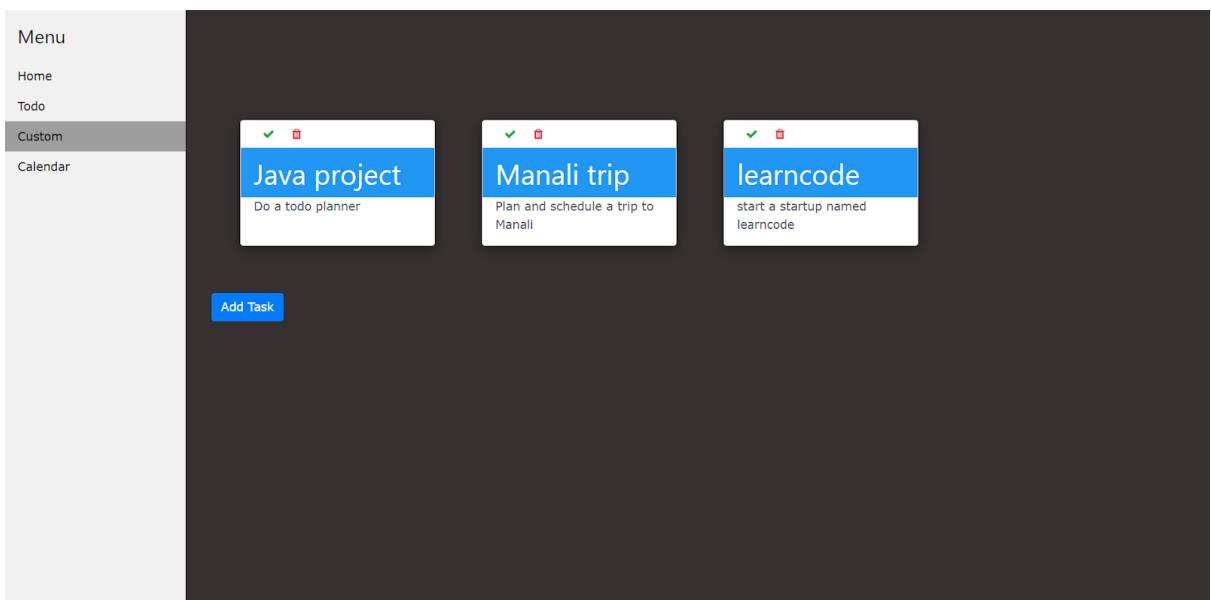
The user will be able to add custom tasks along with subtasks for a particular custom task. The subtask can be individually completed or deleted by clicking the complete and delete button respectively. Completing the custom task completes all the subtasks under it and same with deleting a custom task.



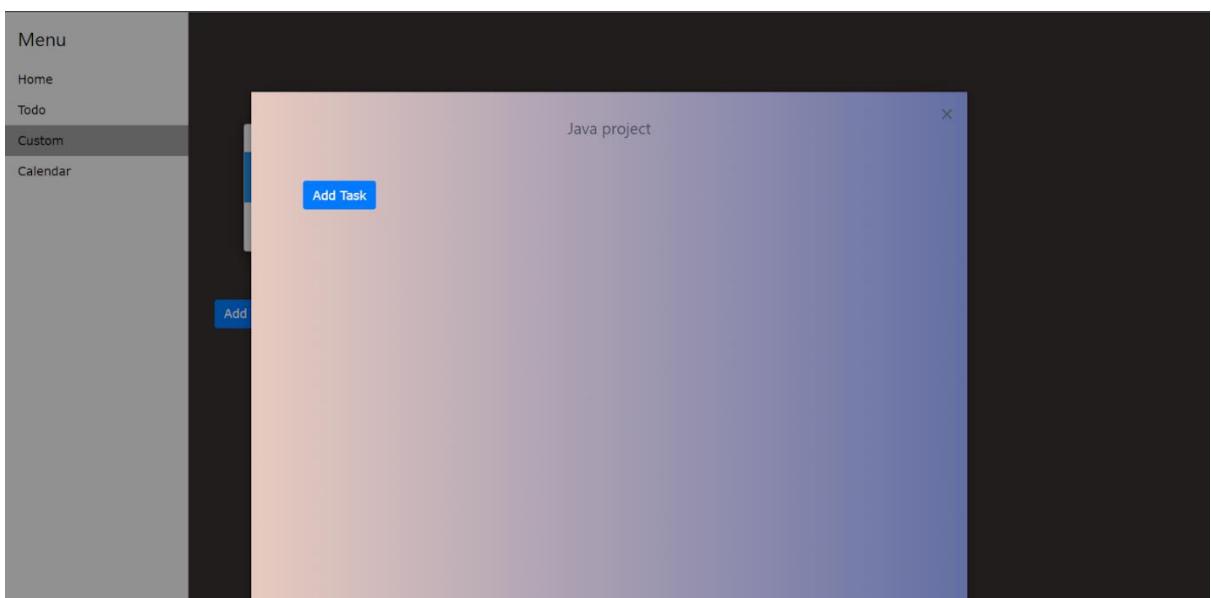
Adding custom task:

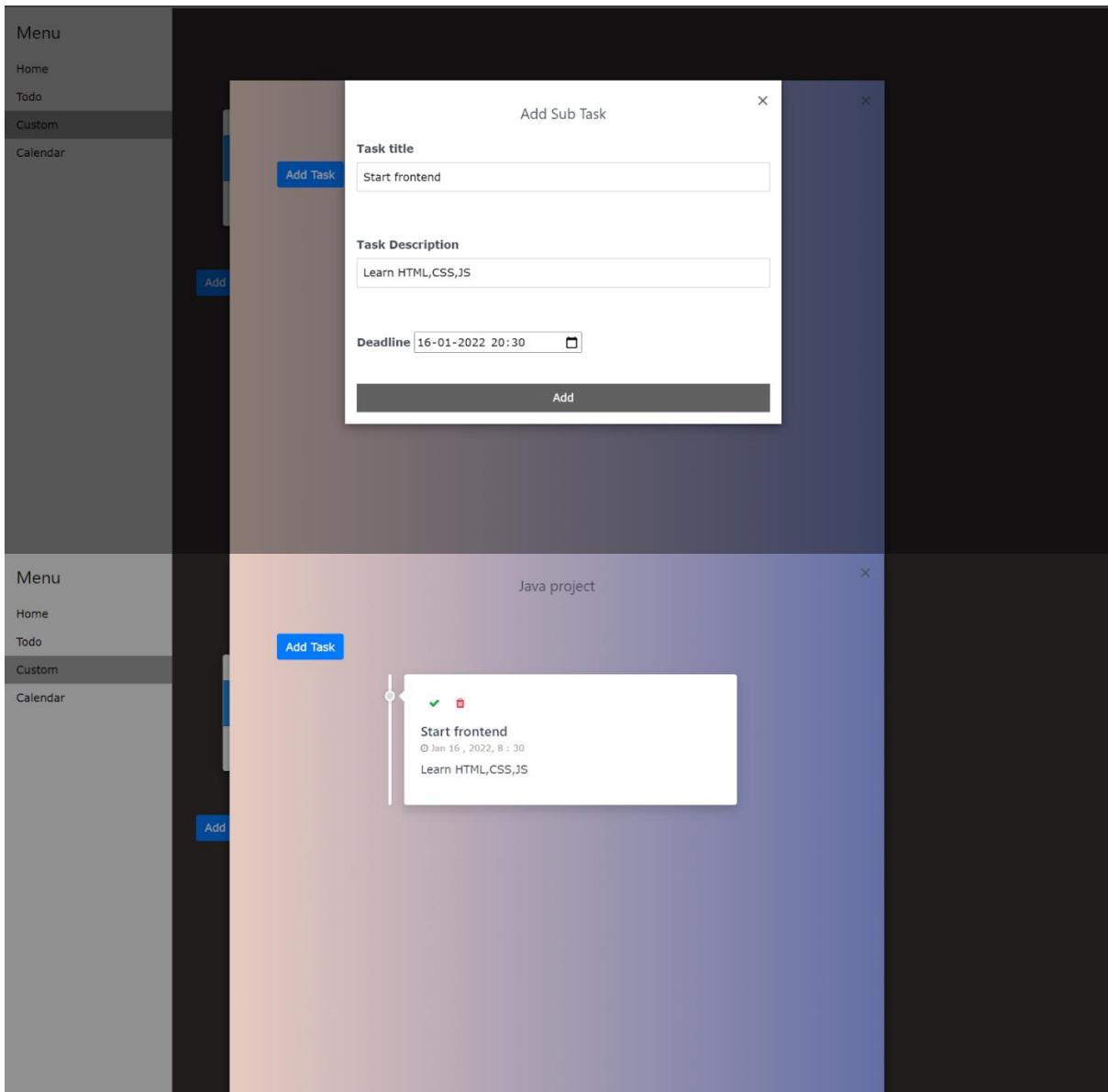


Adding few more custom tasks:

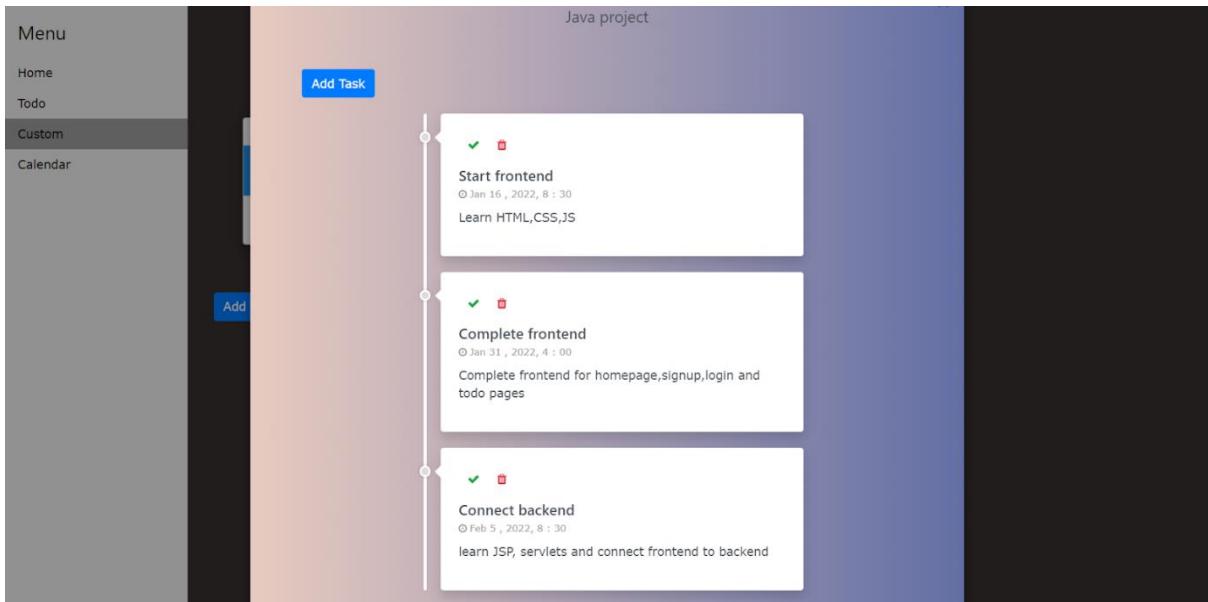


Adding subtasks:

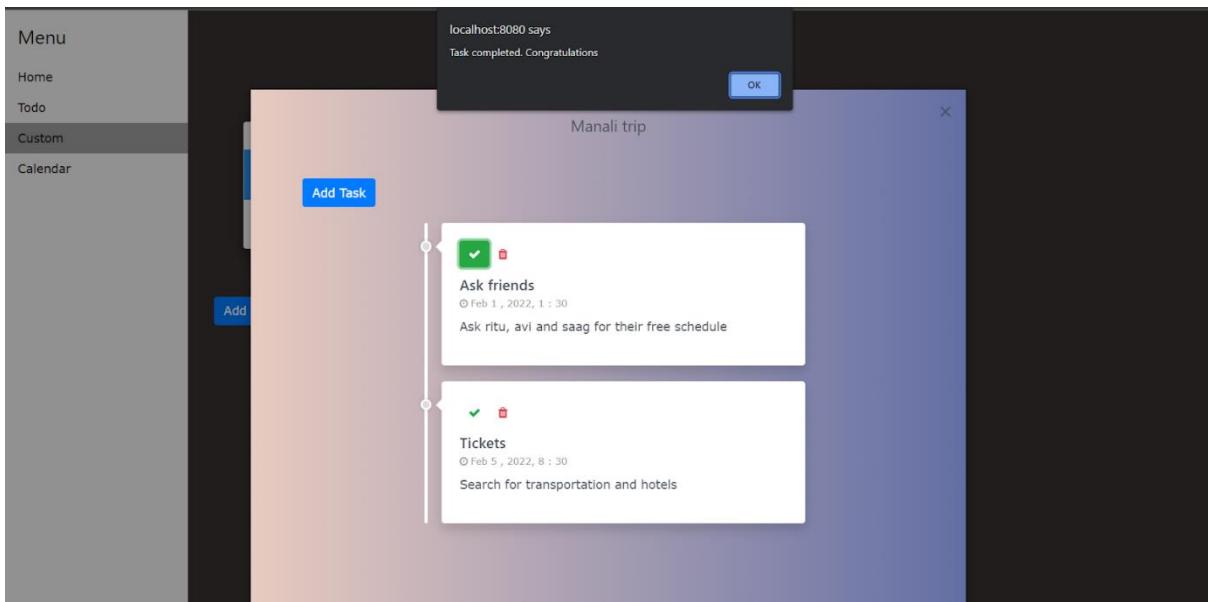


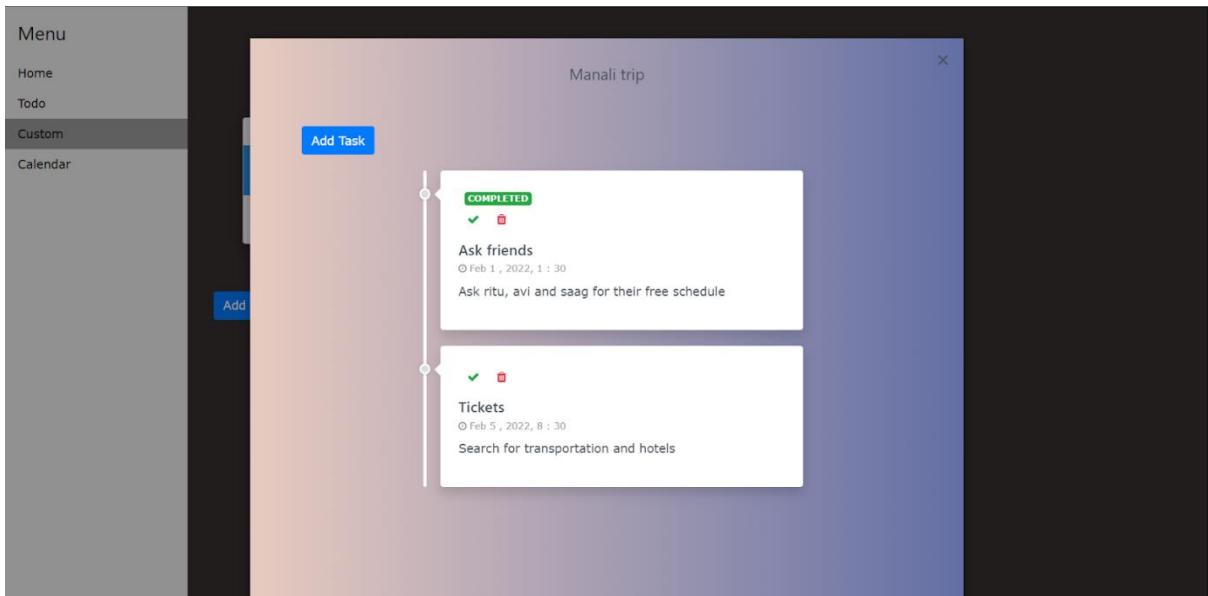


Adding few more sub tasks:

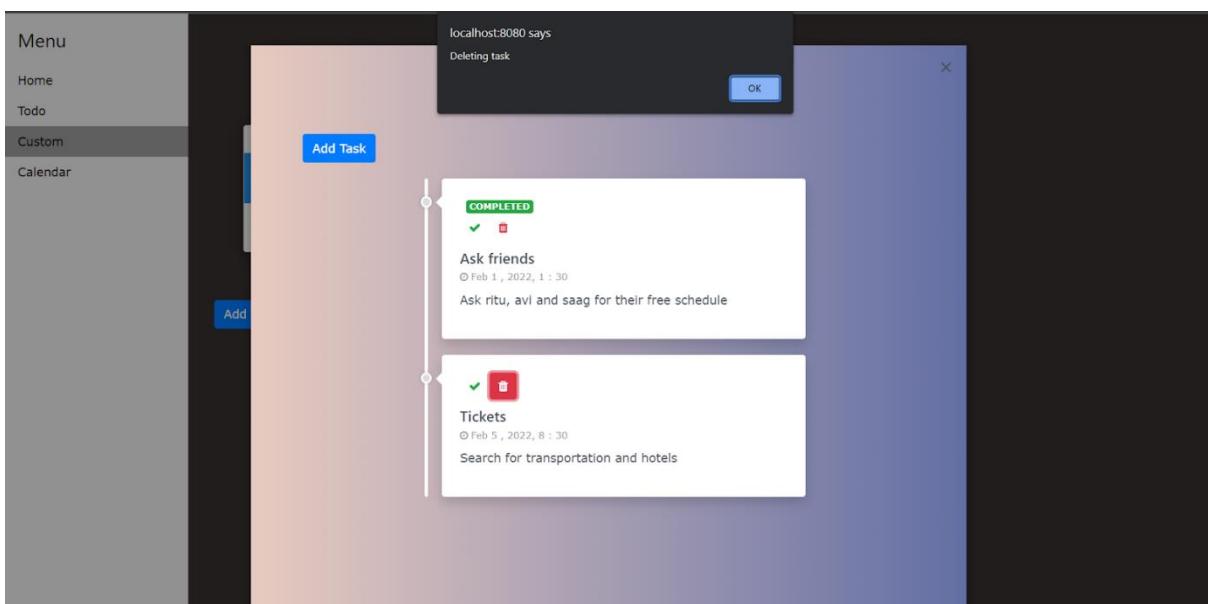


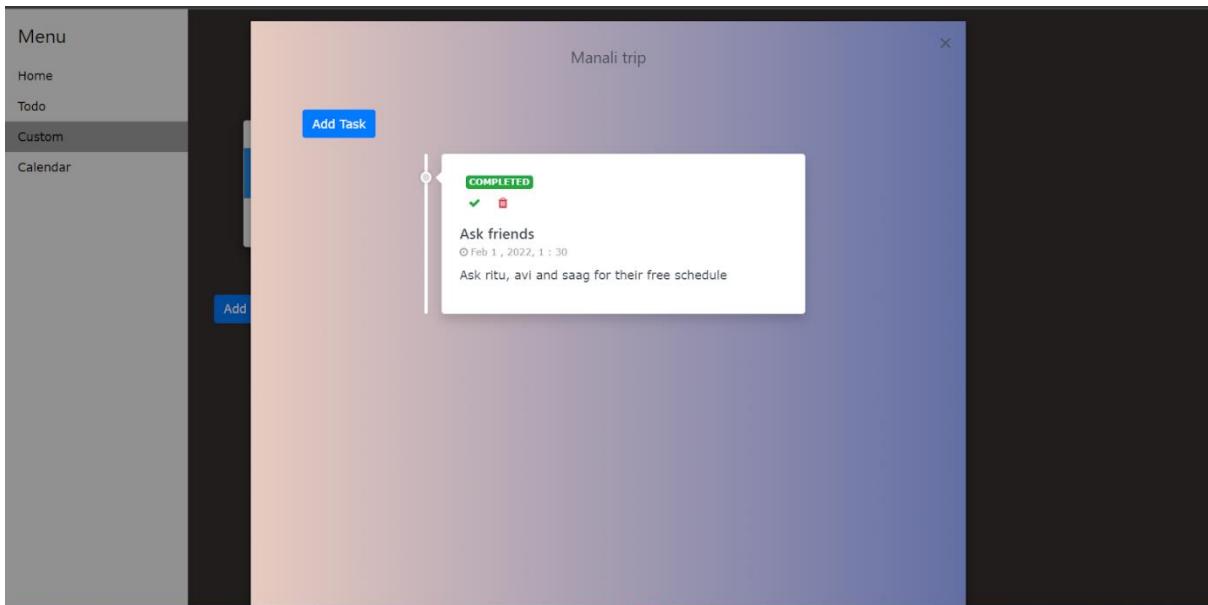
Completing a subtask:



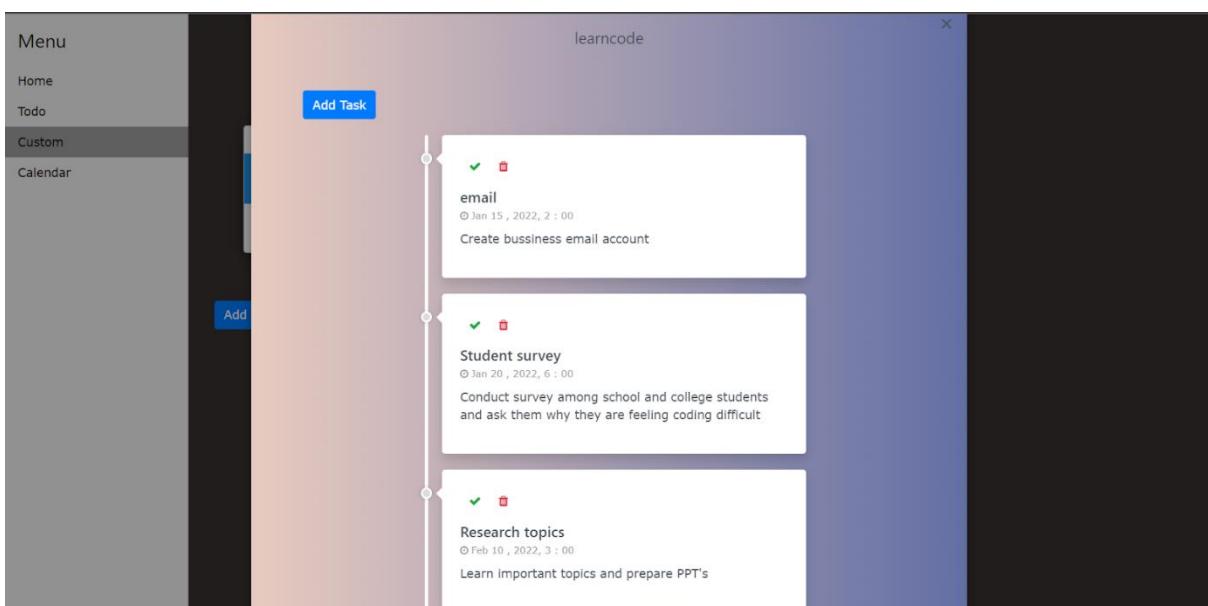


Deleting a subtask:





Completing a custom task:



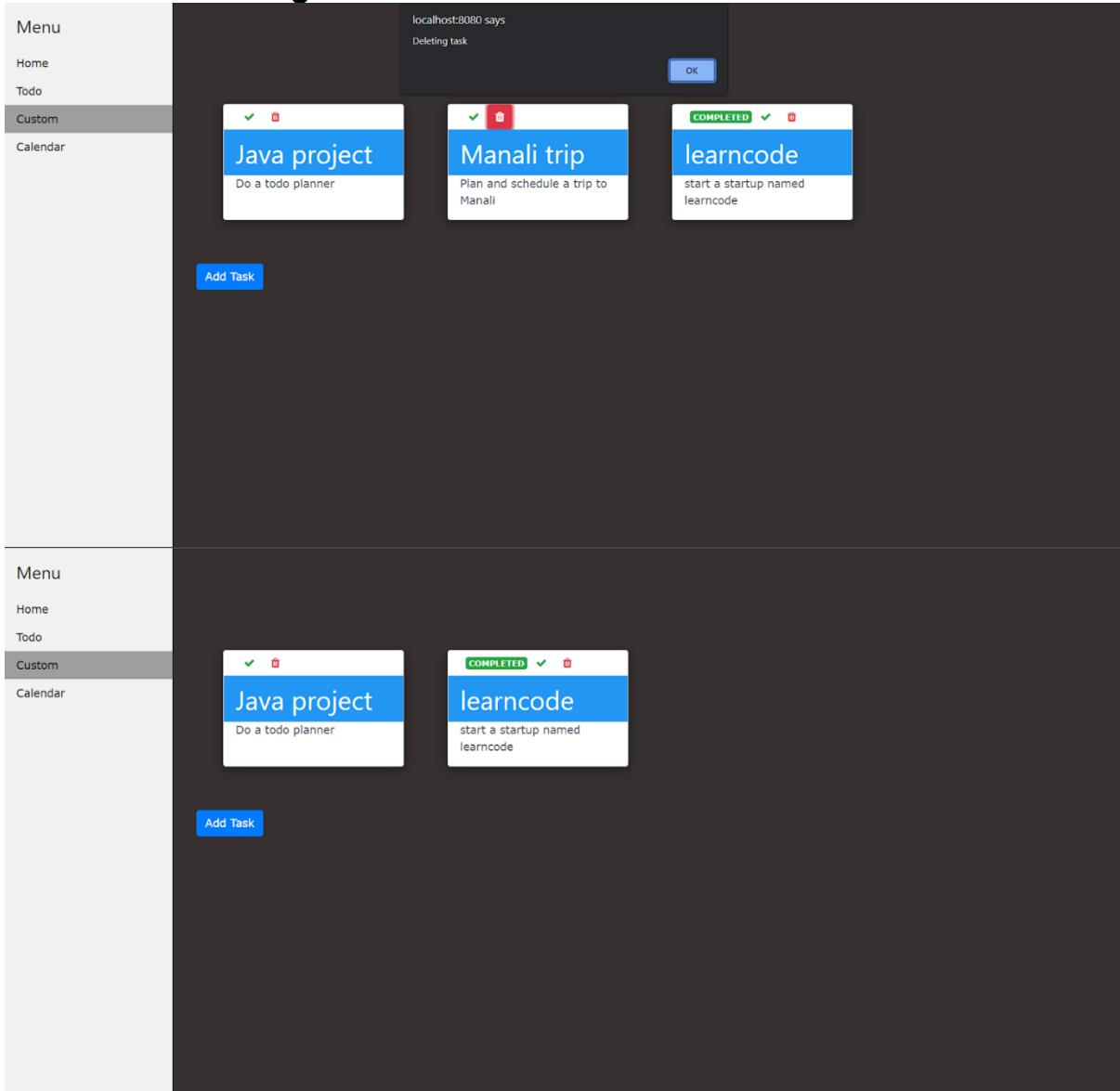
The image consists of three vertically stacked screenshots of a todo application interface, likely from a mobile device.

Screenshot 1: The application has a dark theme. On the left is a vertical menu bar with options: Home, Todo, Custom (which is selected and highlighted in grey), and Calendar. In the center, there are three cards for tasks: "Java project" (Do a todo planner), "Manali trip" (Plan and schedule a trip to Manali), and "learncode" (start a startup named learncode). Each card has a green checkmark icon and a red delete icon. A small blue "OK" button is visible in the top right corner of the screen. At the bottom left is a blue "Add Task" button.

Screenshot 2: Similar to the first screenshot, but the "learncode" task card now has a green "COMPLETED" label at the top instead of a checkmark. The other two tasks remain unchanged.

Screenshot 3: This screenshot shows a detailed view of the completed "learncode" task. The background is divided into three horizontal sections with different colors: pink on the left, purple in the middle, and dark blue on the right. The task card is centered and shows the following details: "email" (COMPLETED, Jan 15, 2022, 2:00, Create business email account), "Student survey" (COMPLETED, Jan 20, 2022, 6:00, Conduct survey among school and college students and ask them why they are finding coding difficult), and "Research topics" (COMPLETED, Feb 10, 2022, 3:00, Research topics). Each card has a green checkmark icon and a red delete icon. At the top right of this screen is a close button (X).

Deleting a custom task:



5. Calendar

This feature allows the user to view the tasks scheduled for a particular day and also add events for that day.

Calendar display for today:

Menu

- Home
- Todo
- Custom
- Calendar

◀ January 2022 ▶

| SUN | MON | TUE | WED | THU | FRI | SAT |
|-----|-----|-----|-----|-----|-----|-----|
| 26 | 27 | 28 | 29 | 30 | 31 | 01 |
| 02 | 03 | 04 | 05 | 06 | 07 | 08 |
| 09 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 | 01 | 02 | 03 | 04 | 05 |

2022-1-14

Tasks

- Complete Java project
- Complete CD model COMPLETED
- Pay current bill

Events

No events for today

Add event

Add event for 14-01-2022:

Menu

- Home
- Todo
- Custom
- Calendar

◀ January 2022 ▶

Add event

Event name
Rishi's birthday

Date
14-01-2022

Add

| SUN | MON | TUE | WED | THU | FRI | SAT |
|-----|-----|-----|-----|-----|-----|-----|
| 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 | 01 | 02 | 03 | 04 | 05 |

2022-1-14

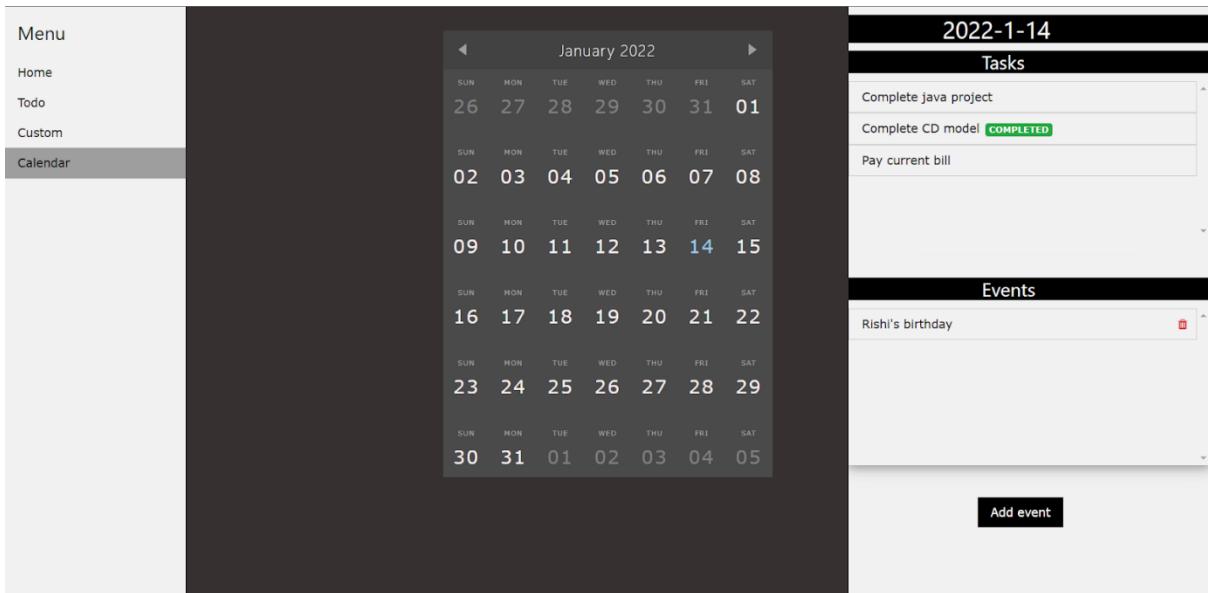
Tasks

- Complete java project
- Complete CD model COMPLETED
- Pay current bill

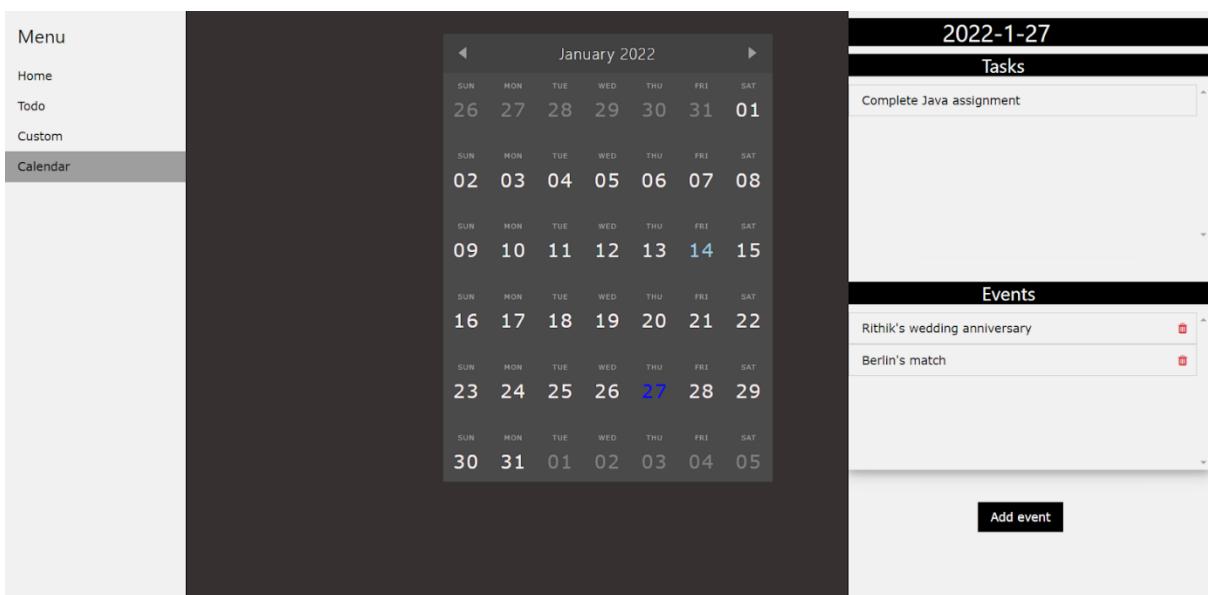
Events

No events for today

Add event



Adding events for a different date:



Deleting event:

The screenshot shows a calendar application interface. On the left is a vertical menu with options: Home, Todo, Custom, and Calendar (which is selected). The main area displays a monthly calendar for January 2022. A modal window titled "localhost:8080 says" is open, displaying the message "Deleting Event" with an "OK" button. The calendar grid shows dates from 01 to 31. The 27th is highlighted in blue, indicating it is selected or being edited.

localhost:8080 says

Deleting Event

OK

| SUN | MON | TUE | WED | THU | FRI | SAT |
|-----|-----|-----|-----|-----|-----|-----|
| 02 | 03 | 04 | 05 | 06 | 07 | 08 |
| 09 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 | 01 | 02 | 03 | 04 | 05 |

2022-1-27

Tasks

Complete Java assignment

Events

Rithik's wedding anniversary

Berlin's match

Add event

2022-1-27

Tasks

Complete Java assignment

Events

Rithik's wedding anniversary

Add event

