E-Voting System - REST API Documentation

# Overview

This document outlines the RESTful API endpoints for an E-Voting System built with ASP.NET Core and Entity Framework Core using a MySQL database. The system supports roles like Voter and Admin, and provides functionalities for elections, candidates, voting, and result viewing.

# Authentication (JWT-based)

## Voter Auth

|  |  |  |
| --- | --- | --- |
| Method | Endpoint | Description |
| POST | /api/auth/voter/register | Register a voter |
| POST | /api/auth/voter/login | Login a voter |

## Admin Auth

|  |  |  |
| --- | --- | --- |
| Method | Endpoint | Description |
| POST | /api/auth/admin/register | Register an admin |
| POST | /api/auth/admin/login | Login an admin |

# Voter APIs

|  |  |  |
| --- | --- | --- |
| Method | Endpoint | Description |
| GET | /api/voters/{id} | Get voter profile |
| PUT | /api/voters/{id} | Update voter profile |

# Admin APIs

|  |  |  |
| --- | --- | --- |
| Method | Endpoint | Description |
| GET | /api/admins | List all admins |
| GET | /api/admins/{id} | Get admin by ID |
| DELETE | /api/admins/{id} | Delete an admin |

# Election APIs

## Public

|  |  |  |
| --- | --- | --- |
| Method | Endpoint | Description |
| GET | /api/elections | Get all elections |
| GET | /api/elections/{id} | Get election by ID |

## Admin Only

|  |  |  |
| --- | --- | --- |
| Method | Endpoint | Description |
| POST | /api/admin/elections | Create an election |
| PUT | /api/admin/elections/{id} | Update an election |
| DELETE | /api/admin/elections/{id} | Delete an election |

# Candidate APIs

## Public

|  |  |  |
| --- | --- | --- |
| Method | Endpoint | Description |
| GET | /api/elections/{id}/candidates | Get candidates by election |

## Admin Only

|  |  |  |
| --- | --- | --- |
| Method | Endpoint | Description |
| POST | /api/admin/elections/{id}/candidates | Add candidate |
| PUT | /api/admin/candidates/{id} | Update candidate |
| DELETE | /api/admin/candidates/{id} | Delete candidate |

# Vote APIs

## Voter Only

|  |  |  |
| --- | --- | --- |
| Method | Endpoint | Description |
| POST | /api/votes | Cast a vote |
| GET | /api/votes/voter/{voterId} | View your vote |

## Admin Only

|  |  |  |
| --- | --- | --- |
| Method | Endpoint | Description |
| GET | /api/admin/votes | List all votes |

# Results APIs

|  |  |  |
| --- | --- | --- |
| Method | Endpoint | Description |
| GET | /api/elections/{id}/results | View election results |
| GET | /api/elections/{id}/live-tally | View live vote tally |

# Notes

- All sensitive endpoints should be secured with JWT authentication.  
- Use role-based access control (`[Authorize(Roles = "Admin")]`, `[Authorize]` etc.).  
- Ensure `Vote` is only allowed once per election per voter.

# Status Codes

- 200 OK - Request succeeded  
- 201 Created - Resource successfully created  
- 400 Bad Request - Invalid input  
- 401 Unauthorized - Not logged in or token expired  
- 403 Forbidden - Not allowed  
- 404 Not Found - Resource not found  
- 500 Internal Server Error - Server error

# Swagger/OpenAPI

To enable Swagger UI:  
```  
builder.Services.AddSwaggerGen();  
app.UseSwagger();  
app.UseSwaggerUI();  
```  
Then navigate to `/swagger` in your browser.

# Contact

For more details, reach out to the development team or project maintainer.