“Digital Electoral System for University Governance”

*Software Requirements Specification* (SRS)

### Table of Contents

1. [Introduction](#_v7t71yszvzha)
2. [Purpose](#_l66o92yentrx)
3. [Scope](#_5y7lotupcw1t)
4. [Definitions, Acronyms, Abbreviations](#_ybrope1ajchl)
5. [References](#_rjbup6xpqzbb)
6. [Overall Description](#_kio5j4vtjz05)
7. [Overall Description](#_ax0t1vb64hq0)
8. [Product Perspective](#_rzr0fhyf7vkl)
9. [Product Functions](#_usxzn4lbdp0p)
10. [User Characteristics](#_eh4v90i5ohe2)
11. [User Constraints](#_xxai7m8phbfu)
12. [Assumptions & Dependencies](#_inq2m361tn98)
13. [Apportioning Requirements](#_tvjwe28kr5v6)
14. [Specific Requirements](#_3tg2wzp5if8v)
15. [Interface Requirements](#_yqvtzeyknfu3)
16. [Functional Requirements](#_p9zo90v232u7)
17. [Performance Requirements](#_enogb5nmavfh)
18. [Logical Database Requirements](#_tfdybeyil14s)
19. [Design Constraints](#_pnmklk1aagac)
20. [Software System Attributes](#_pnxomy5zkiw6)
21. [Supporting Information](#_c5r0d0sj1eq3)

- [Appendix-I](#_i2iuqvq1goh) ; [Appendix-II](#_9cyhbb39qm9p) ; [Appendix-III](#_we3txl8zn8dd) ; [Appendix-IV](#_9capivd8yqyg) ; [Appendix-V](#_8fzz9jm9941k) -

# INTRODUCTION

This section provides an overview of the entire SRS.

## Purpose

**a)** This Software Requirements Specification (SRS) aims to define the functional and non-functional requirements for developing a Digital Electoral System for University Governance. It outlines the system's features, functionalities, constraints, and performance expectations, serving as a development, testing, and deployment guide.

**b)** The intended audience for this SRS includes:

* University administrators such as Deans and Election Chairs are involved in managing university-level elections.
* Candidates participating in the nomination process.
* Voters who will use the online voting platform.

## Scope

**a)** The software product to be produced is the "Digital Electoral System for University Governance."

**b)** The software will:

* Facilitate candidate registration and nomination processes for university-level elections.
* Provide a secure online voting platform for eligible voters during specified election periods.
* Ensure user authentication, confidentiality, and anonymity of votes.
* Generate real-time election results and notifications for users.

The software will not:

* Conduct offline voting or manage physical ballots.
* Handle non-electoral tasks unrelated to the university governance system.

**c)** The Digital Electoral System aims to:

* Simplify the registration and voting processes for university elections.
* Increase participation and engagement among students, faculty, and administrators.
* Ensure the integrity, security, and transparency of the electoral system.
* Streamline election management tasks for administrators, reducing manual effort and errors.
* Reduce the costs required for the arrangements of an election.

## Definitions, Acronyms, Abbreviations

* SRS: Software Requirements Specification - The description of a particular product, program, or set of programs that perform a set of functions in a target environment.
* DBMS: Database Management System - Software for managing databases and handling data operations.
* GUI/UI: (Graphical) User Interface - The visual interface through which users interact with software.
* DESUG: Digital Electoral System for University Governance.

## References

The following documents are referenced in this SRS:

* ***Students’ Union Election Notification***
  + Date: November 01, 2023
  + Published by: Students’ Union Election Commission 2023 - 2024, University of Hyderabad
* ***Lyngdoh Committee Rules***
  + Date: May 23, 2006
  + Published by: Ministry of Human Resource Development,

Department of Secondary and Higher Education,

Government of India

* ***IEEE Recommended Practice for Software Requirements Specifications***
  + Date: October 20, 1998
  + Published by: Software Engineering Standards Committee of the IEEE Computer Society

## Overall Description

**a)** The rest of the Software Requirements Specification (SRS) contains detailed information about the Digital Electoral System for University Governance. It includes sections on the overall description of the system, specific requirements, and supporting information.

**b)** The SRS is organized into the following main sections:

* Introduction: Provides an overview of the entire SRS, including the purpose, intended audience, scope, and definitions.
* Overall Description: Describes the product perspective, functions, user characteristics, constraints, assumptions, and dependencies.
* Specific Requirements: Details interface requirements, functional requirements, performance requirements, database requirements, design constraints, and system attributes.
* Supporting Information: This may include appendices, additional documents, or supplementary information related to the Digital Electoral System.

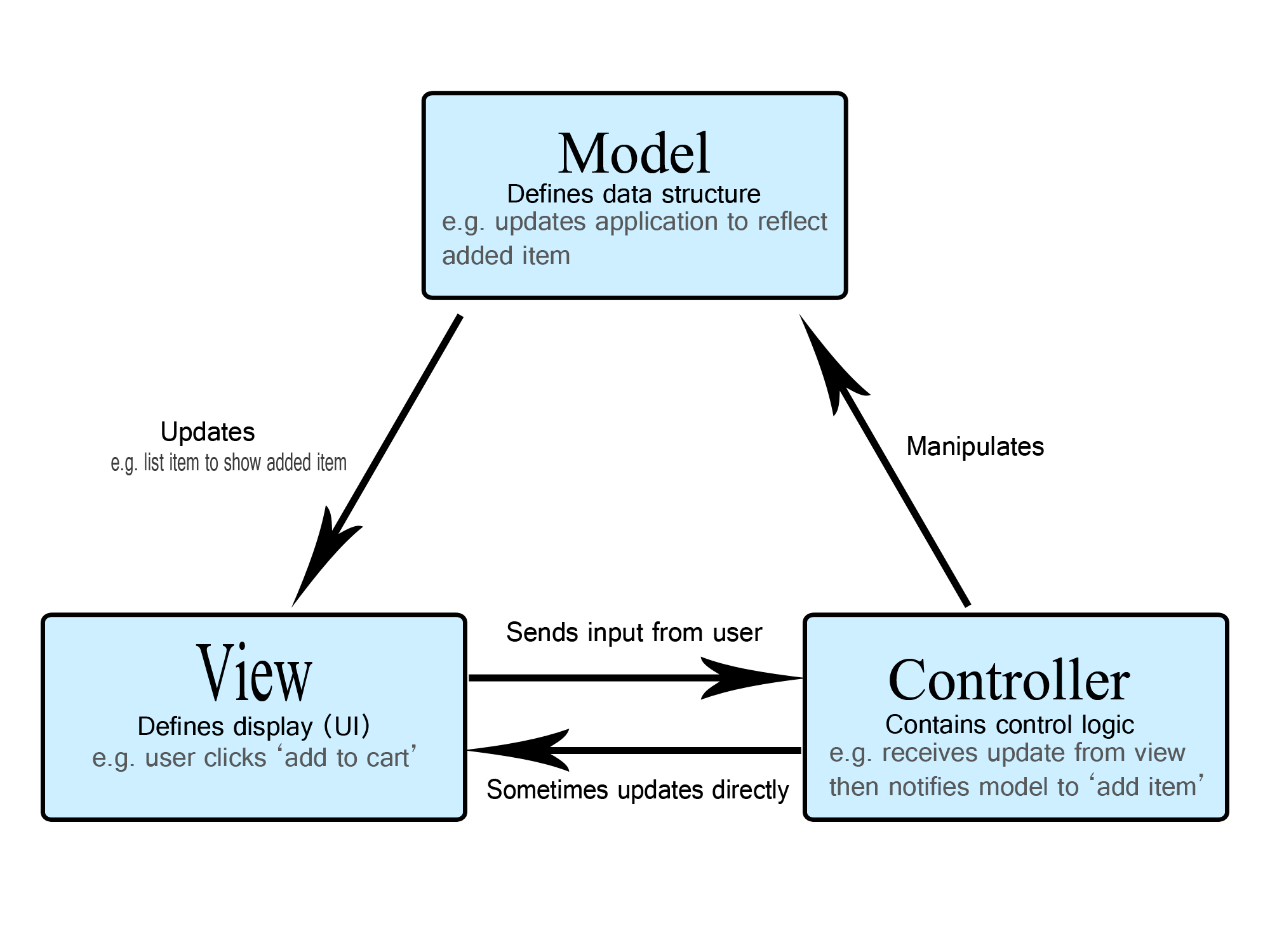
# OVERALL DESCRIPTION

This section of the SRS describes the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3 of the SRS, and makes them easier to understand.

## Product Perspective

The Digital Electoral System for University Governance is designed to operate as an independent and self-contained software product. It is not a component of a larger system but serves as a standalone solution specifically tailored for managing university-level elections.

The abstract architecture of the system:



([Source](https://developer.mozilla.org/en-US/docs/Glossary/MVC))

## Product Functions

The Digital Electoral System for University Governance encompasses several major functions that contribute to its overall operation and utility. These functions are summarized below in a way that is easily understandable to stakeholders and readers of the document:

* ***Login*** - Users log in securely using their university credentials to access the system.
* ***Cast Vote*** - Eligible voters can securely cast their votes online during specified election periods.
* ***View Results*** - Users can view real-time election results once voting is complete.
* ***File Nomination*** - Candidates register for elections and undergo approval processes by proposers, seconders, deans, and the election chair.
* ***Manage User Roles*** - The election chair can delegate work and manage user roles with varying privileges.
* ***Add/Remove Election*** - The election chair creates and manages elections, defining eligibility criteria and voting periods.
* ***Generate Results*** - Real-time results are generated and made available after the voting process is finished.

## User Characteristics

The intended users of the Digital Electoral System for University Governance possess certain general characteristics that influence the design and functionality of the software. These characteristics include:

1. Educational Level: Users range from university students to faculty members and administrators. They have varying levels of educational attainment, from undergraduate to postgraduate degrees.
2. Experience: Users may have diverse levels of experience with technology and electoral processes. Some users may be familiar with online voting systems, while others may require more guidance and support.
3. Technical Expertise: Users' technical expertise varies, from basic computer literacy to advanced knowledge of software applications. The system is designed to accommodate users with varying technical abilities.

## User Constraints

The development of the Digital Electoral System for University Governance is subject to various constraints that shape the design and implementation of the software. These constraints include:

a) Regulatory Policies: Compliance with university electoral policies, data protection regulations, and privacy laws.

b) Hardware Limitations: Adherence to hardware specifications and compatibility with standard computing devices (e.g., desktops, laptops, mobile devices).

c) Interfaces to Other Applications: Integration with university databases for user authentication and academic record verification.

d) Parallel Operation: Ensuring the system can handle concurrent user interactions and voting processes during peak periods without performance degradation.

e) Audit Functions: Implementing audit trails and logs to track system activities, user interactions, and voting events for accountability and transparency.

f) Control Functions: Implementing access controls, user roles, and permissions to manage and secure system functionalities.

g) Higher-Order Language Requirements: Utilizing programming languages and frameworks suitable for developing secure and scalable web-based applications.

h) Signal Handshake Protocols: Implementing reliable communication protocols (e.g., HTTPS, SSL/TLS) for secure data transmission and user interactions.

i) Reliability Requirements: Ensuring the system operates reliably without frequent downtime or disruptions during critical election periods.

j) Criticality of the Application: Recognizing the importance of the electoral system's integrity, security, and accuracy in university governance processes.

k) Safety and Security Considerations: Implementing robust security measures to protect user data, prevent unauthorized access, and ensure the confidentiality and anonymity of votes.

## Assumptions & Dependencies

The development and functionality, of the Digital Electoral System for University Governance, are influenced by various assumptions and dependencies. These factors, while not design constraints, can impact the requirements stated in the SRS.

## Apportioning Requirements

Certain requirements of the Digital Electoral System for University Governance may be deferred or delayed until future versions of the system. These requirements, while important, may not be critical for the initial release or may require additional development time or resources.

* Casting Vote
* Generating Results
* Managing multiple elections for various purposes
* Enhanced Security Features
* Accessibility Improvements
* Real-time Monitoring and Reporting

# SPECIFIC REQUIREMENTS

This section of the SRS contains all of the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements.

## Interface Requirements

1. **User Interface**

[see Appendix-I]

1. **Hardware Interface**

-

1. **Software Interface**

* HTML
* tailwindcss
* JavaScript
* JSP and Java Servlets
* MySQL DBMS
* JDBC for MySQL database connection.

1. **Communication Interface**

* Internet
* Email

## Functional Requirements

1. **Use Case Model**

[see Appendix-II]

1. **Use Case Specifications**

[see Appendix-III]

1. **Activity Diagram**

[see Appendix-IV]

1. **Analysis Classes**

[see Appendix-V]

## Performance Requirements

100s of requests per second (information based on Tomcat server specification)

## Logical Database Requirements

Relational Database Management System(RDBMS) - MySQL

MySQL Database Connector - JDBC

## Design Constraints

The guidelines/standards in the following documents have to be followed:

* Students Union Notification, UoH ([Source](http://intranet.uohyd.ac.in/wordpress/wp-content/uploads/2023/11/Students-Union-Elections-Notification-2023-24.pdf))
* Lyngdoh Committee Rules, Govt. of India ([Source](https://www.ugc.gov.in/oldpdf/students_pdf/lyngdoh_committeemhrd2712.pdf))

## Software System Attributes

1. **Reliability**

The system must have high reliability, particularly at the time of elections.

Testing must be done to check the reliability of the system.

1. **Availability**

Backup servers and databases must be used to ensure system availability and reduce downtime.

1. **Security**

Hashed passwords must be used to ensure they are not revealed in the network.

Encryption techniques or blockchain technologies must be used in recording votes to make them securely recorded and tamper-proof.

1. **Maintainability**

Modularity needs to be followed in the design and development process so that the addition of a new feature or the updation of an existing feature would be easy.

1. **Portability**

The application can be deployed on the cloud which can be accessed by a huge audience.

Also could be sold to other institutions/organizations or could be given to them by implementing various subscription plans.

# SUPPORTING INFORMATION

The supporting information includes the following:

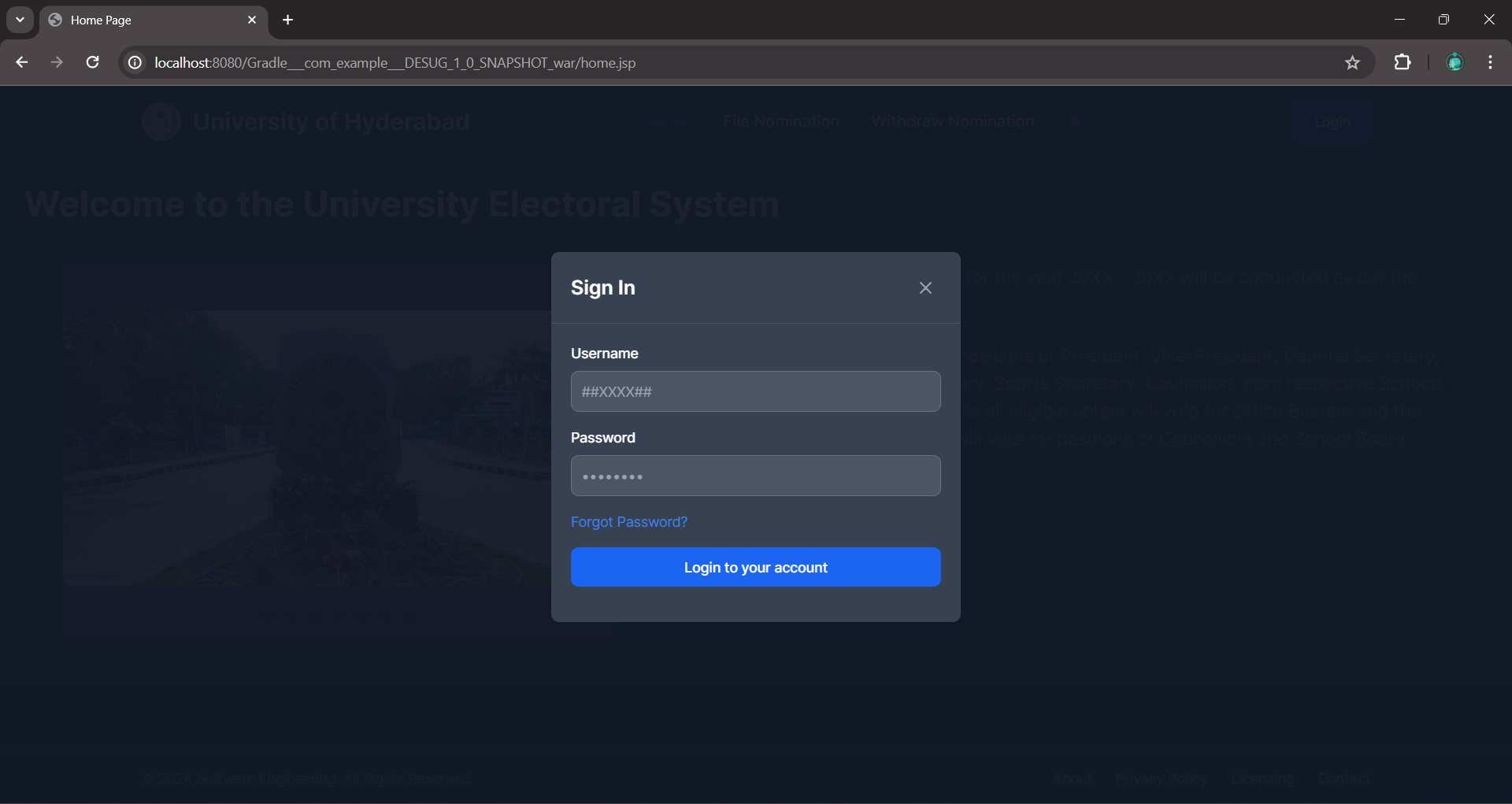
1. [Table of contents](#_e45rdukzaa6t) [see the beginning of this document]
2. Appendixes [see from the next page]

#### **Appendix-I:** User Interfaces

→ General Interfaces



**HomePage**

****

**Login**

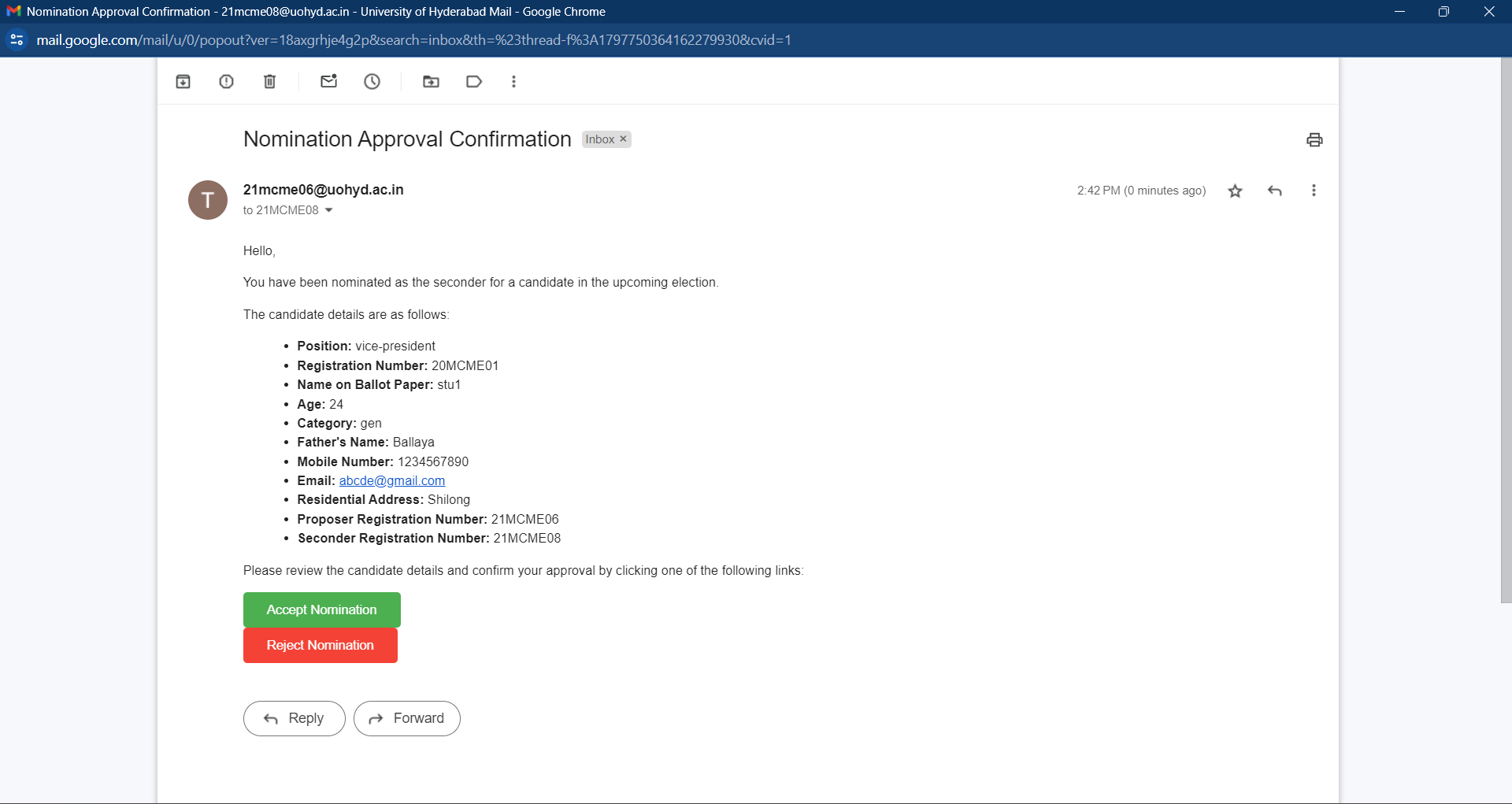


**Change Password**

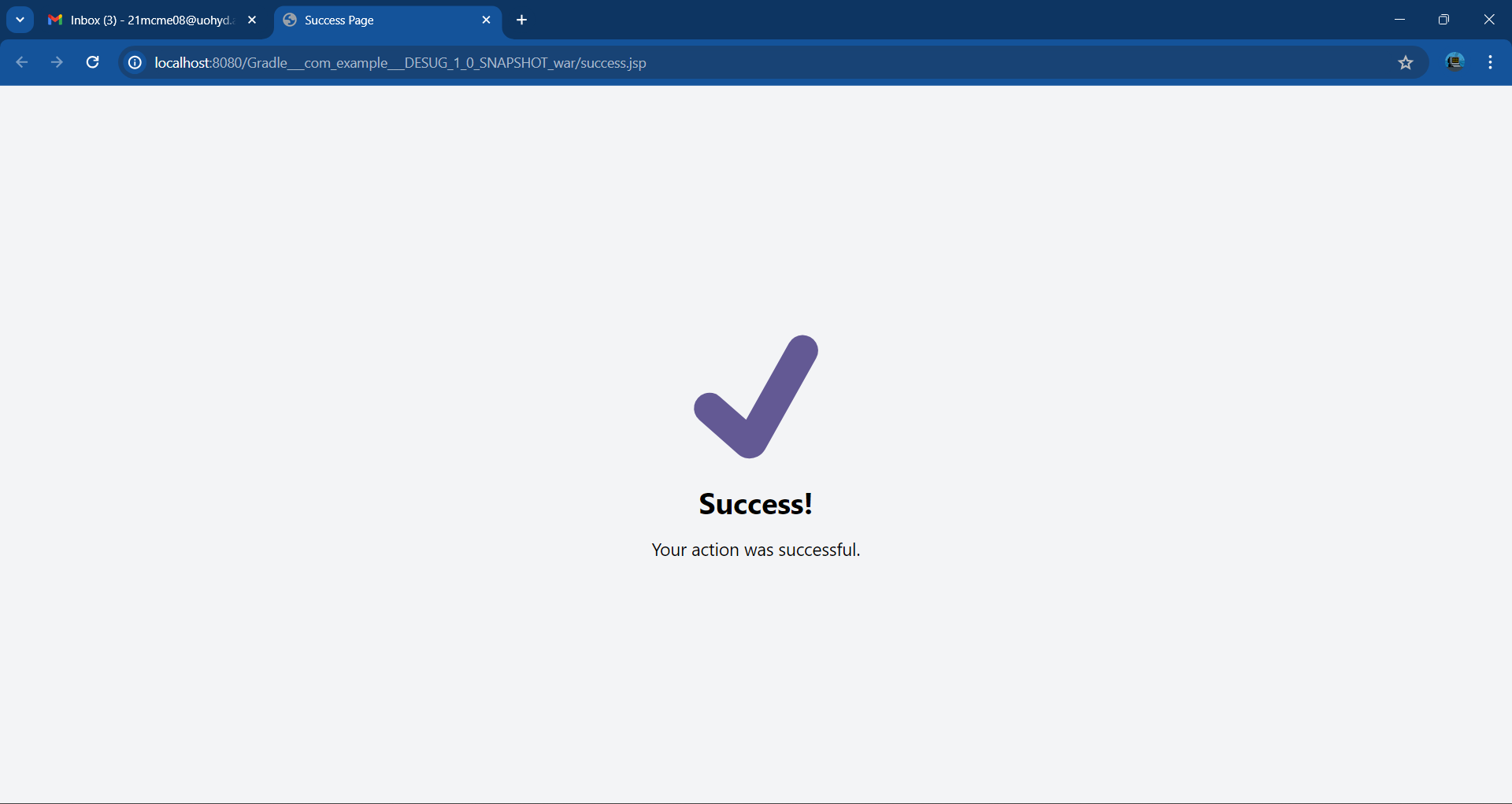


**Forgot Password**

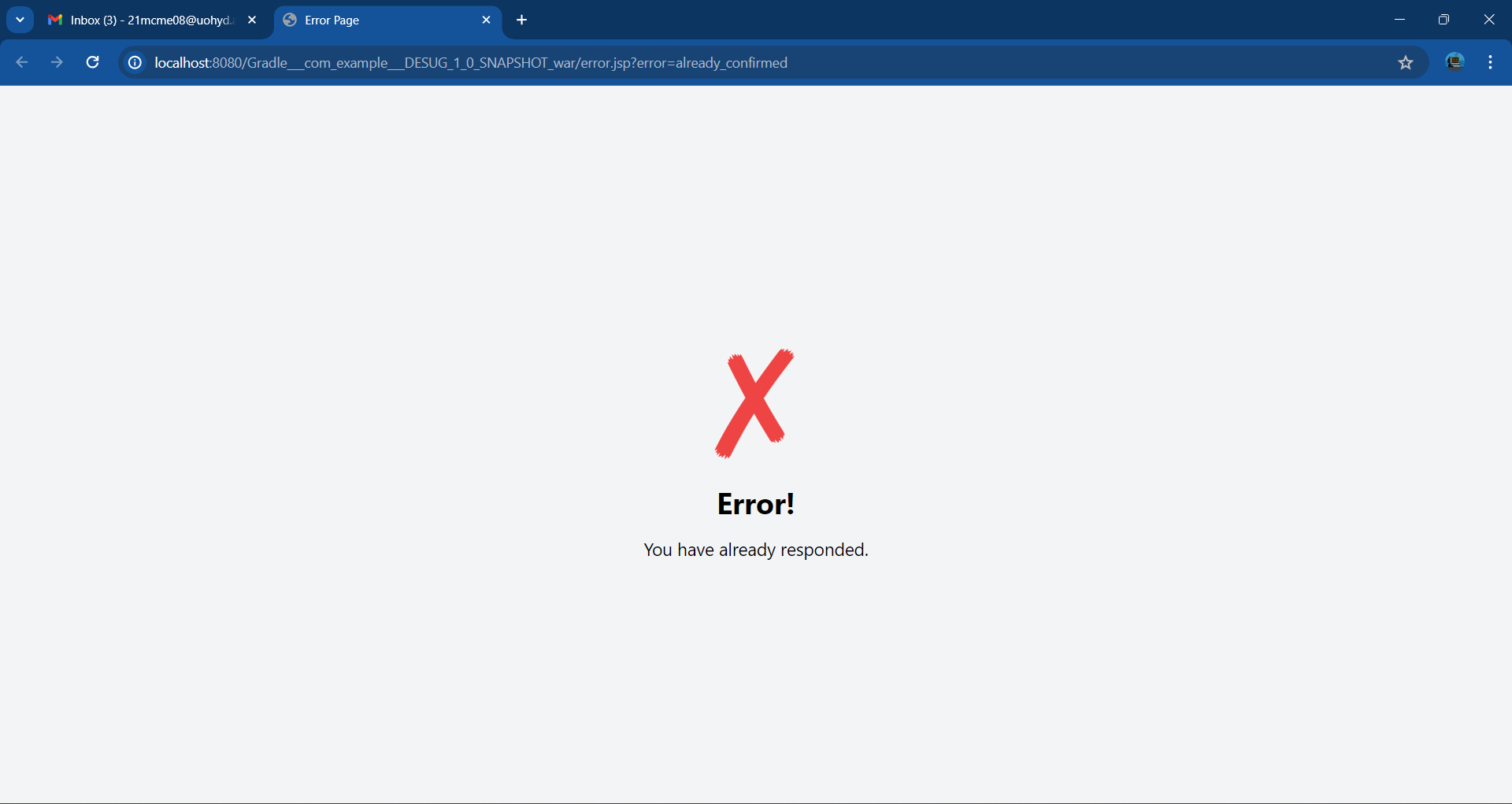
→ Automated Mails



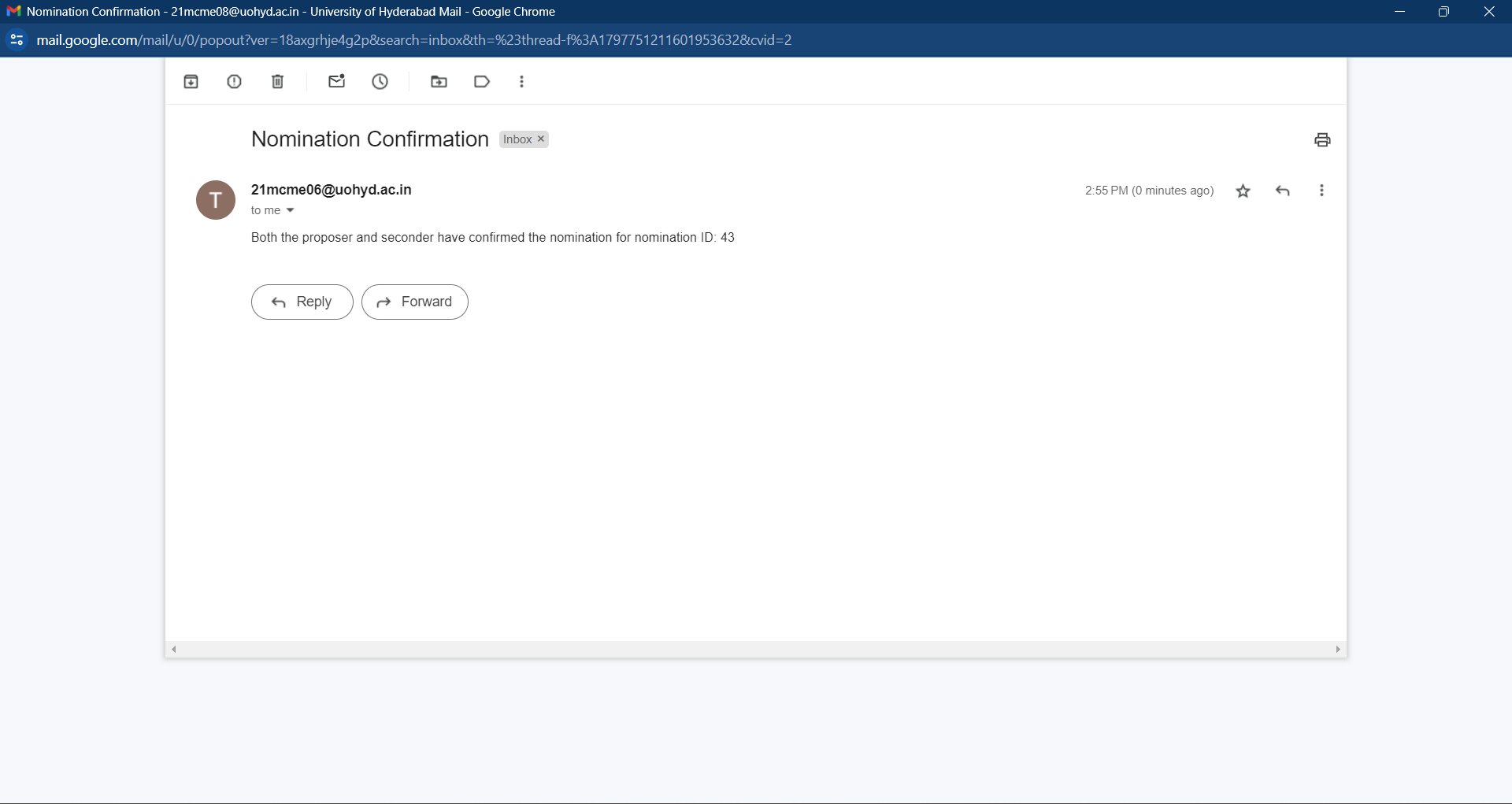
**Mail to Proposer/Seconder**

****

**Accepted/Rejected by Proposer/Seconder**

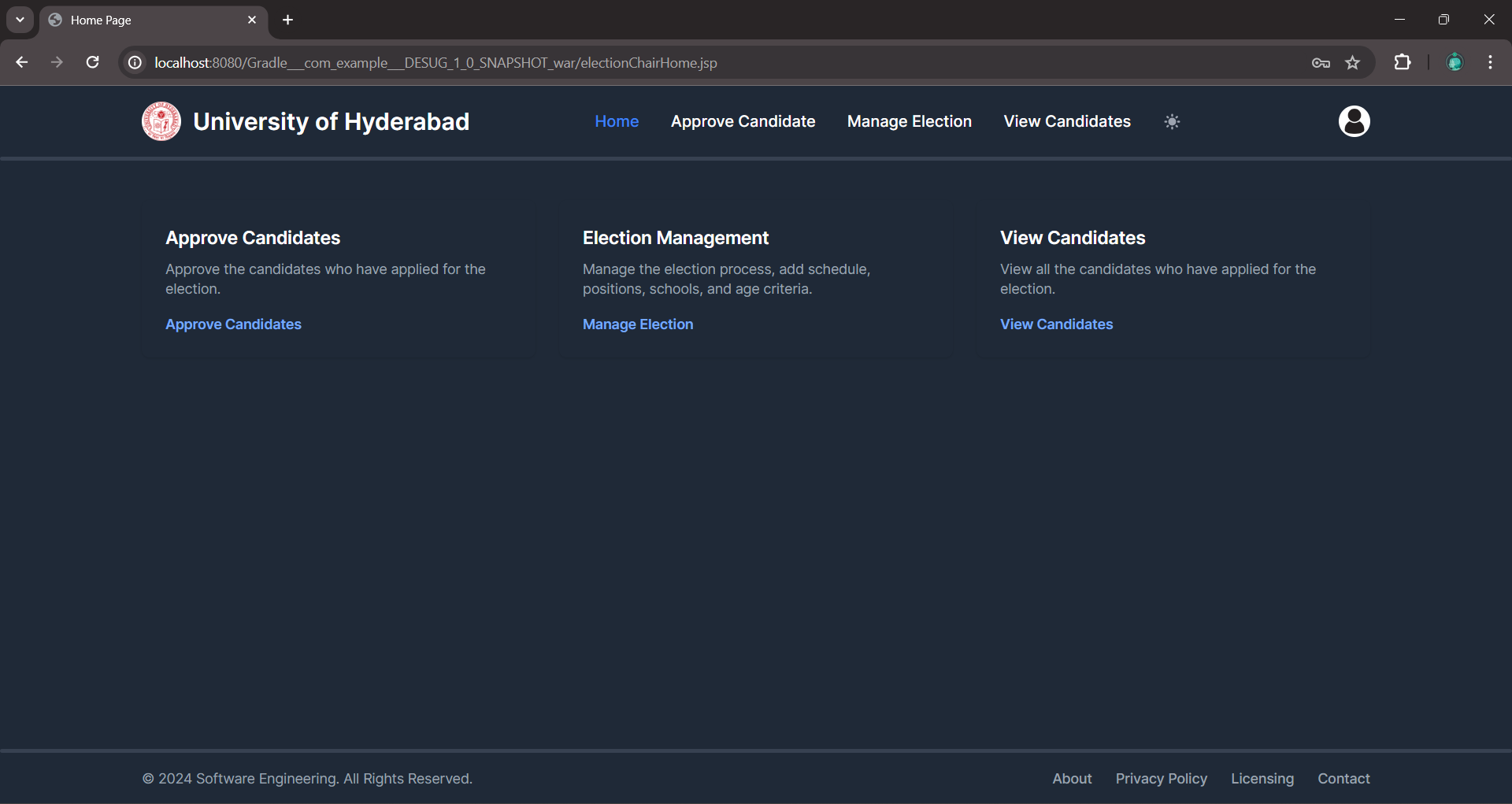


**Error in Acceptance/Rejection**

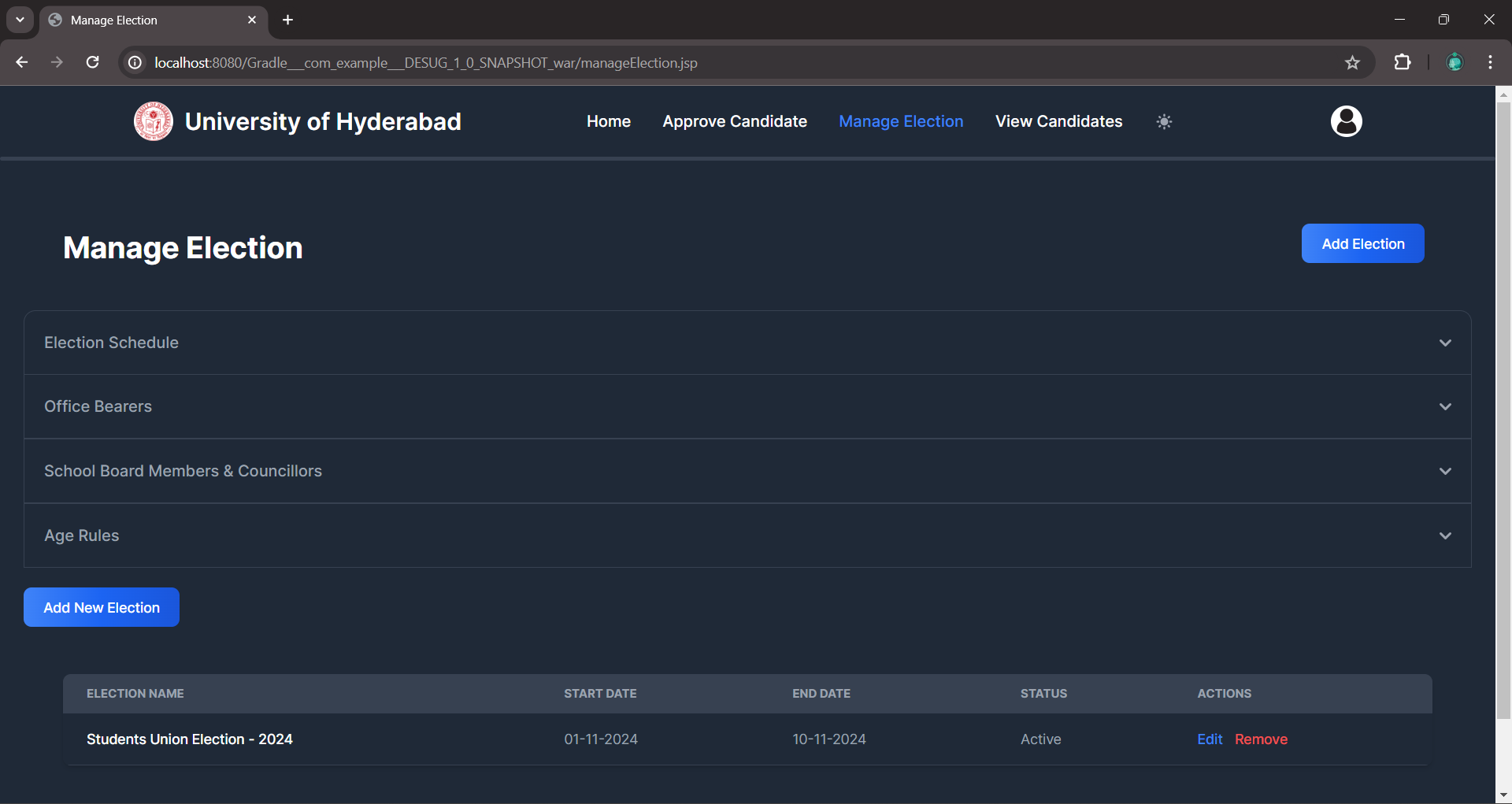


**Mail to Dean**

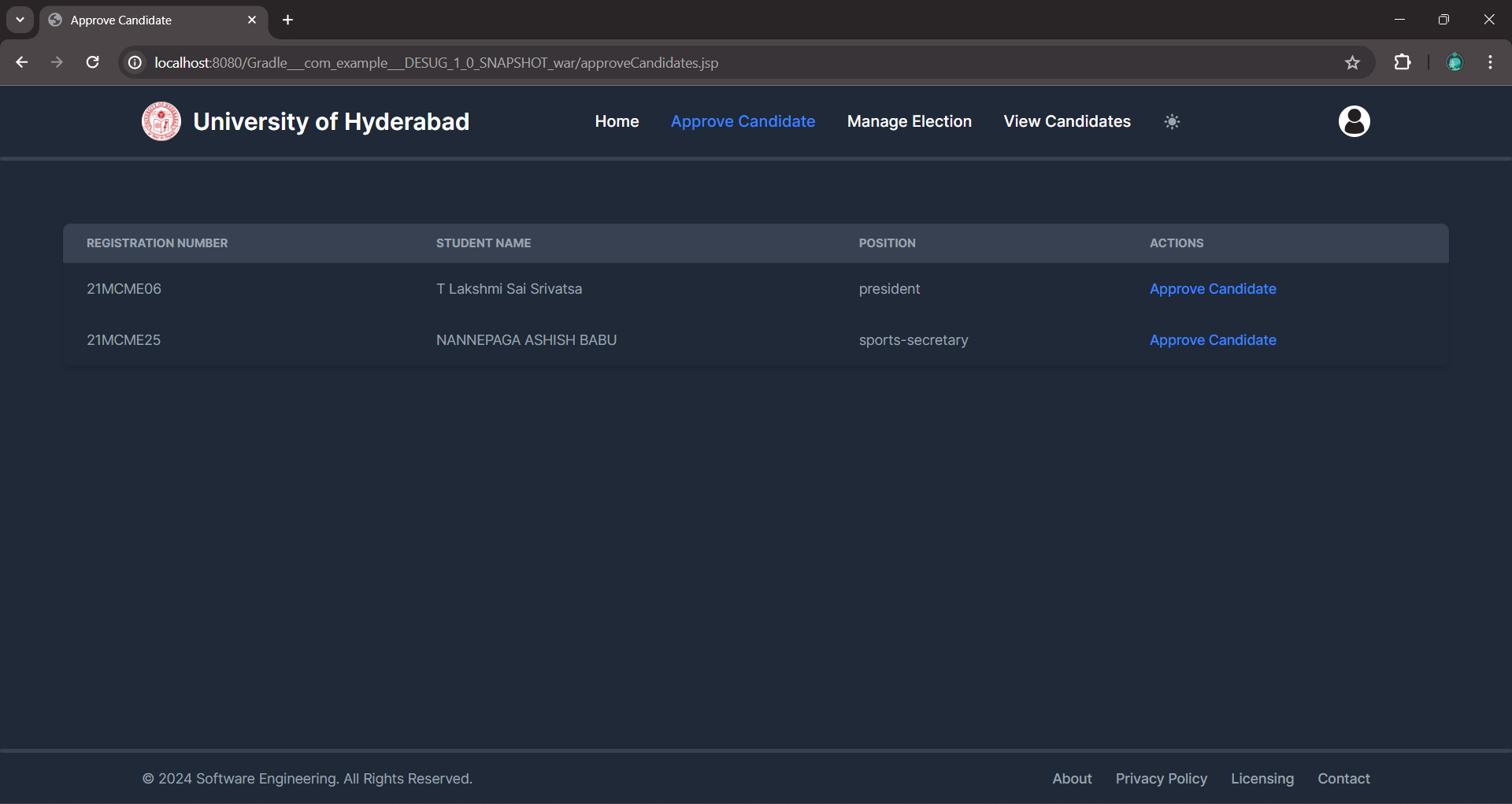
→ Election Chair Interfaces



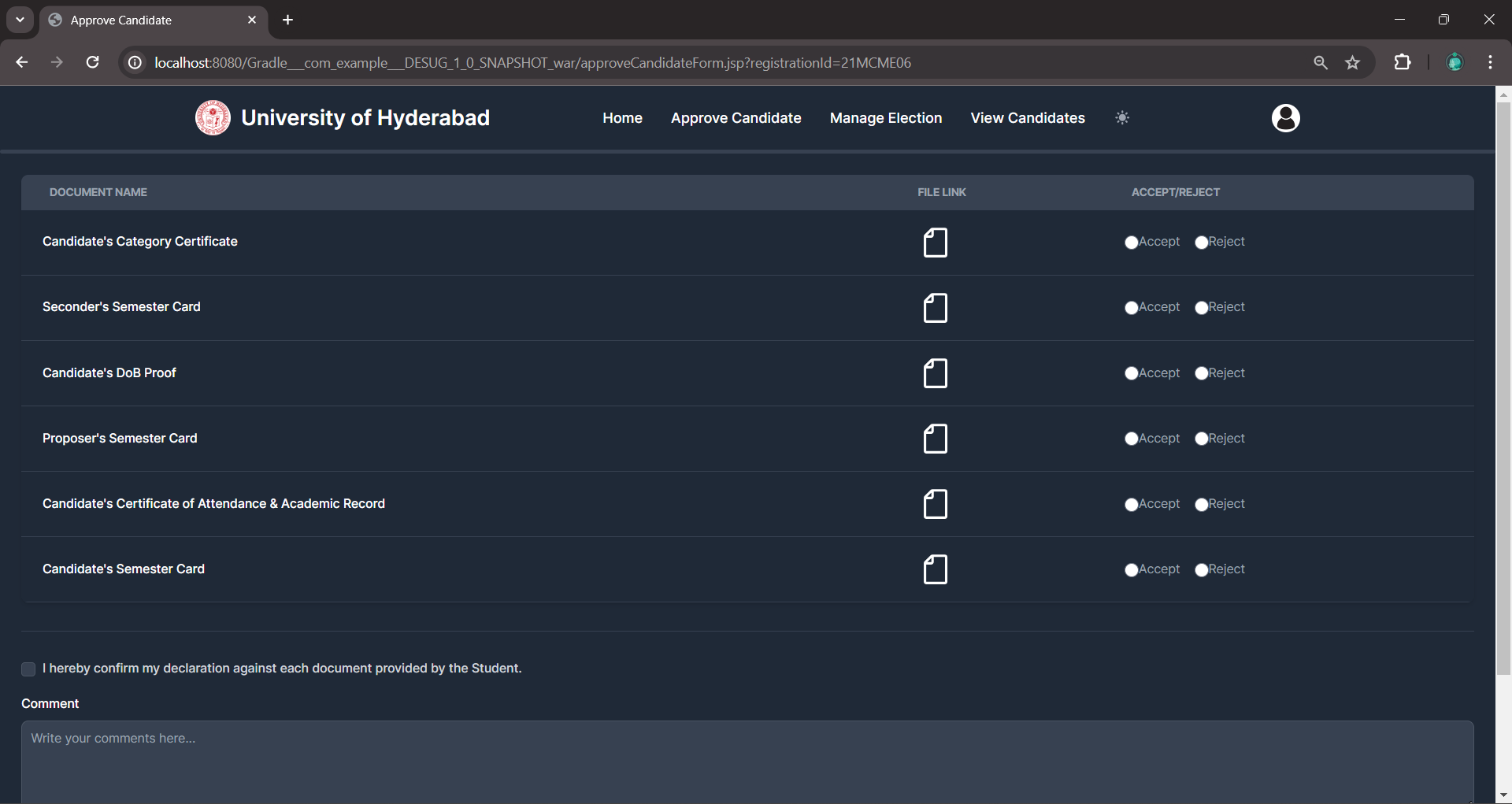
**Election Chair’s HomePage**



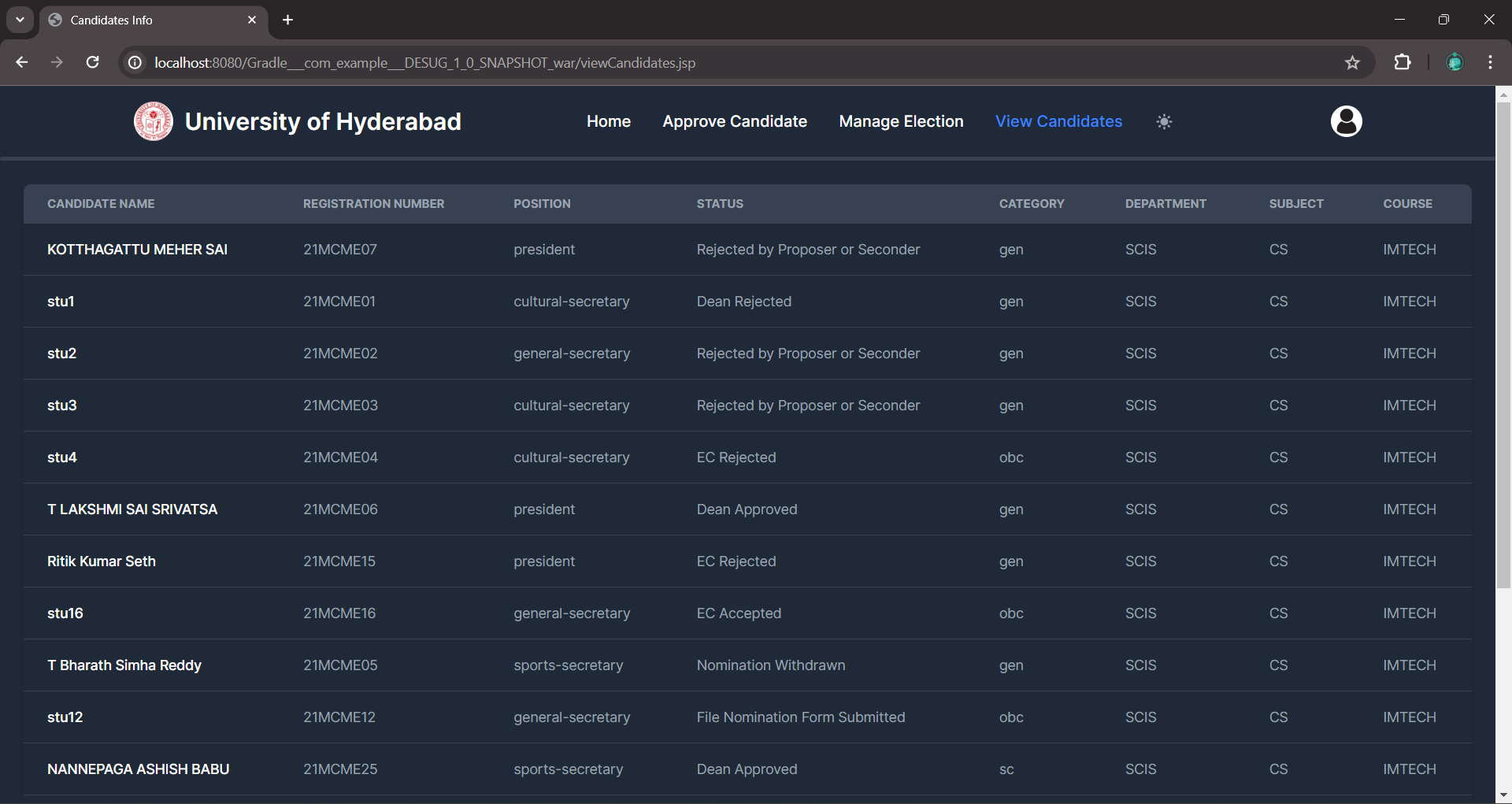
**Manage Election**



**List of Candidates to be Approved**

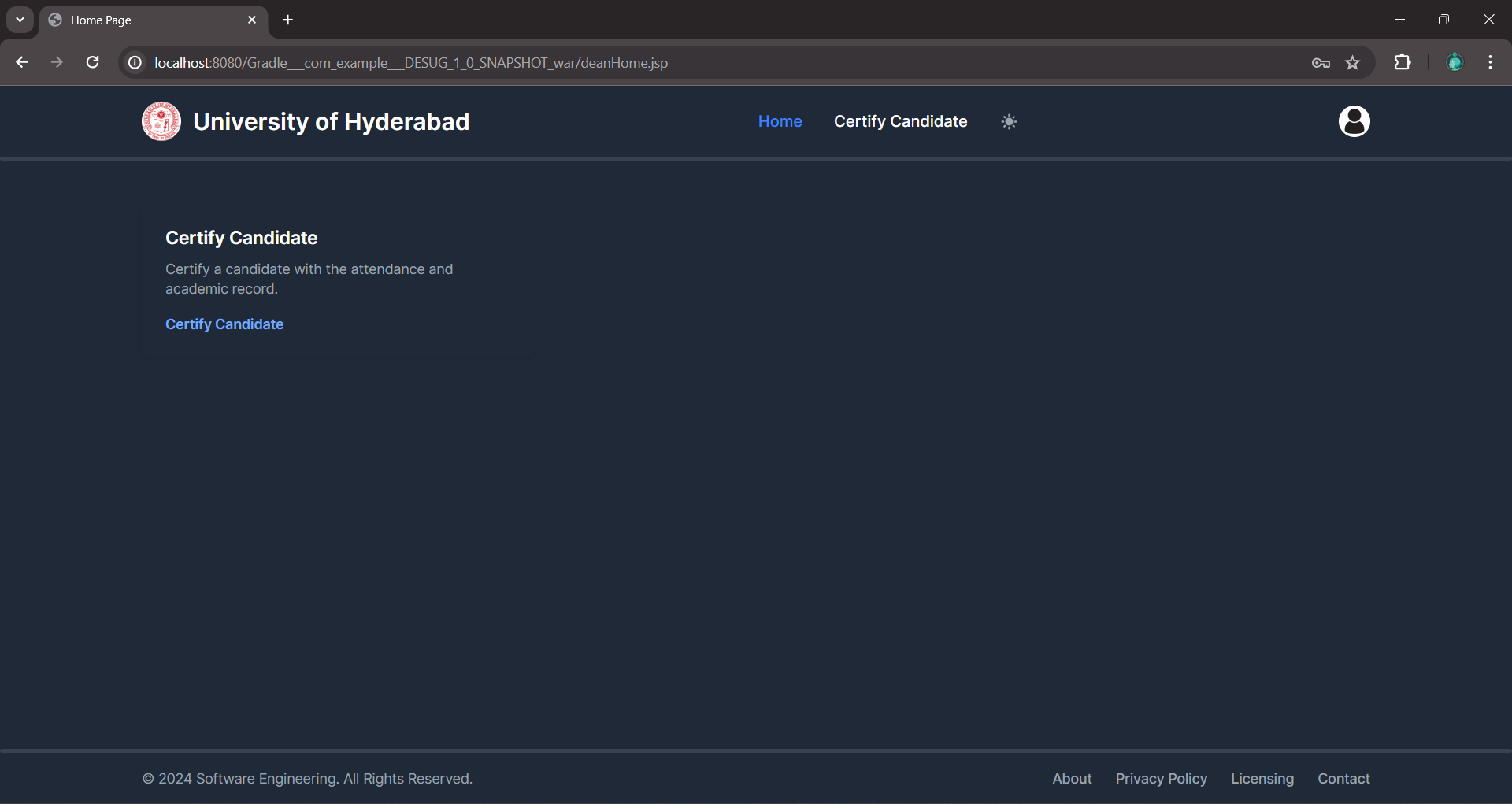


**Approve Candidate Form**

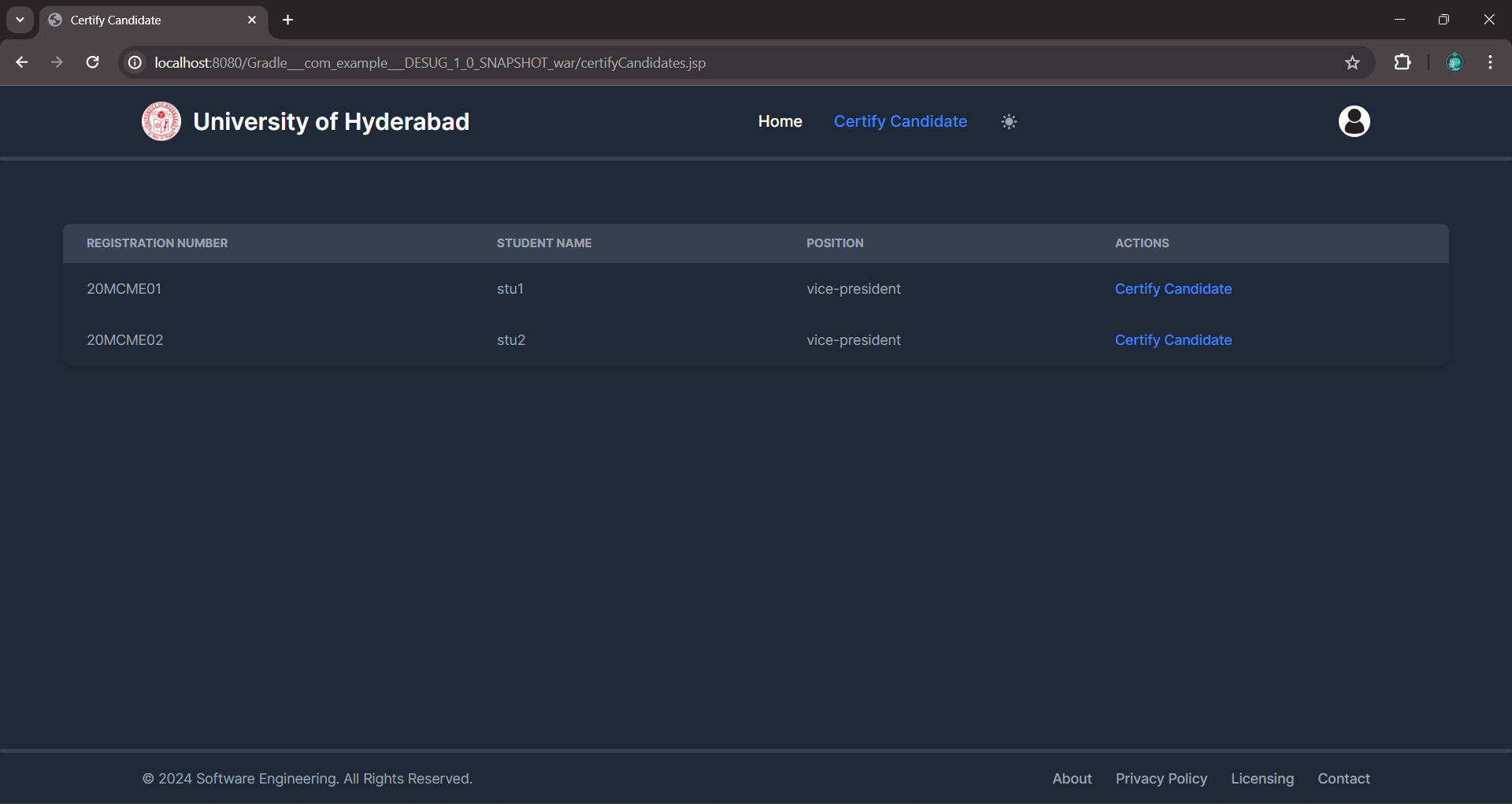


**All Candidates List with their Status**

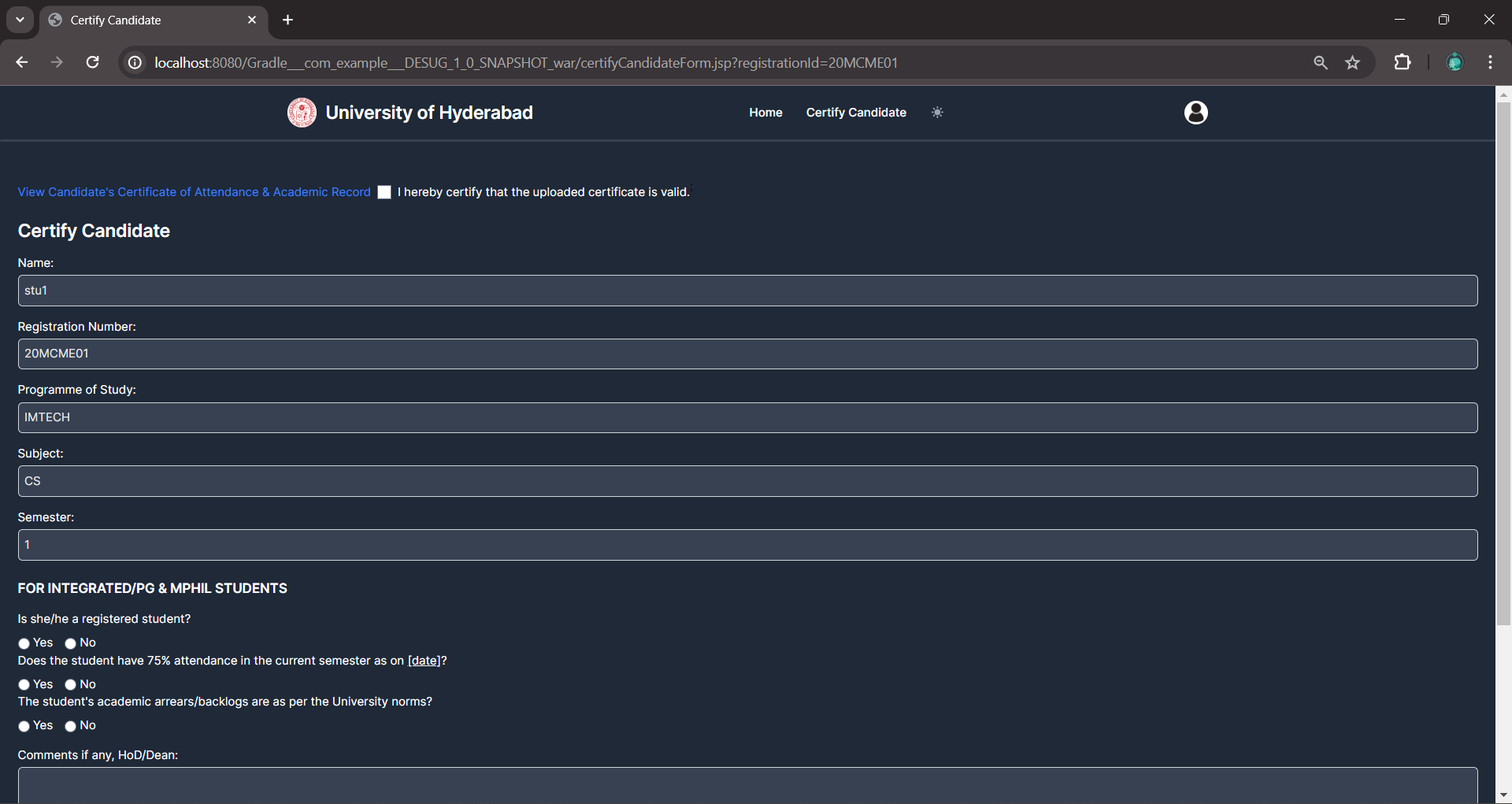
→ Dean Interfaces



**Dean’s HomePage**

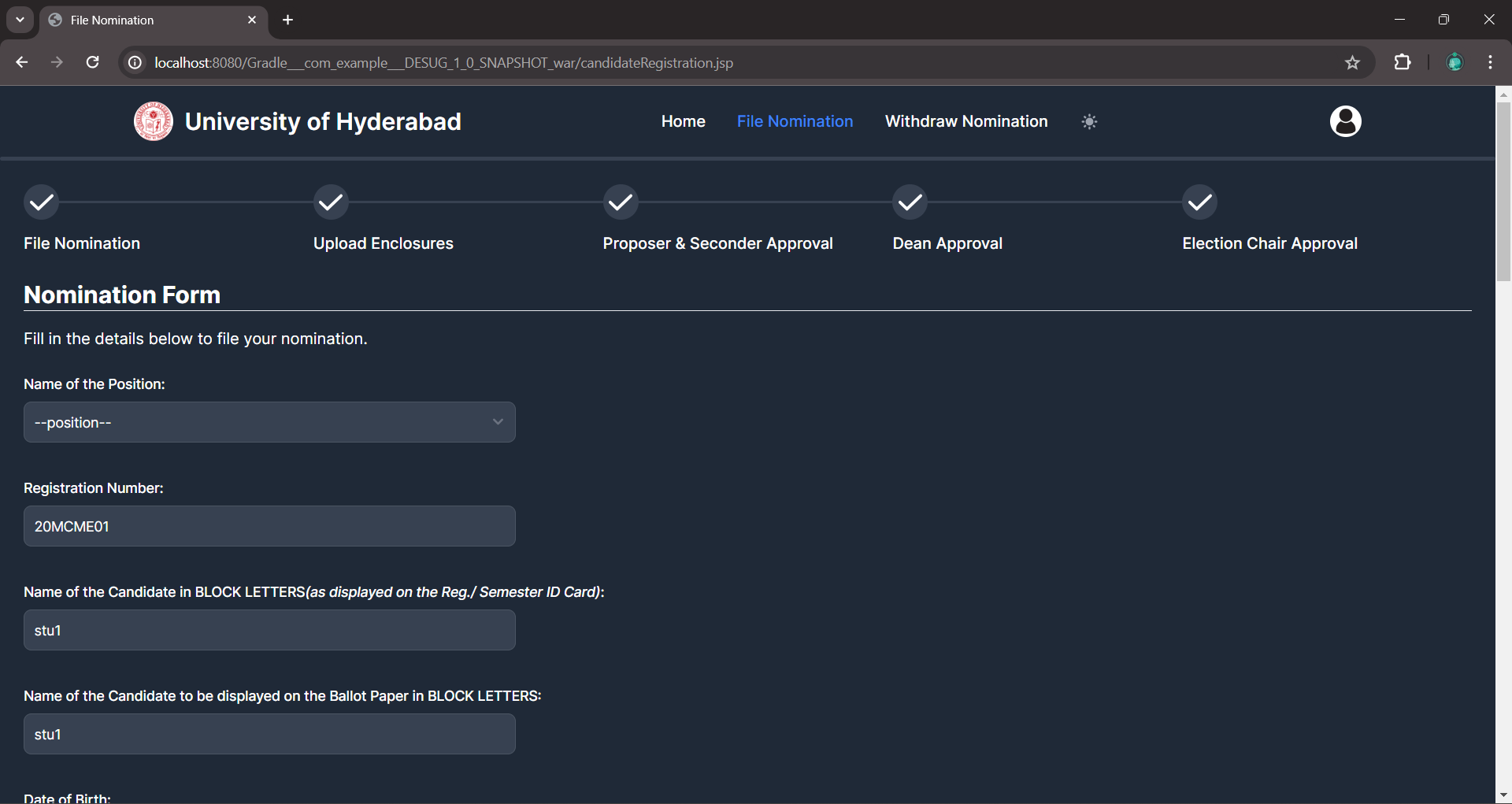


**List of Candidates to be Certified**

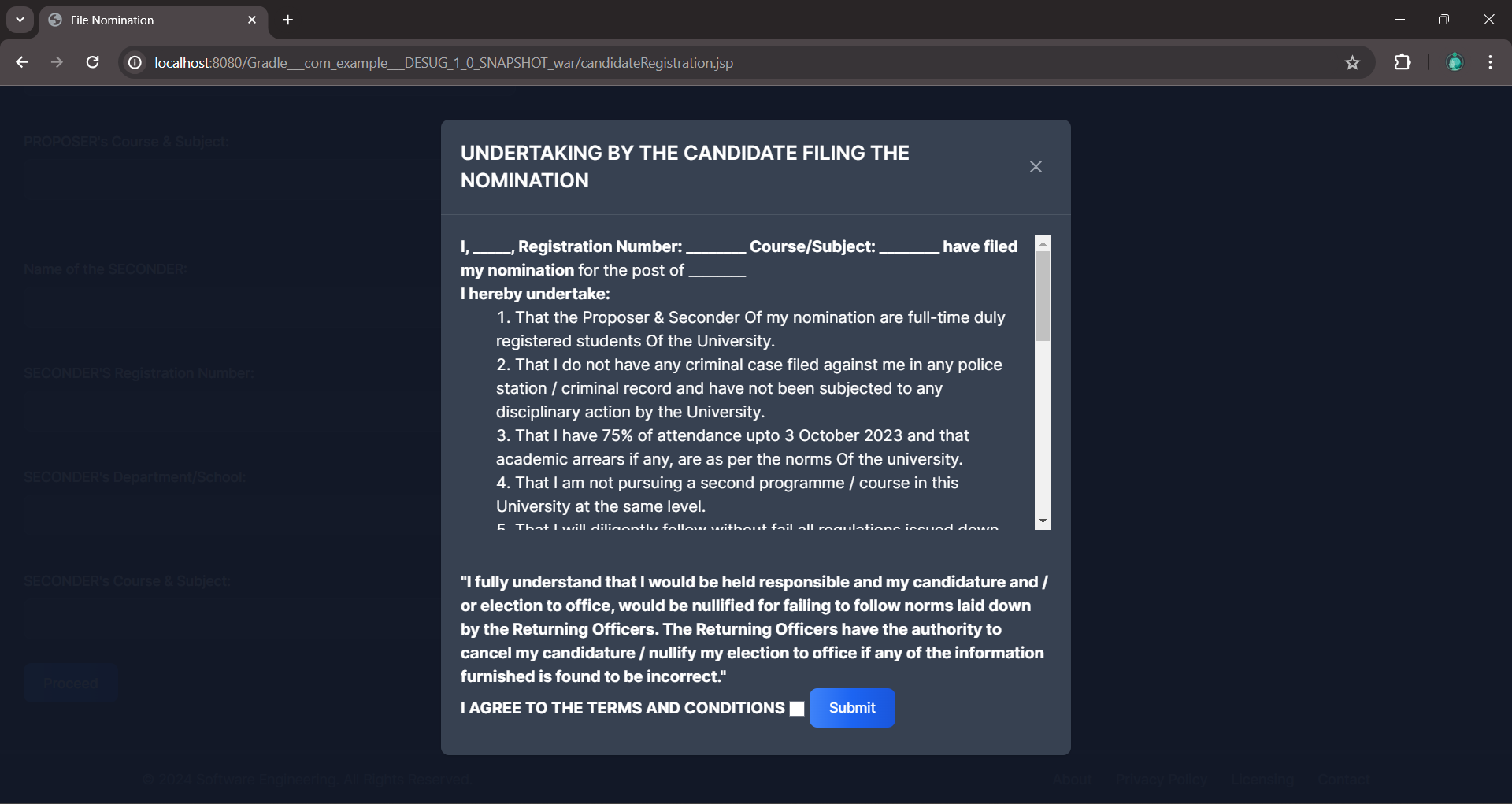


**Certify Candidate Form**

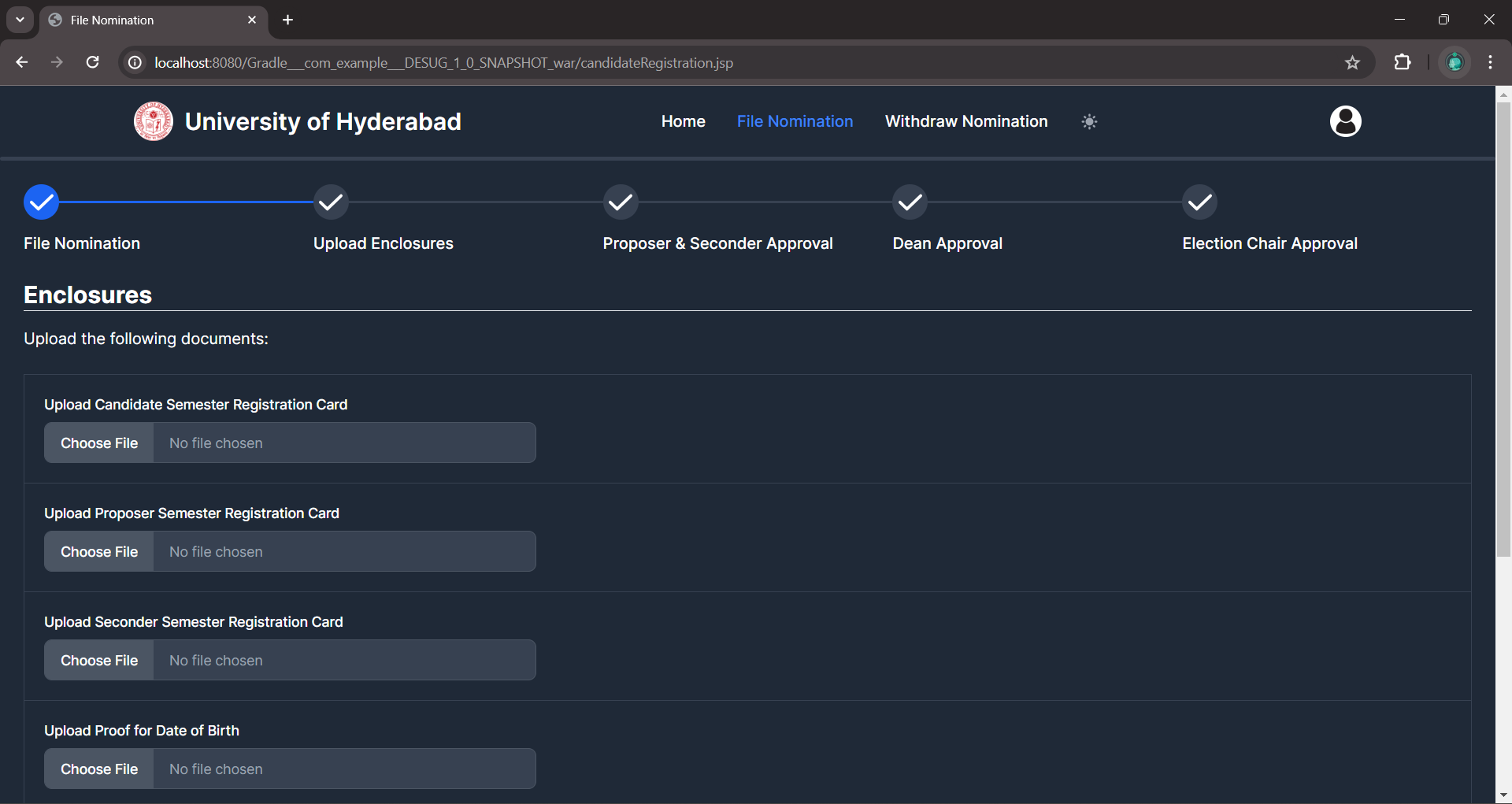
→ Student Interfaces



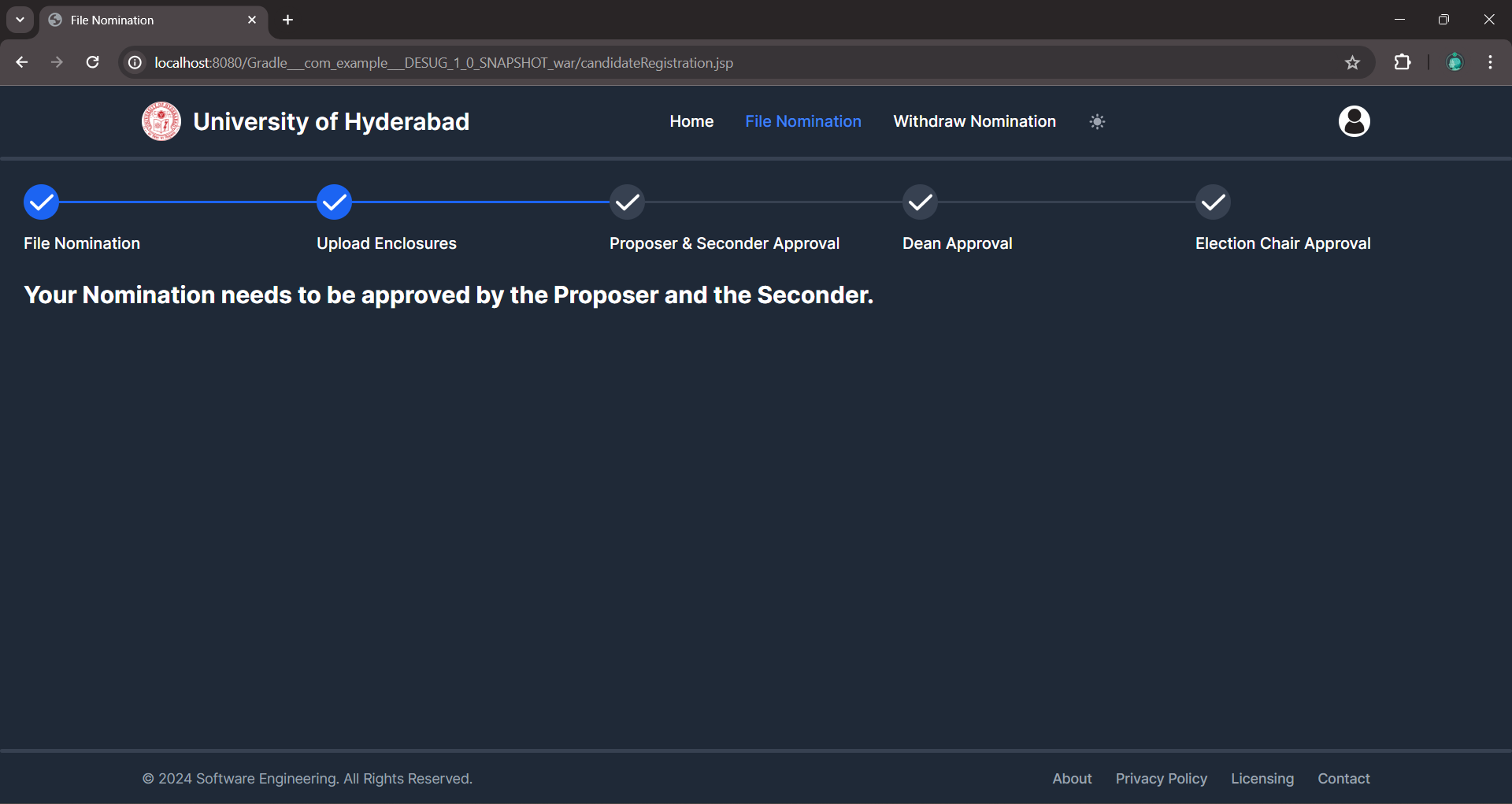
**Nomination Form**



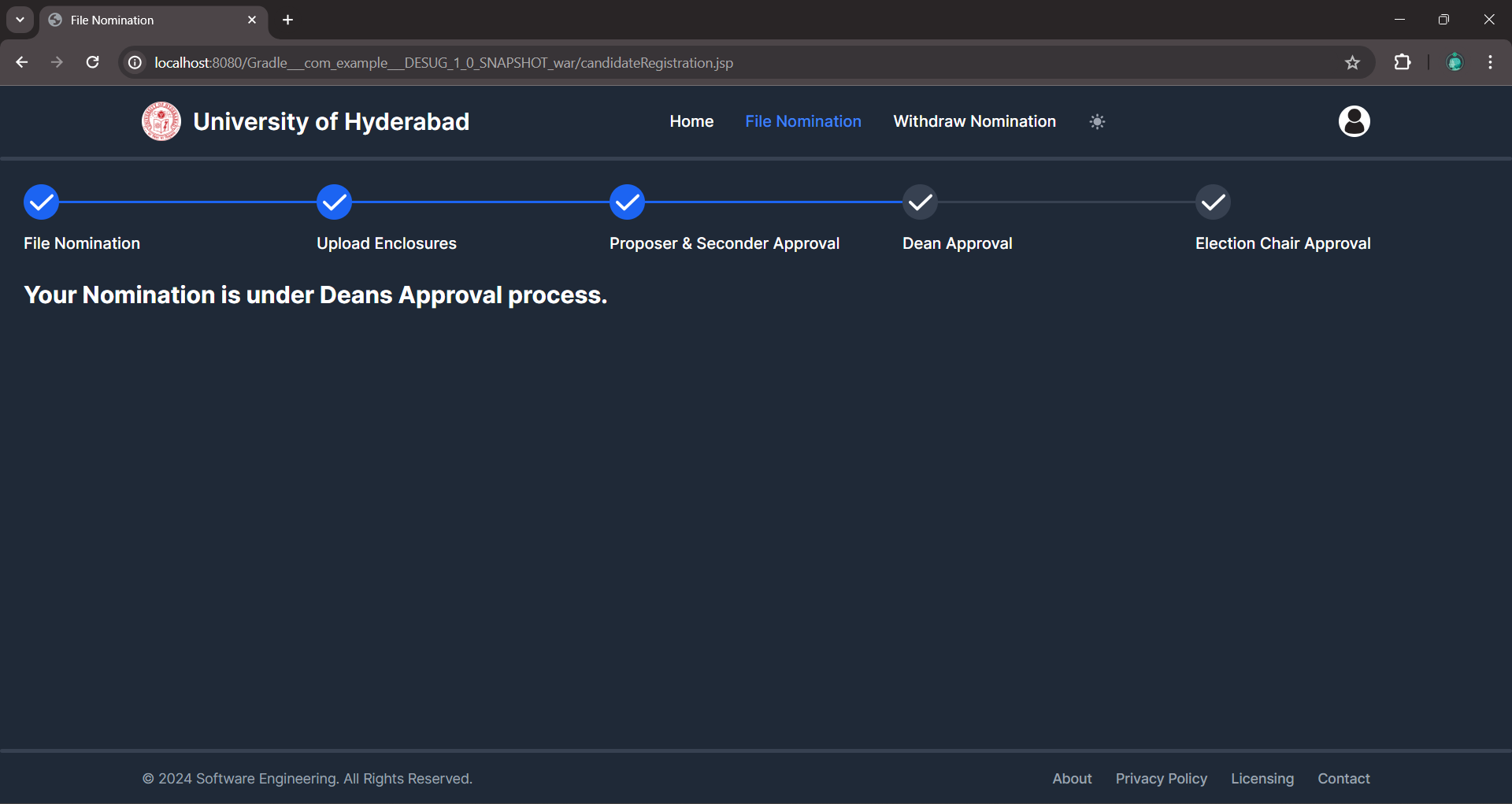
**Undertaking Form**



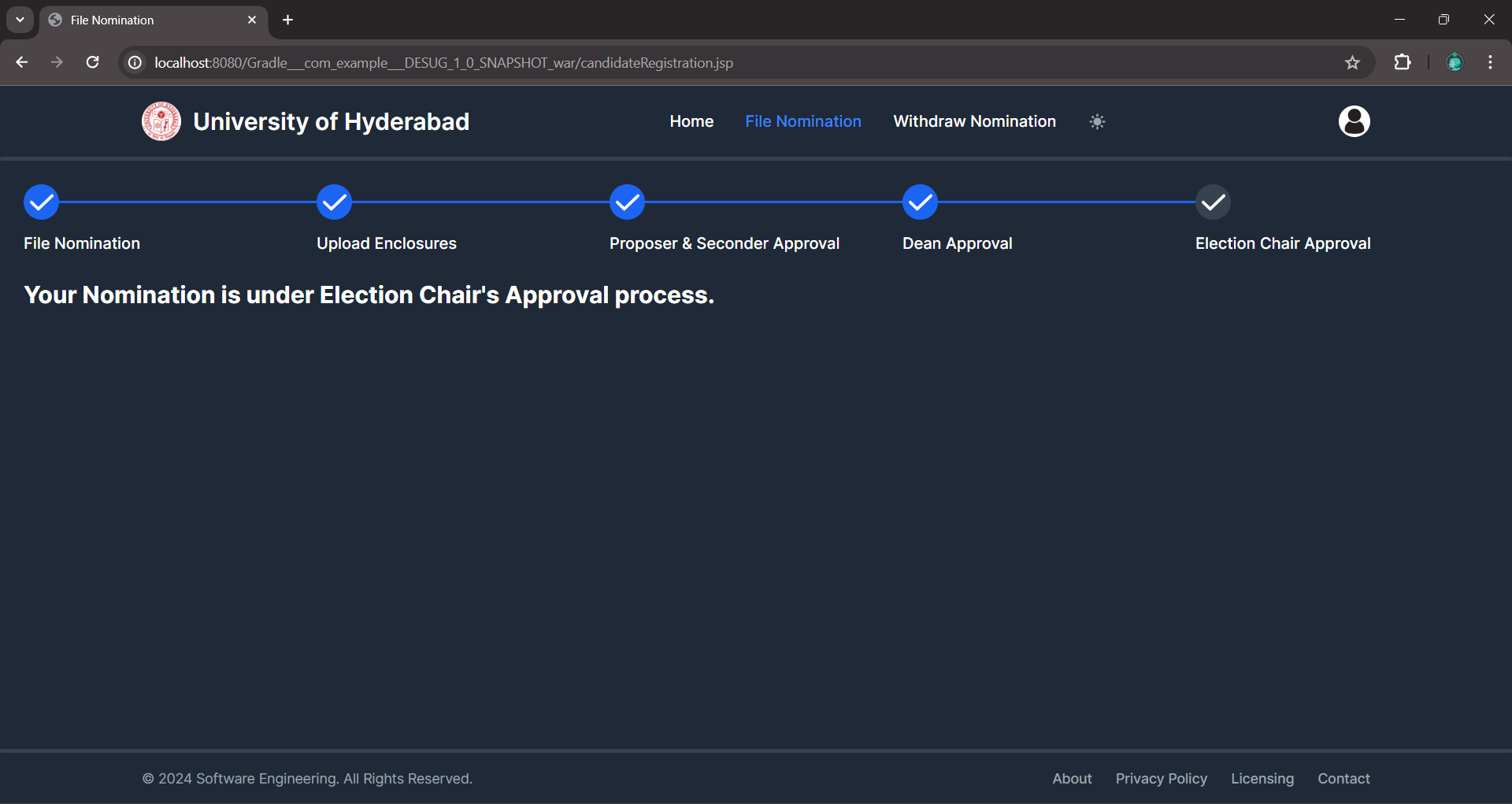
**Enclosures Form**



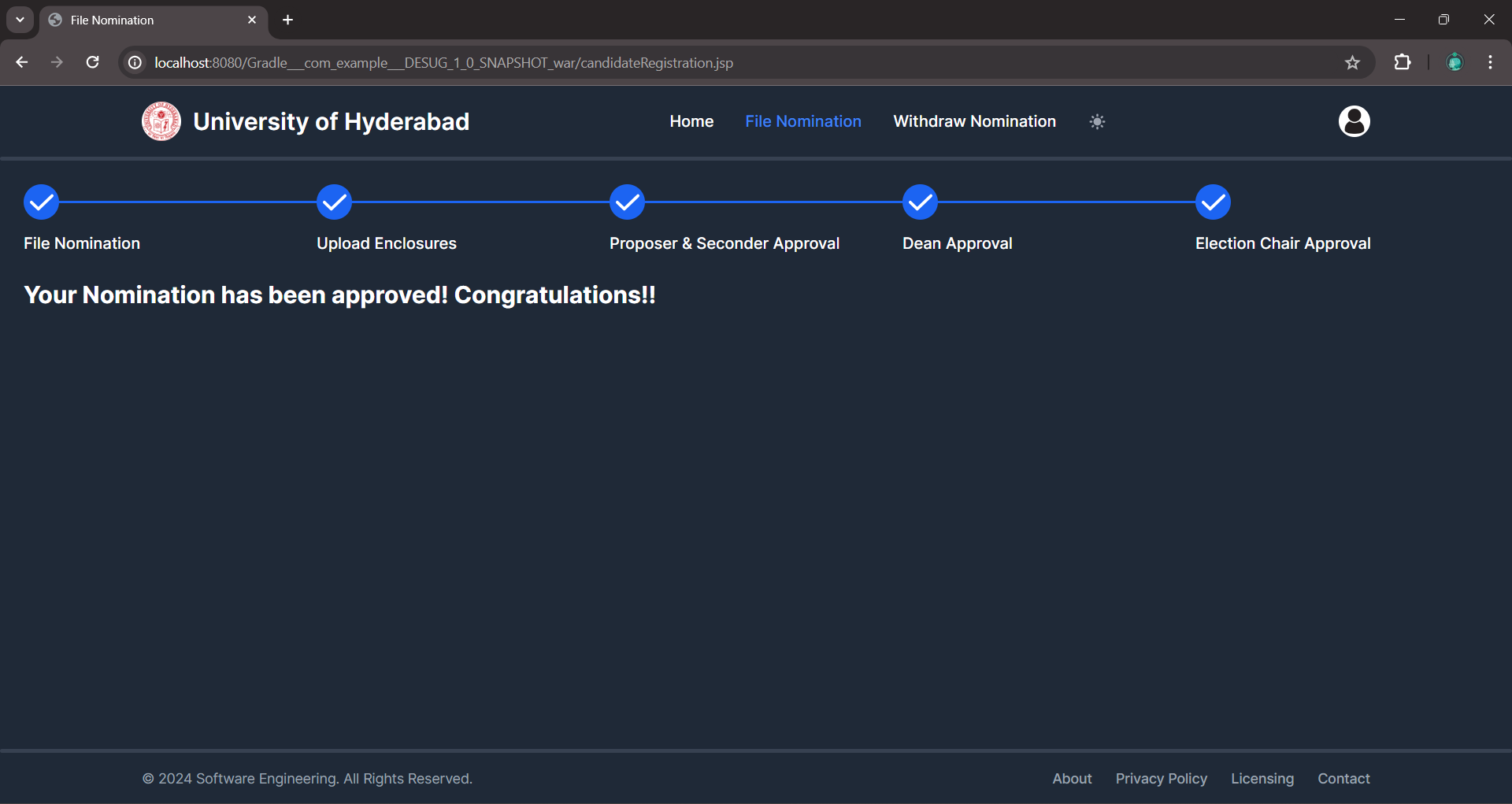
**Wait for Proposer & Seconder Approval**



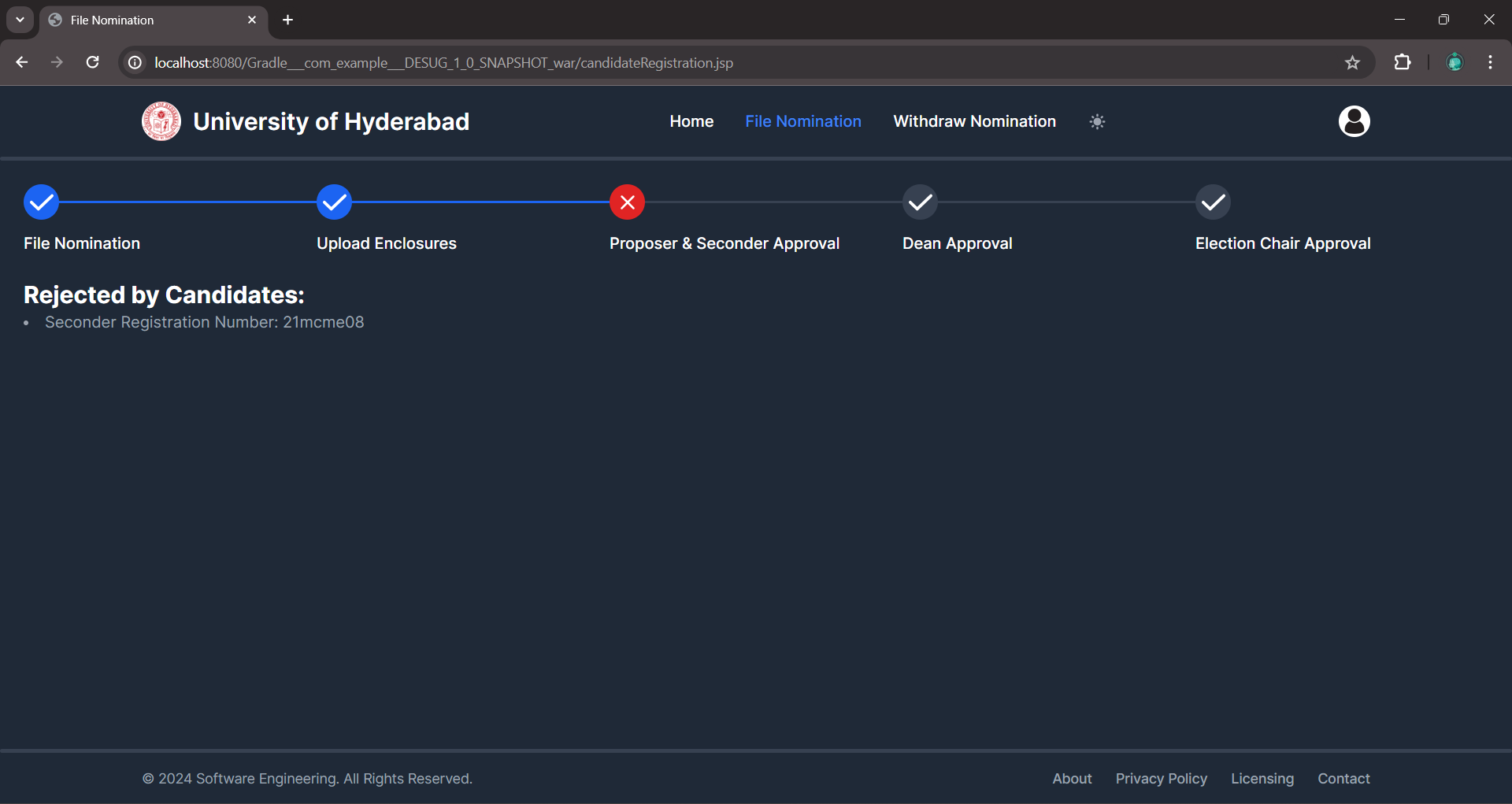
**Wait for Dean's Approval**



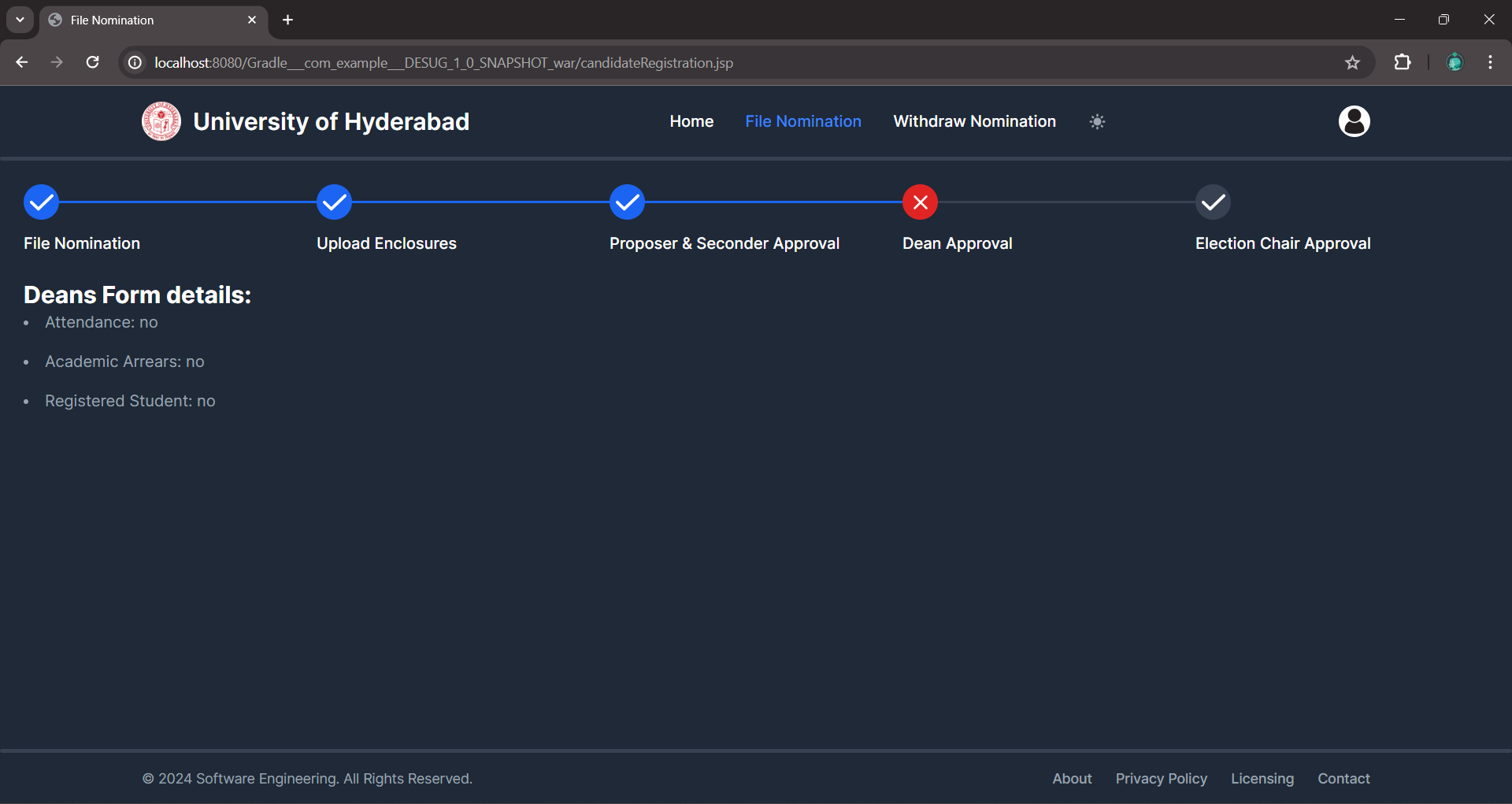
**Wait for the Election Chair’s Approval**



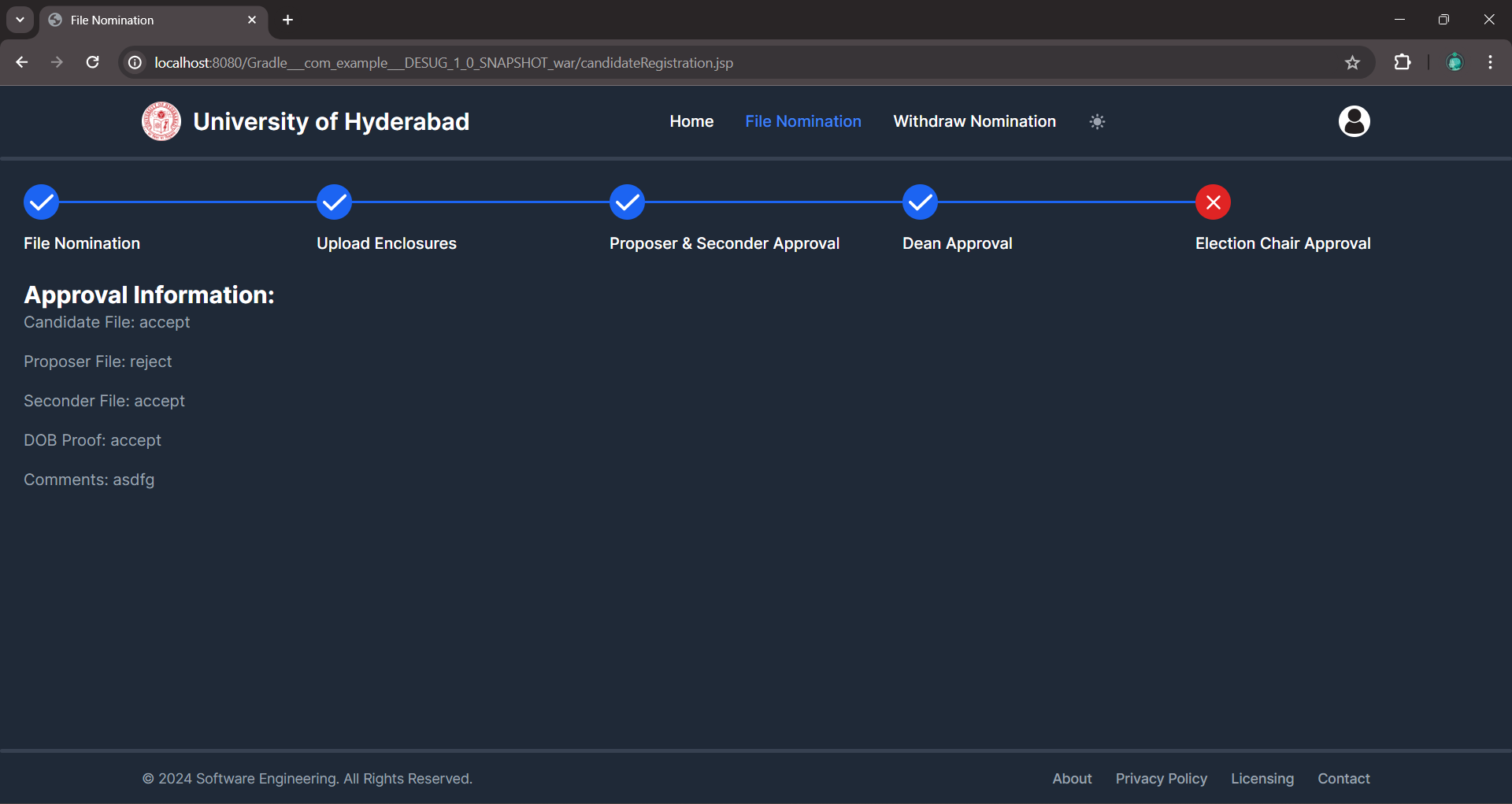
**Nomination Approved**



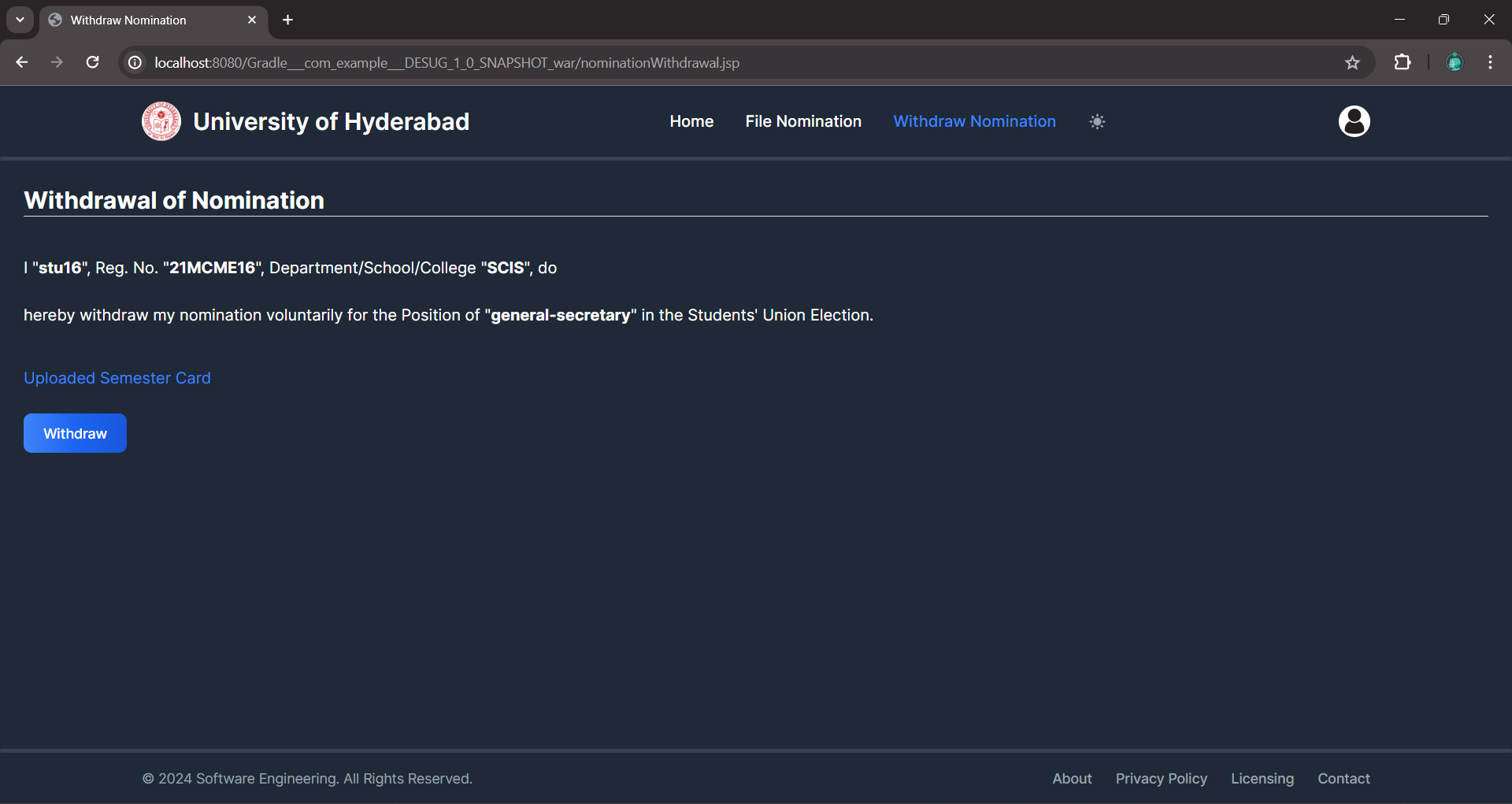
**Rejection by Proposer and/or Seconder**



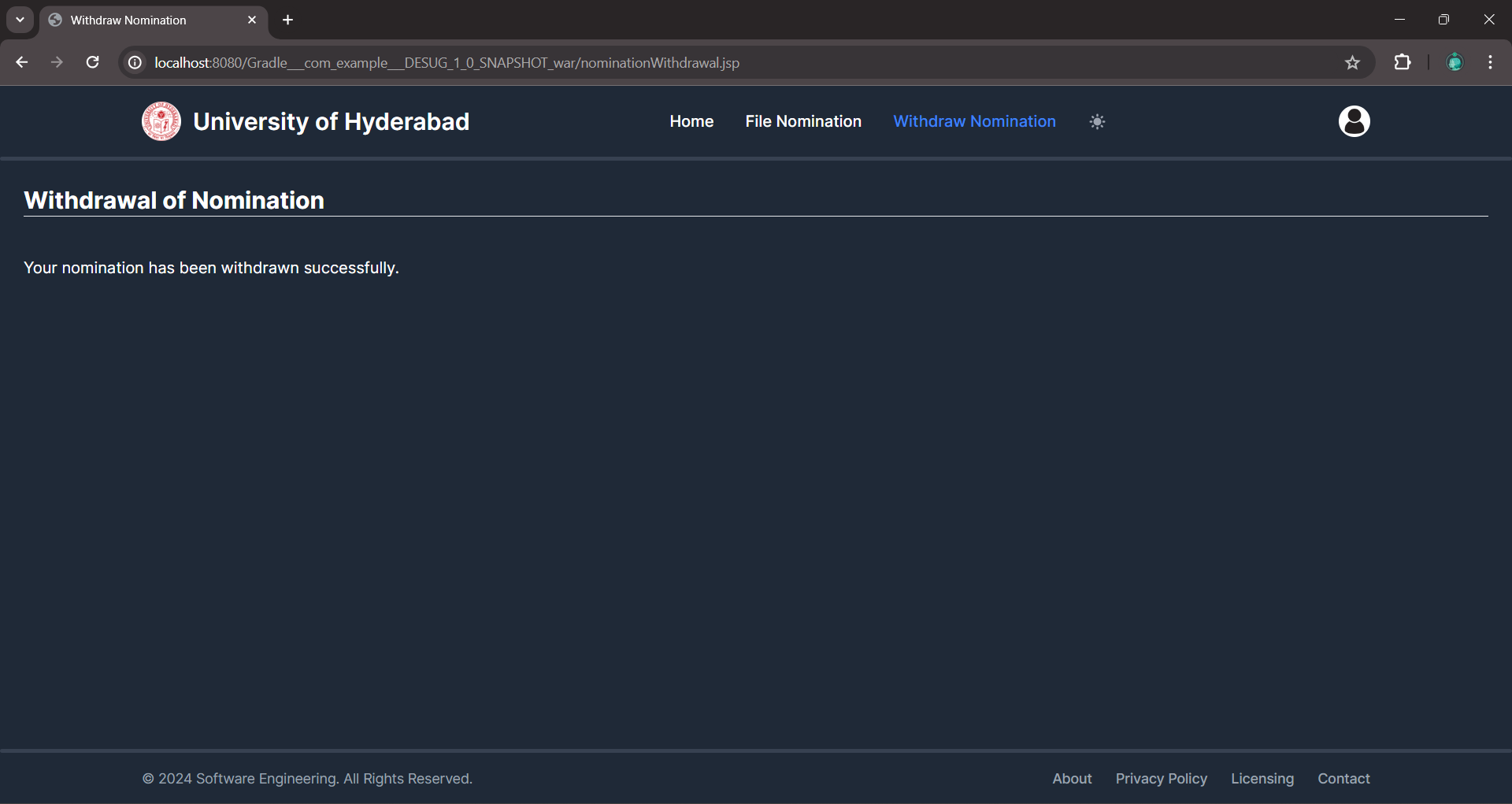
**Rejection by Dean**



**Rejection by Election Chair**

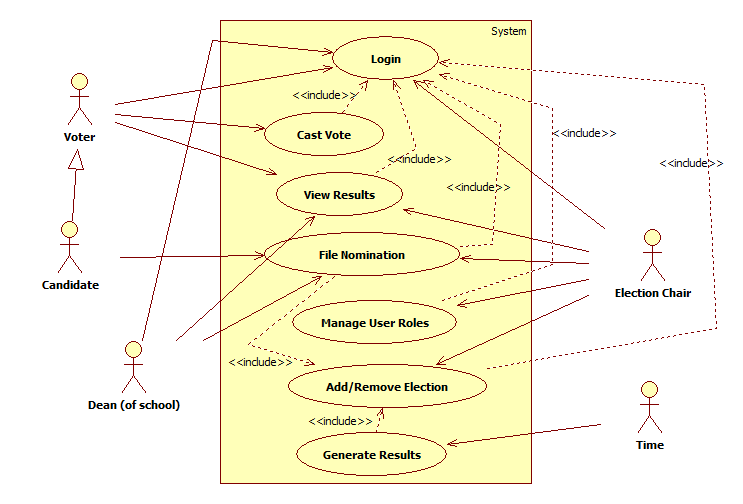


**Nomination Withdrawal Form**



**After the Withdrawal of the Nomination**

#### **Appendix-II:** Use Case Model / Use Case Diagram



#### **Appendix-III:** Use Case Specifications

* Login:

| Use Case ID: | 001 | | |
| --- | --- | --- | --- |
| Use Case Name: | Login | | |
| Created By: | TLS Srivatsa | Last Updated By: | A Sidhartha |
| Date Created: | 09-02-2024 | Date Last Updated: | 22-03-2024 |

| Actors: | Voter, Dean (of school), Election Chair |
| --- | --- |
| Description: | An actor enters their username and password to login to the system |
| Preconditions: |  |
| Postconditions: | The actor gets logged in to their account |
| **Normal Flow:** | 1. The actor enters their username in the textbox associated with the username. 2. The actor enters their password in the textbox associated with the password. 3. The actor hits the login button. [A1][A2] 4. The actor gets logged in. |
| **Alternative Flows:** | 1. The username or password is entered wrong. 2. The actor gets a message with the wrong username or password and is asked to re-enter the credentials. 3. If incorrect credentials are entered more than 3 times continuously, a warning message is sent to the registered mail. 4. The user selects the ‘Forgot Password?’ option. 5. The actor will receive a verification code in the email that he need to use to change his/her password. |
| Exceptions: |  |
| Includes: |  |
| Priority: | Medium |
| Frequency of Use: | Frequently |
| Business Rules: |  |
| Assumptions: | All the users of the system are registered with the University for the current semester. |

* Cast Vote:

| Use Case ID: | 002 | | |
| --- | --- | --- | --- |
| Use Case Name: | Cast Vote | | |
| Created By: | A Sidhartha  TLS Srivatsa | Last Updated By: | A Sidhartha  TLS Srivatsa |
| Date Created: | 09-02-2024 | Date Last Updated: | 20-02-2024 |

| Actors: | Voter |
| --- | --- |
| Description: | The user casts their vote |
| Preconditions: | Login use case has to be executed successfully |
| Postconditions: | Casted votes are recorded with the token used |
| **Normal Flow:** | 1. Generate token - token is sent to registered university email. 2. Cast your vote for the respective candidate. 3. Confirm and Next [A2] (OR) Previous. 4. Repeat steps 3 to 4 through all the candidates for which the user is eligible to vote. 5. Review your choices and change them if necessary. 6. Enter the token and Submit [A1]. |
| **Alternative Flows:** | A1. If the token is invalid or expired, redirect to the home page and show a warning message  A2. Validation of the choices |
| Exceptions: |  |
| Includes: |  |
| Priority: | High |
| Frequency of Use: | Frequently |
| Business Rules: |  |
| Assumptions: |  |

* File Nomination:

| Use Case ID: | 004 | | |
| --- | --- | --- | --- |
| Use Case Name: | File Nomination | | |
| Created By: | A Sidhartha | Last Updated By: | A Sidhartha  TLS Srivatsa |
| Date Created: | 29-02-2024 | Date Last Updated: | 08-03-2024 |

| Actors: | Candidate, Dean (of school), Election Chair |
| --- | --- |
| Description: | The candidate files his nomination |
| Preconditions: | Login use case has to be executed successfully and election has to be added successfully |
| Postconditions: | The candidate is added to the list of valid nominations |
| **Normal Flow:** | 1. The Candidate fills out the nomination form with their information along with proposer and seconder details. 2. The candidate includes his/her enclosures. 3. The candidate confirms his nomination and submits the Undertaking form. 4. The Proposer and Seconder receive notifications of their involvement in the nomination. 5. The confirm button is enabled for the Candidate after approval by the Proposer and Seconder.[A1] 6. The nomination is submitted automatically after reaching the time limit, with approval done. 7. Dean certifies the candidate with a Certificate of Attendance & Academic Record (or) approves the one enclosed by the candidate.[A2] 8. The Election Chair verifies the candidates’ details and finalizes his/her candidature.[A3][A4] |
| **Alternative Flows:** | 1. The candidate is notified if the Proposer or Seconder does not approve their consent, and is provided to update his nomination form. 2. The candidate’s Certificate of Attendance & Academic Record is forwarded to the Election Chair for approval if the Dean is not available. 3. The candidate’s nomination is rejected if any of the details or documents are found invalid. 4. The Election Chair has the authority to cancel the candidature of a candidate if found violating the rules and regulations of the election process. |
| Exceptions: | 1. Students who are enrolled for a second PG programme in the University are not eligible to contest. 2. Students who are in 3 semesters of M.Phil are also not eligible to contest. 3. Similarly, Ph.D. Scholars in their sixth year or extension are not eligible to contest. 4. Distance / proximate education students are not eligible to contest. 5. Re-admission / Recourse students are not eligible. |
| Includes: |  |
| Priority: | High |
| Frequency of Use: | Frequently |
| Business Rules: | * A student who has attained the age of 17 years and who has not completed 28 years on the date of election notification is eligible to file nomination as a candidate. * The candidate can only file a single nomination. * The candidate shall not have a previous criminal record and should not have been subjected to any disciplinary action by the Proctorial Board/Disciplinary Committee of the University. |
| Assumptions: | * The dates from the Election form will be taken into account for displaying the availability of filling forms for the Students. * Academic programmes are classified as under:  1. Post-Graduate Programmes: M.Sc, M.A, MCA, 5-year Masters (Integrated), Integrated   M.Tech.   1. Research Programmes: M. Tech, Ph.D, Integrated Ph.D. |

* Manage User Roles:

| Use Case ID: | 005 | | |
| --- | --- | --- | --- |
| Use Case Name: | Manage User Roles | | |
| Created By: | A Sidhartha  TLS Srivatsa | Last Updated By: | A Sidhartha  TLS Srivatsa |
| Date Created: | 21-03-2024 | Date Last Updated: | 21-03-2024 |

| Actors: | Election Chair |
| --- | --- |
| Description: | The Election Chair can manage the privileges of the users |
| Preconditions: | Login use case has to be executed successfully |
| Postconditions: | Privileges given are successfully reflected |
| **Normal Flow:** | 1. Select the person from the Election Chair. 2. Select the work to be delegated. 3. The person gets the privilege of performing the tasks that are assigned to him/her. 4. Notify the person regarding his task. |
| **Alternative Flows:** |  |
| Exceptions: |  |
| Includes: |  |
| Priority: | Low |
| Frequency of Use: | Rarely |
| Business Rules: |  |
| Assumptions: |  |

* Add/Remove Election:

| Use Case ID: | 006 | | |
| --- | --- | --- | --- |
| Use Case Name: | Add/Remove Elections | | |
| Created By: | TLS Srivatsa  A Sidhartha | Last Updated By: | A Sidhartha  TLS Srivatsa |
| Date Created: | 21-03-2024 | Date Last Updated: | 21-03-2024 |

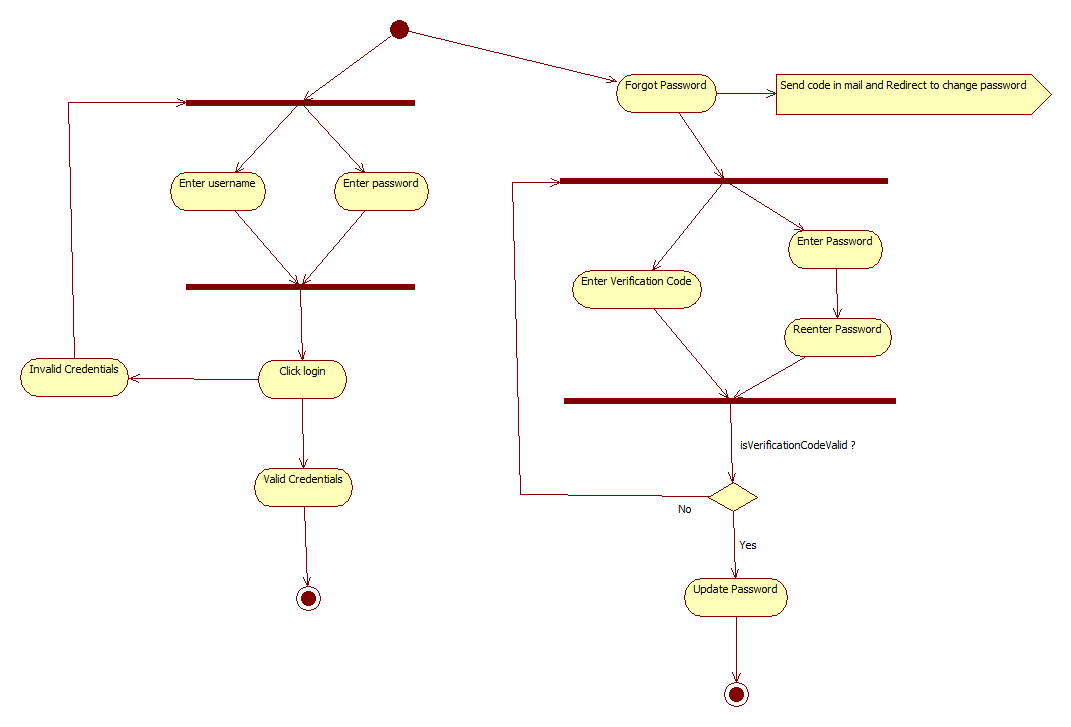
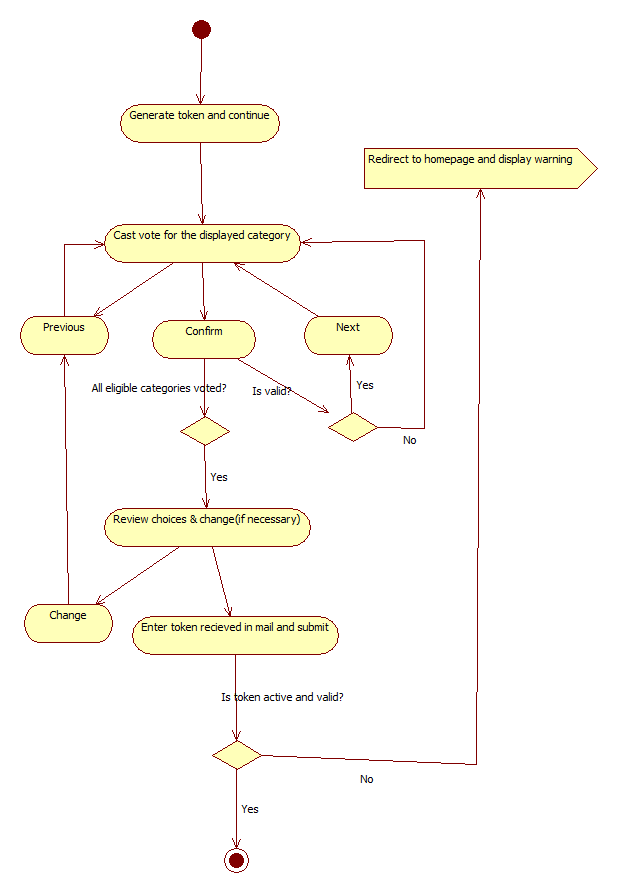
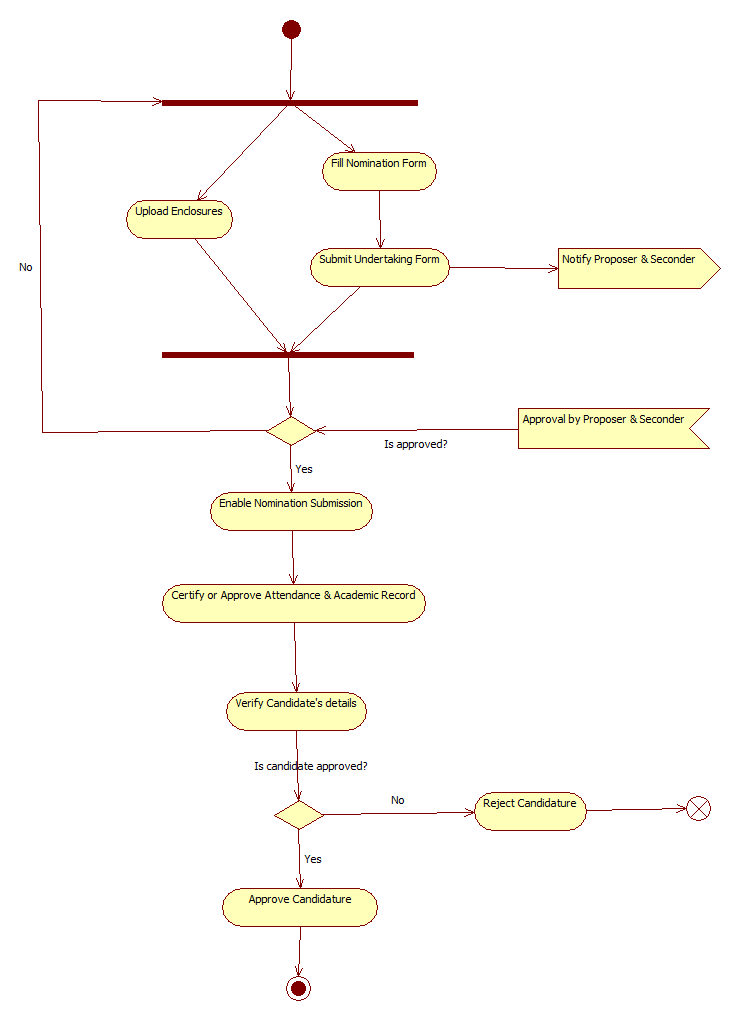
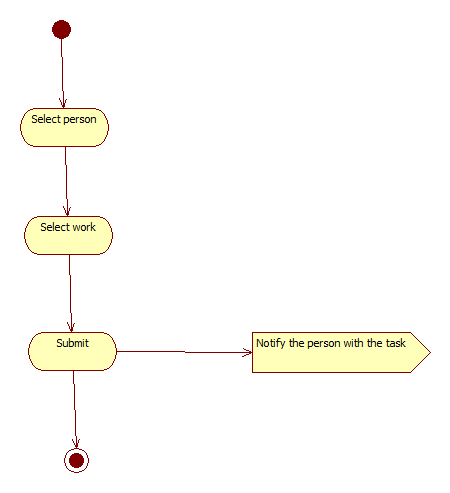
| Actors: | Election Chair |
| --- | --- |
| Description: | The Election Chair adds the Elections |
| Preconditions: | Login use case has to be executed successfully |
| Postconditions: | Election gets successfully added |
| **Normal Flow:** | 1. After logging into the system as Election Chair, adding/removing an election is provided on the home page. 2. The Actor, after the ‘add election’ is selected, will be redirected to a form page, where he fills out the election form. 3. In this form, the positions and number of positions with the scope of the electoral should be added and selected in the form. 4. The dates for the election schedule need to be filled for various events such as Filing of nominations, Withdrawal of nominations, and Submission of polling & counting Agents. 5. An option to remove will be given next to the added Election in the Election List on the home page which removes the Election and respective forms that have been registered against the election. |
| **Alternative Flows:** |  |
| Exceptions: |  |
| Includes: |  |
| Priority: | High |
| Frequency of Use: | Occasionally |
| Business Rules: | * A student who has attained the age of 17 years and who has not completed 28 years on the date of election notification is eligible to file nomination as a candidate. * The candidate can only file a single nomination. * The candidate shall not have a previous criminal record and should not have been subjected to any disciplinary action by the Proctorial Board/Disciplinary Committee of the University. |
| Assumptions: | * Academic programmes are classified as under:  1. Post-Graduate Programmes: M.Sc, M.A, MCA, 5-year Masters (Integrated), Integrated   M.Tech.   1. Research Programmes: M. Tech, Ph.D, Integrated Ph.D. |

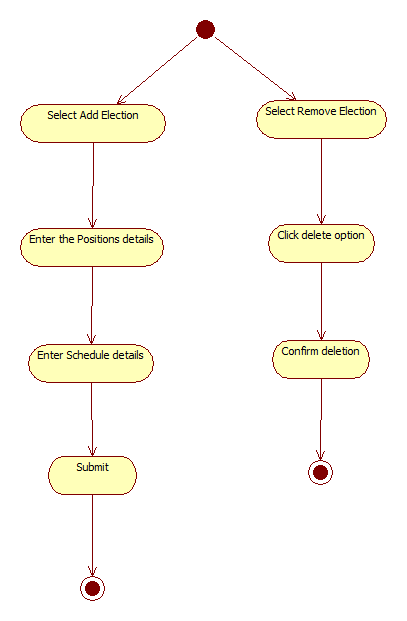
* Generate Results:

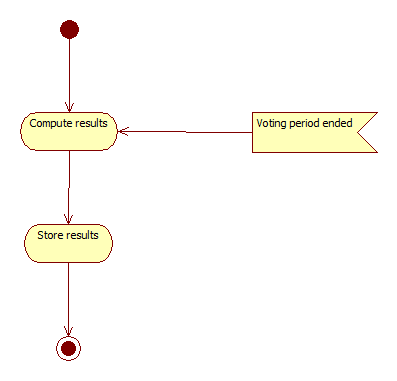
| Use Case ID: | 007 | | |
| --- | --- | --- | --- |
| Use Case Name: | Generate Results | | |
| Created By: | A Sidhartha | Last Updated By: | A Sidhartha  TLS Srivatsa |
| Date Created: | 21-03-2024 | Date Last Updated: | 21-03-2024 |

| Actors: | Time |
| --- | --- |
| Description: | Generate the results of the election after the Voting period ends |
| Preconditions: | Login and Add/Remove election use cases have to be executed successfully |
| Postconditions: | Results of the election are generated successfully |
| **Normal Flow:** | 1. After the voting time is completed, results are automatically computed by the system. 2. Users can view the generated results by navigating to the respective page. |
| **Alternative Flows:** |  |
| Exceptions: |  |
| Includes: |  |
| Priority: | High |
| Frequency of Use: | Occasionally |
| Business Rules: |  |
| Assumptions: |  |

#### **Appendix-IV:** Activity Diagrams

* Login:
* Cast Vote:
* File Nomination:
* Manage User Roles:
* Add/Remove Election:



* Generate Results:  
  

#### **Appendix-V:** Analysis Classes

(according per the design)

* HomePage
* LoginServlet
* Login
* CandidateRegistrationPage
* SubmitNominationServlet
* Nomination
* DeansHomePage
* ElectionChairsHomePage
* DelegateWorkPage
* DelegateWorkServlet
* ManageElectionPage
* ManageElectionServlet
* CastVotePage
* CastVoteServlet
* GenerateResultServlet