1. Modelo (Model)

O Modelo é responsável por gerenciar os dados e a lógica de negócio. No seu código, o Modelo é composto por duas classes: `Task` e `TaskManager`.

- **Classe `Task`**:
  - Esta classe representa uma tarefa individual.
  - Possui propriedades como:
    - `description`: A descrição da tarefa.
    - `dueDate`: A data de vencimento da tarefa.
    - `completed`: Um booleano que indica se a tarefa foi concluída (inicialmente `false`).
- **Classe `TaskManager`**:
  - Gerencia um array de tarefas.
  - Métodos:
    - `addTask(description, dueDate)`: Cria uma nova tarefa usando a classe `Task` e adiciona ao array de tarefas.
    - `removeTask(index)`: Remove uma tarefa do array com base no índice fornecido.
    - `toggleTaskCompletion(index)`: Alterna o status de conclusão de uma tarefa (marca como concluída ou não).
    - `getTasks()`: Retorna a lista atual de tarefas.

2. Visão (View)

A Visão é a parte que lida com a interface do usuário. No seu código, isso inclui o HTML e o CSS que criam a estrutura visual da aplicação.

- **HTML**:
  - Define a estrutura da interface com elementos como:
    - Um título (`<h1>`) para a lista de tarefas.
    - Um campo de texto (`<input type="text">`) para o usuário adicionar uma nova tarefa.
    - Um campo de data (`<input type="date">`) para selecionar a data de vencimento.
    - Um botão (`<button>`) que usa um ícone de adição para enviar a nova tarefa.

        - Uma lista (`<ul>`) onde as tarefas adicionadas serão exibidas.
- **CSS**:
    - Estiliza a interface, incluindo a cor de fundo, cores do texto, espaçamento e layout dos elementos.
    - Garante que a aplicação seja responsiva e tenha uma aparência atraente.

3. Controlador (Controller)

O Controlador é o intermediário que conecta a Visão e o Modelo. Ele manipula a lógica de interação do usuário.

- **Eventos e Manipulação**:
    - O controlador escuta eventos de interação do usuário, como o envio do formulário de tarefas.
    - Quando o formulário é enviado:
        - O controlador coleta os dados dos campos (descrição da tarefa e data de vencimento).
        - Chama o método `addTask` do `TaskManager` para adicionar a nova tarefa.
        - Atualiza a Visão para exibir a nova tarefa na lista.
- **Atualização da Lista**:
    - O controlador também é responsável por atualizar a interface sempre que uma tarefa é adicionada, removida ou marcada como concluída. Isso pode incluir re-renderizar a lista de tarefas a partir do Modelo.

Resumo do Funcionamento

1. **Adicionar Tarefas**:
    a. O usuário preenche o campo de descrição e a data, e clica no botão de adicionar.
    b. O controlador captura esses dados, cria uma nova tarefa no modelo e atualiza a visão para incluir a nova tarefa.
2. **Marcar Tarefas como Concluídas**:
    a. O usuário pode clicar em uma tarefa para marcá-la como concluída.
    b. O controlador altera o status da tarefa no modelo e atualiza a visão para refletir essa mudança.
3. **Remover Tarefas**:
    a. O usuário pode remover uma tarefa, o controlador chama o método correspondente no modelo e a visão é atualizada.