Aplicação Web, votação candidatos.

Disponibilizar arquivos seguindo o seguinte padrão de pastas:

index.html

/js

— model.js

— view.js

— controller.js

— app.js

Após realização de teste no código disponibilizado a seguir, realizar a seguinte modificação.

* Incluir botão para remover voto por item da lista ( candidato)
* Incluir formulário para inserir novo candidato na lista
* Substituir trecho de código de montagem fixa de candidatos, para inserção dinâmica no array, conforme recebimento de dados do formulário do index.

Modelo (js)

Gerencia os dados dos candidatos e os votos.

class VotingModel {

constructor() {

this.candidates = [

{ name: "Candidato 1", votes: 0 },

{ name: "Candidato 2", votes: 0 },

{ name: "Candidato 3", votes: 0 },

];

}

addVote(index) {

this.candidates[index].votes++;

}

removeCandidate(index) {

this.candidates.splice(index, 1);

}

getCandidates() {

return this.candidates;

}

}

Visão

Gerencia o DOM, exibindo os candidatos e atualizando os resultados.

class VotingView {

constructor() {

this.candidateList = document.getElementById("candidateList");

this.addVoteButtons = [];

}

displayCandidates(candidates) {

this.candidateList.innerHTML = "";

candidates.forEach((candidate, index) => {

const li = document.createElement("li");

li.textContent = `${candidate.name} - Votos: ${candidate.votes}`;

const voteButton = document.createElement("button");

voteButton.textContent = "Votar";

voteButton.dataset.index = index;

const removeButton = document.createElement("button");

removeButton.textContent = "Remover";

removeButton.dataset.index = index;

li.appendChild(voteButton);

li.appendChild(removeButton);

this.candidateList.appendChild(li);

});

}

}

Controlador

Gerencia as interações entre o modelo e a visão.

class VotingController {

constructor(model, view) {

this.model = model;

this.view = view;

this.view.candidateList.addEventListener("click", (e) => {

const index = e.target.dataset.index;

if (e.target.textContent === "Votar") {

this.model.addVote(index);

} else if (e.target.textContent === "Remover") {

this.model.removeCandidate(index);

}

this.updateView();

});

this.updateView();

}

updateView() {

this.view.displayCandidates(this.model.getCandidates());

}

}

HTML

<ul id="candidateList"></ul>

Inicialização

const model = new VotingModel();

const view = new VotingView();

const controller = new VotingController(model, view);