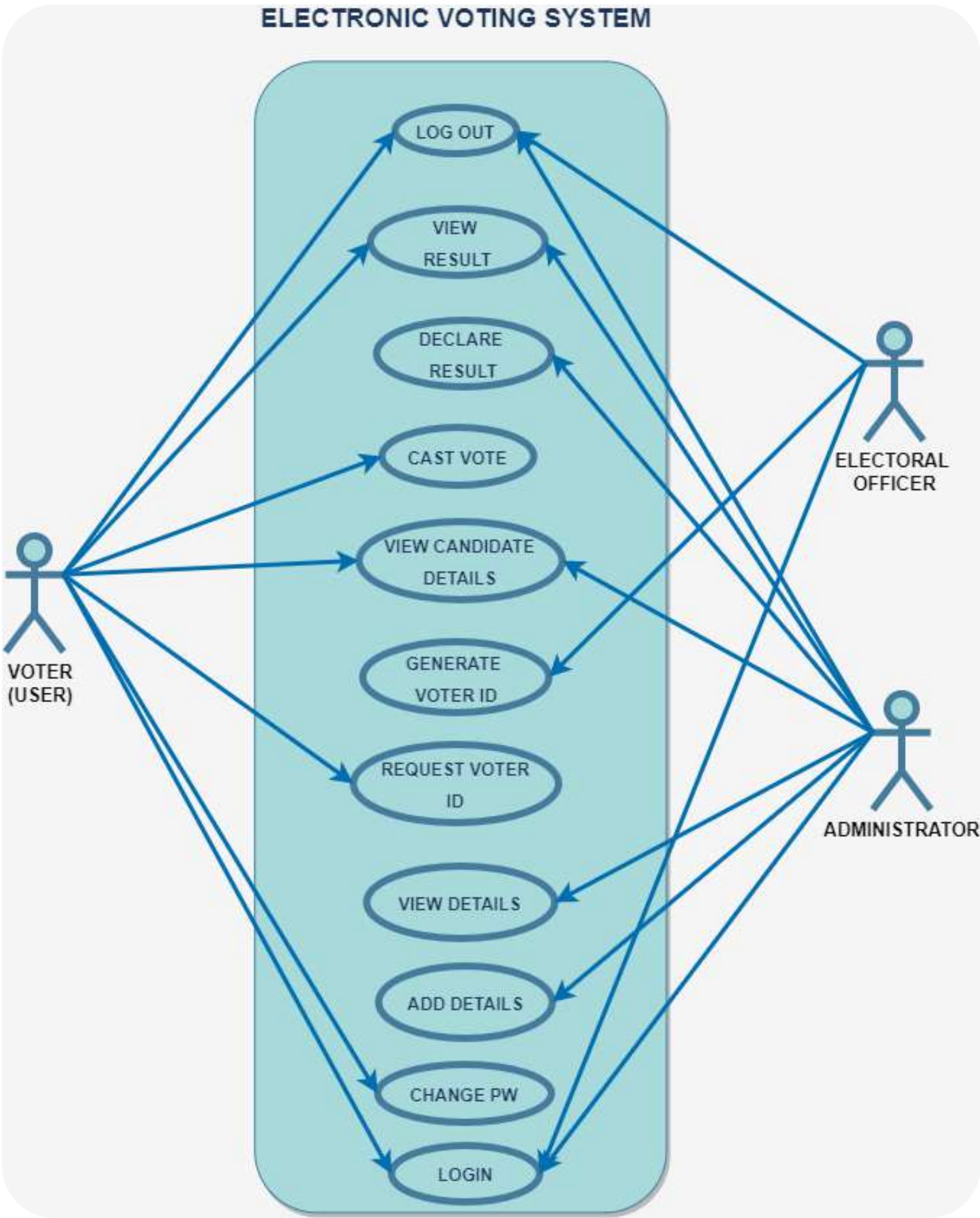
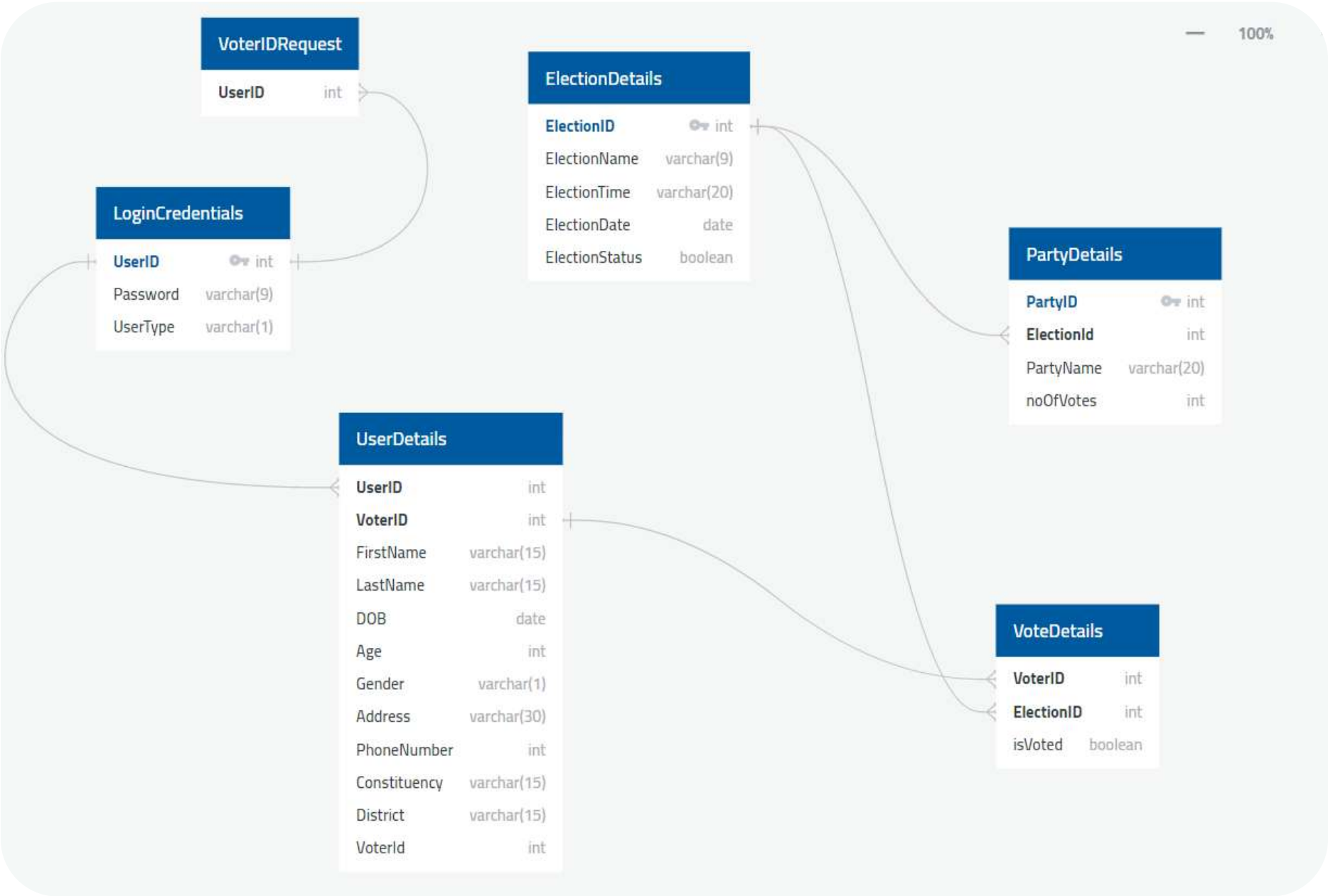


TEAM 4

ELECTRONIC VOTING SYSTEM

ADID	Name	Requirement
SH40114838	Shoubhik	US001,AD003, AD005,AD007
KA40115193	Keerthi	AD-010,US-007
AM40116111	Koushik	EX-001
SU40115879	Suraj	AA-001,AD-001, AD-002,AD-004, AD-006,AD-007, AD-008,EO-001, EO-002,US-002, US-003,US-004, US-006

# DB VIEW & USE CASE



# Landing Page

V Ind

My **Vote.**

My **Right.**


—Voting is not only our right

it is our power

Are you new here? **Register**


Login

Register



V Ind

Home



Login In to Voter INDIA

Not registered yet ? **Register**

User ID

Unique user ID

Password

\*\*\*\*\*

Login

V Ind

Home

First Name

Enter your first name

DOB

dd-mm-yyyy

Last Name

Enter your last name

Age

age

Phone

Enter your number

District

Enter your district

Address

Enter your address

Constituency

Enter your constituency

Register

V Ind

Home

Give your details

Your UserID

39

Your password

\*\*\*\*\*

Login

Register



# Landing Page Controller

```
// ROOT PAGE
@RequestMapping("/")
public String index() {
    return "index";
}

// LOGIN PAGE
@RequestMapping("/loginPage")
public String loginPage() {
    return "loginPage";
}

// REGISTER PAGE
@RequestMapping("/registerPage")
public ModelAndView registerPage() {

    // Create a new object to Login Credentials
    LoginCredentials loginCred = new LoginCredentials();

    // Set the userType to 'U' for User
    loginCred.setUserType('U');

    // Save the object to the database using DATA JPA
    LoginCredentials registerCred = loginRepo.save(loginCred);

    // Return the register page with unique user id
    ModelAndView mv = new ModelAndView("registerPage");
    mv.addObject("userid", registerCred.getUserId());

    return mv;
}
```

```
@RequestMapping("/login")
public ModelAndView login(LoginCredentials loginCred) {

    // LOGIN ACTION - Check if the user exists in the database

    Optional<LoginCredentials> currentLoginCred = loginRepo.findById(loginCred.getUserId());
    if (currentLoginCred.isPresent()) {
        if (currentLoginCred.get().getPassword().equals(loginCred.getPassword())) {
            return redirectUserToDestination(currentLoginCred.get());
        } else {
            ModelAndView mv = new ModelAndView("loginPage");
            mv.addObject("error", "Invalid Password");
            return mv;
        }
    } else {
        ModelAndView mv = new ModelAndView("loginPage");
        mv.addObject("error", "Invalid UserId");
        return mv;
    }
}

@RequestMapping("/register")
public ModelAndView register(UserDetails user, int userid, String password) {

    // Set the password to LoginCredentials object and save it to the database
    LoginCredentials loginCred = new LoginCredentials(userid, password, 'U');
    loginRepo.save(loginCred);

    // Set the user id to UserDetails object
    user.setUserId(userid);
    userRepo.save(user);

    // add object to the model and redirect to the login page
    ModelAndView mv = new ModelAndView("loginPage");
    mv.addObject("msg", "User Registered Successfully");

    return mv;
}
```

```

// Redirect the user to the appropriate page
public ModelAndView redirectUserToDestination(LoginCredentials lc) {
    ModelAndView mv = new ModelAndView();
    UserDetails userDetails = userRepo.findById(lc.getUserId()).get();
    mv.addObject("userDetail", userDetails);
    if (lc.getUserType() == 'A') {
        mv.setViewName("admin/homePage");
        return mv;
    } else if (lc.getUserType() == 'E') {
        Iterable<VoterIdRequest> pending = voterIdRepo.findAll();
        mv.addObject("pending", pending);
        mv.setViewName("electoral/homePage");
        return mv;
    } else {
        if (userDetails.getVoterId() != 0 && userDetails.getVoterId() != 1) {
            try {
                VoteDetails voteDetails = voteRepo.findById(userDetails.getVoterId()).get();
                ElectionDetails electionDetails =
electionRepo.findById(voteDetails.getElectionId()).get();
                mv.addObject("electionDetails", electionDetails);
            } catch (Exception e) {
                mv.addObject("electionDetails", null);
            }
        }
        mv.setViewName("voter/homePage");
        return mv;
    }
}

```


# Voter Portal

V Ind

Log Out

Welcome

Suraj



Voter ID

Don't have a Voter ID? [Request](#)


Election has not been Scheduled Yet.

V Ind

Log Out

Welcome

Suraj




ELECTION NAME	ELECTION TIME	ELECTION DATE	ELECTION STATUS	ELECTION ACTION
PM	14:00	2022-06-01	On Going	<div>VOTE</div>

V Ind

Log Out





Welcome

Suraj



Voter ID

19134996

PARTY LOGO	PARTY NAME	VOTE
	Leo	<input type="radio"/>
	Virgo	<input type="radio"/>
	Libra	<input type="radio"/>
	Pisces	<input type="radio"/>

Cast Vote

V Ind

Log Out

Welcome

Suraj



ELECTION NAME	ELECTION TIME	ELECTION DATE	ELECTION STATUS	ELECTION ACTION
PM	14:00	2022-06-01	End	<div>RESULTS</div>



# Voter Controller

```
@RequestMapping("/requestVoterId")
public ModelAndView requestVoterId(@RequestParam int userId) {
    ModelAndView mv = new ModelAndView();
    UserDetails userDetails = userRepo.findById(userId).get();
    userDetails.setVoterId(1);
    userRepo.save(userDetails);
    mv.addObject("userDetail", userDetails);
    mv.setViewName("voter/homePage");
    return mv;
}

@RequestMapping("/voteAction")
public ModelAndView voteAction(@RequestParam int userId, @RequestParam int electionId) {
    ModelAndView mv = new ModelAndView();
    UserDetails userDetails = userRepo.findById(userId).get();
    VoteDetails voteDetails = voteRepo.findById(userDetails.getVoterId()).get();
    boolean isVoted = voteDetails.isVoted();
    Iterable< PartyDetails> partyList = partyRepo.findByElectionId(electionId);
    mv.addObject("partyList", partyList);
    mv.addObject("electionId", electionId);
    mv.addObject("isVoted", isVoted);
    mv.addObject("userDetail", userDetails);
    mv.setViewName("voter/votingPage");
    return mv;
}

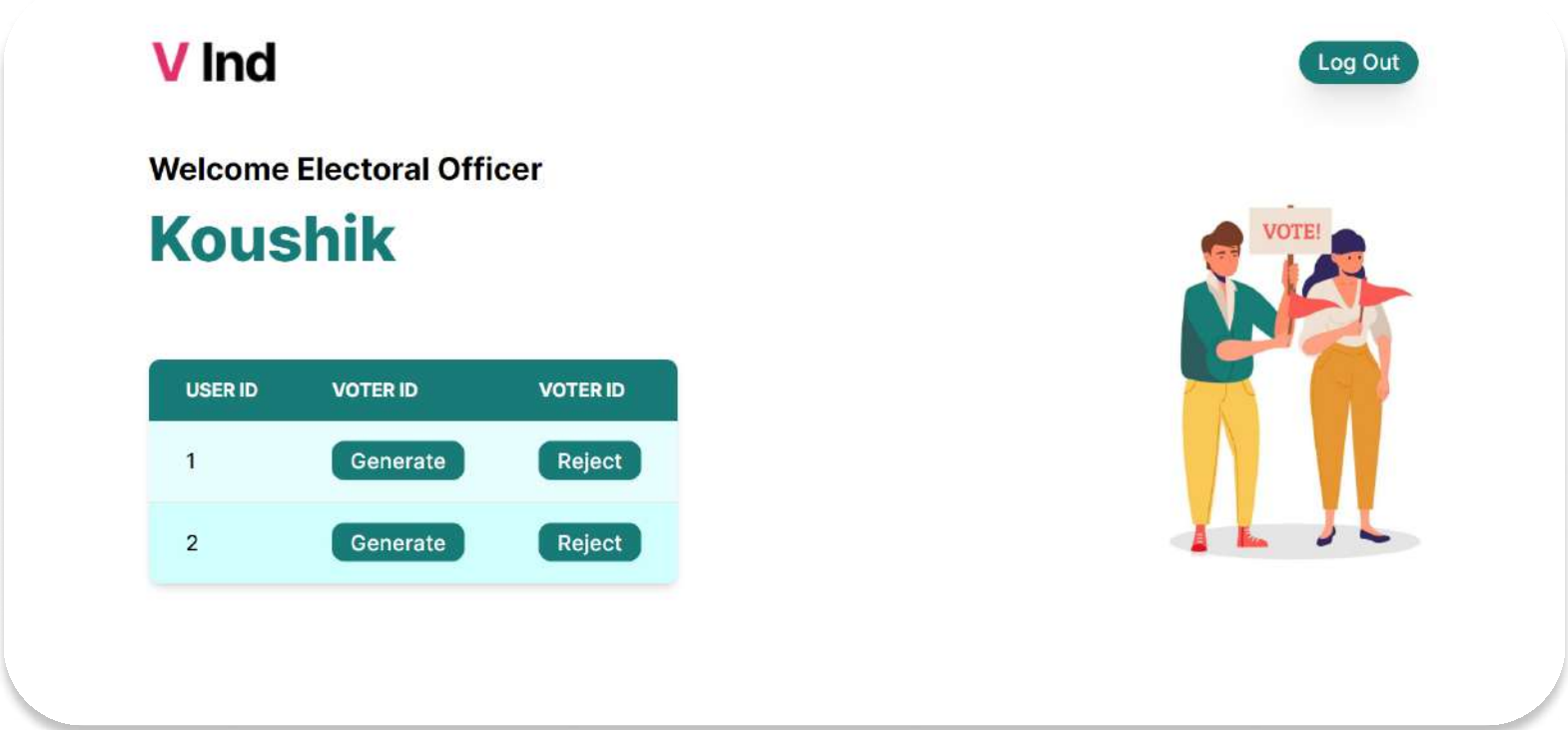
@RequestMapping("/castVote")
public ModelAndView castVoter(@RequestParam int userId, @RequestParam int electionId,
@RequestParam int partyId) {
    ModelAndView mv = new ModelAndView();
    UserDetails userDetails = userRepo.findById(userId).get();

    VoteDetails voteDetails = voteRepo.findById(userDetails.getVoterId()).get();
    voteDetails.setVoted(true);
    voteRepo.save(voteDetails);
    mv.addObject("isVoted", voteDetails.isVoted());

    PartyDetails partyDetails = partyRepo.findById(partyId).get();
    partyDetails.setNoOfVotes(partyDetails.getNoOfVotes() + 1);
    partyRepo.save(partyDetails);

    if (userDetails.getVoterId() != 0 && userDetails.getVoterId() != 1) {
        try {
            ElectionDetails electionDetails =
electionRepo.findById(voteDetails.getElectionId()).get();
            mv.addObject("electionDetails", electionDetails);
        } catch (Exception e) {
            mv.addObject("electionDetails", null);
        }
    }
    mv.addObject("msg", "Voted successfully");
    mv.addObject("userDetail", userDetails);
    mv.setViewName("voter/homePage");
    return mv;
}
```

# Electoral Officer Portal



## EO Controller

```
@RequestMapping("generateVoterId")
public ModelAndView generateVoterId(@RequestParam int userId, @RequestParam int currentUserId,
    @RequestParam Boolean reject) {
    ModelAndView mv = new ModelAndView();

    // Generate a 8 digit random number
    if (!reject) {
        int voterId = 0;
        do {
            voterId = (int) (Math.random() * 1000000000);
        } while (voterIdRepo.findById(voterId).isPresent());

        UserDetails generatedVoterUser = userRepo.findById(userId).get();
        generatedVoterUser.setVoterId(voterId);
        userRepo.save(generatedVoterUser);
    }

    // Remove Generated VoterID User from pending list
    voterIdRepo.deleteById(userId);

    UserDetails userDetails = userRepo.findById(currentUserId).get();
    mv.addObject("userDetail", userDetails);

    Iterable<VoterIdRequest> pending = voterIdRepo.findAll();
    mv.addObject("pending", pending);
    if (!reject) mv.addObject("msg", "Voter ID generated successfully");
    else mv.addObject("msg", "Voter ID rejected successfully");
    mv.setViewName("electoral/homePage");
    return mv;
}
```




# Admin Portal

V Ind

Welcome ADMIN

Athma



Log Out

Add Details

ELECTION DETAILS

PARTY DETAILS

CANDIDATE DETAILS

VOTER DETAILS

View Details

ELECTION DETAILS


PARTY DETAILS

CANDIDATE DETAILS

V Ind

Welcome ADMIN

Athma




Log Out

USER ID	USER NAME	VOTER ID
1	Athma	Not Requested
2	Suraj	Approve
3	Shoubhik	11862418
4	Koushik	Not Requested
5	Keerthi	11946745

V Ind

Welcome ADMIN

Athma



Log Out

Add Election Details

Election ID

Add election id

Election Date

dd-mm-yyyy

Election Name

Add election Name

Election Status

Add election status

Election Time


--:-- --

Add

V Ind

Welcome ADMIN

Athma



Log Out

Add Party Details

Party ID

Add party id

Election ID

Add election id

Party Name

Add party name

Add

Welcome ADMIN

Athma



Add Candidate Details

Voter ID

Election ID

Add

Welcome ADMIN

Athma



ELECTION ID	ELECTION NAME	ELECTION TIME	ELECTION DATE	ELECTION ACTION
1	CM	16:00	2222-02-02	Result
2	PM	14:00	2022-06-01	Declare

Welcome ADMIN

Athma



PARTY ID	ELECTION ID	PARTY NAME	NO OF VOTES
1	2	Leo	0
2	2	Virgo	0
3	2	Libra	0
4	2	Pisces	0

Welcome ADMIN

Athma



VOTER ID	ELECTION ID
11862418	2
11946745	2
19134996	2





## Admin Controller

```
@RequestMapping("addElection")
public ModelAndView addElection(@RequestParam int currentUserId) {
    ModelAndView mv = new ModelAndView();
    UserDetails currentUser = userRepo.findById(currentUserId).get();
    mv.addObject("userDetail", currentUser);
    mv.setViewName("admin/add/electionDetails");
    return mv;
}

@RequestMapping("addElectionDetails")
public ModelAndView addElectionDetails(@RequestParam int currentUserId, ElectionDetails
electionDetails){
    ModelAndView mv = new ModelAndView();
    electionRepo.save(electionDetails);
    UserDetails currentUser = userRepo.findById(currentUserId).get();
    mv.addObject("userDetail", currentUser);
    mv.addObject("msg", "Election details added successfully");
    mv.setViewName("admin/homePage");
    return mv;
}

@RequestMapping("addParty")
public ModelAndView addParty(@RequestParam int currentUserId) {
    ModelAndView mv = new ModelAndView();
    UserDetails currentUser = userRepo.findById(currentUserId).get();
    mv.addObject("userDetail", currentUser);
    mv.setViewName("admin/add/partyDetails");
    return mv;
}

@RequestMapping("addPartyDetails")
public ModelAndView addPartyDetails(@RequestParam int currentUserId, PartyDetails partyDetails){
    ModelAndView mv = new ModelAndView();
    partyRepo.save(partyDetails);
    UserDetails currentUser = userRepo.findById(currentUserId).get();
    mv.addObject("userDetail", currentUser);
    mv.addObject("msg", "Party details added successfully");
    mv.setViewName("admin/homePage");
    return mv;
}

@RequestMapping("addCandidate")
public ModelAndView addCandidate(@RequestParam int currentUserId) {
    ModelAndView mv = new ModelAndView();
    UserDetails currentUser = userRepo.findById(currentUserId).get();
    mv.addObject("userDetail", currentUser);
    mv.setViewName("admin/add/candidateDetails");
    return mv;
}

@RequestMapping("addCandidateDetails")
public ModelAndView addCandidateDetails(@RequestParam int currentUserId, VoteDetails voteDetails){
    ModelAndView mv = new ModelAndView();
    voteDetails.setVoted(false);
    voteRepo.save(voteDetails);
    UserDetails currentUser = userRepo.findById(currentUserId).get();
    mv.addObject("userDetail", currentUser);
    mv.addObject("msg", "Vote details added successfully");
    mv.setViewName("admin/homePage");
    return mv;
}
```



```

@GetMapping("allVoter")
public ModelAndView showAllVoterDetails(@RequestParam int currentUserId){
    Iterable<UserDetails> allUser = userRepo.findAll();
    ModelAndView mv = new ModelAndView("admin/view/userDetails");
    mv.addObject("allUser", allUser);

    UserDetails currentUser = userRepo.findById(currentUserId).get();
    mv.addObject("currentUser", currentUser);
    return mv;
}

@GetMapping("allElection")
public ModelAndView showAllElectionDetails(@RequestParam int currentUserId){
    Iterable<ElectionDetails> electionDetails = electionRepo.findAll();
    ModelAndView mv = new ModelAndView("admin/view/electionDetails");
    mv.addObject("allElection", electionDetails);

    UserDetails currentUser = userRepo.findById(currentUserId).get();
    mv.addObject("currentUser", currentUser);
    return mv;
}

@GetMapping("allParty")
public ModelAndView showAllPartyDetails(@RequestParam int currentUserId){
    Iterable<PartyDetails> partyDetails = partyRepo.findAll();
    ModelAndView mv = new ModelAndView("admin/view/partyDetails");
    mv.addObject("allParty", partyDetails);

    UserDetails currentUser = userRepo.findById(currentUserId).get();
    mv.addObject("currentUser", currentUser);
    return mv;
}

@GetMapping("allCandidate")
public ModelAndView showAllCandidateDetails(@RequestParam int currentUserId){
    Iterable<VoteDetails> voteDetails = voteRepo.findAll();
    ModelAndView mv = new ModelAndView("admin/view/candidateDetails");
    mv.addObject("allCandidate", voteDetails);

    UserDetails currentUser = userRepo.findById(currentUserId).get();
    mv.addObject("currentUser", currentUser);
    return mv;
}

```

```

@RequestMapping("addVoterRequest")
public ModelAndView addVoterRequest(@RequestParam int userId, @RequestParam int currentUserId){
    ModelAndView mv = new ModelAndView();

    VoterIdRequest voterIdRequest = new VoterIdRequest();
    voterIdRequest.setUserId(userId);
    voterIdRepo.save(voterIdRequest);

    UserDetails currentUser = userRepo.findById(currentUserId).get();
    mv.addObject("userDetail", currentUser);
    mv.addObject("msg", "Voter ID Request sent to EO");
    mv.setViewName("admin/homePage");
    return mv;
}

@GetMapping("declareResult")
public ModelAndView declareResult(@RequestParam int electionId){
    // Creating a model view with name resultPage
    ModelAndView mv = new ModelAndView("admin/view/resultPage");

    // Getting party which has max no of votes
    PartyDetails partyDetails = partyRepo.findMaxByNoOfVotes();

    // Getting election details
    ElectionDetails electionDetails = electionRepo.findById(electionId).get();
    // Set election status to true that means it has been ended
    electionDetails.setElectionStatus(true);
    electionRepo.save(electionDetails);
    mv.addObject("partyDetails", partyDetails);
    return mv;
}

```