***By – Himanshu / Vanessa / Eric***

***Project Report: Employee Management System***

***1. Project Scope***

***- Users: - Administrators - Employees***

***- Beneficiaries: - Streamlining leave management - Providing detailed payroll information***

***- Potential Clients: Companies seeking an employee management system with leave and payroll functionalities.2. Project Functionalities***

***- Authentication: - Secure login system using POST requests. - Differentiation between administrators and employees.***

***- Role-Based Access: - Admin and employee logins trigger different server endpoints.***

***- Features Window: - Navigation to modules (profile, registration, payroll, clock management, timesheets, leave tracking).***

***- Server Communication: - Utilizing HttpClient for secure server communication. - JSON serialization and deserialization.***

***- Error Handling: - Messages for invalid logins and database connection issues. - User-friendly feedback.***

***- Post-Midterm Updates: - Enhanced UI for improved experience. - Improved security. - Additional features based on user feedback.***

***- Leave Page Functionality: - Calculation of leave duration. - Retrieval and submission of leave requests.***

***- Payroll Page Functionality: - Calculation and display of payroll information. - Use of a separate class (Payroll Class). - Printing payroll information.***

***- Clock Page Functionality: - Clock submission for Punch In, Meal Out, Meal In, and Punch Out. - User feedback through Message Box prompts. - Display of recent time entries in a DataGrid.***

***- Profile page Functionality Profile Retrieval: Uses asynchronous HttpClient for server communication to fetch user profile information based on the user ID.***

***User Type Differentiation: Distinguishes between administrators (starting with "A") and employees (starting with "E") for targeted data retrieval.***

***UI Update: Dynamically updates UI elements with retrieved user profile details.***

***- Registration page Functionality***

User ***Operations: Handles user creation, retrieval, update, and deletion through server communication.***

***HTTP Requests: Utilizes HttpClient for sending POST, GET, PUT, and DELETE requests to the server.***

***Input Validation: Validates input fields before processing requests, ensuring completeness.***

***Feedback: Provides basic user feedback through MessageBox prompts.***

***Enhancements: Consider refining error handling and optimizing code segments for better maintainability.***

***-Time Sheet page Functionality- User Identification: - Retrieves and displays user information based on their role (Administrator or Employee).***

***- TimeSheet Retrieval: - Fetches and populates the time entries grid based on selected start and end dates.***

***- Total Hours Calculation: - Computes and displays the total hours worked during the specified period.***

***- Print Functionality: - Implements a simple print functionality for the displayed content.***

***- Input Validation: - Consider validating user input for date selection to enhance robustness.***

***- Code Modularity: - Evaluate opportunities for modularizing code segments for better maintainability.***

***- Exception Handling: - Implement error handling mechanisms for potential network or API-related issues.***

**Note*: in the Deliverable 1 there were certain portals we could not finish on time and can be implemented in future versions. Employee performance portal, employee benefits portal, management hierarchy, career opportunities.***

***3. We have used SQL version 19 and,***

***We have implemented all the portals to work through RestAPI for improved communication between machines***

***- Controller: - Handles user authentication, user management, timesheets, and leave requests. - Dependencies and Constructor: Depends on IConfiguration for accessing configuration settings. - Admin and Employee Login: AdminLogin and EmpLogin methods handle login for administrators and employees.***

***- Administrator Table: - CRUD methods for the Administrator table***

***- Employee Table: - CRUD methods for the Employee table***

***- TimeSheet Table: - CRUD methods for the TimeSheet table***

***- Leave/MaxLeave Table: - Methods for leave requests and MaxLeave table***

***- Security and Validation: - Implement security measures for authentication and authorization. - Use HTTPS for encrypted communication. - Apply input validation to handle invalid or malicious inputs.***

***- Logging and Error Handling: - Integrate logging for API requests, errors, and critical events. - Implement comprehensive error handling for database interactions and API responses.***

***- Consistency and Documentation: - Ensure consistent naming conventions and coding styles. - Add comments for complex logic or non-trivial sections. - Consider Swagger/OpenAPI for interactive API documentation.***

***- Testing: - Implement unit tests to verify method correctness and API reliability.***

***- Middleware: - Consider implementing middleware for tasks like exception handling and request/response logging.4. NUnit Test Report Highlights for Payroll Class***

***- Tested Class: - Payroll Class in the E Figuro WPF Models namespace.***

***- Setup Method: - Setup initializes Payroll Class with parameters.***

***- Assertions: - Used N Unit assertions for value and type validation.***

***- Results: - All test methods pass, confirming method accuracy.***

***- Recommendations: - Enhance coverage with edge and failure scenario tests. - Integrate tests into continuous integration processes.***

***- Next Steps: - Maintain and update the test suite as code evolves. - Collaborate with developers to address identified issues.5. Project Setup***

***- Requirements: - Utilizes HttpClient for server communication. - Requires Visual Studio for development.***

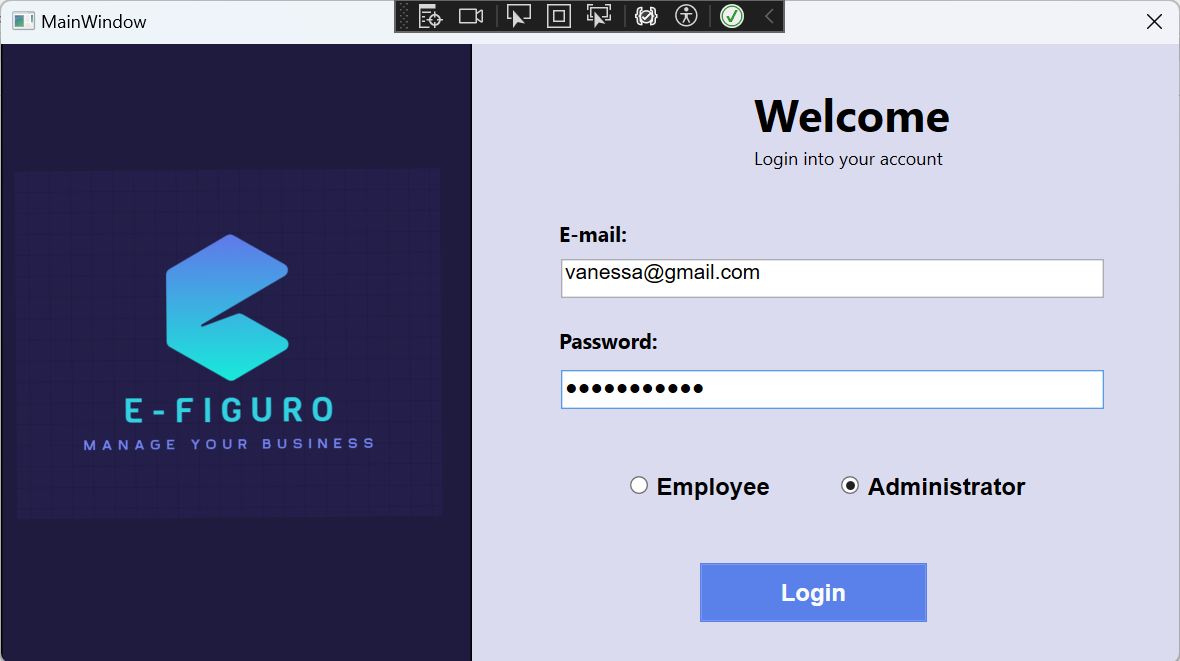
***- Installation Guide: - Step-by-step instructions for local setup. - Configuration details for the database and server URLs. - Sections requiring local customization.6. Work Progress in Future***

***- Future Functionalities: - Integration with additional modules (e.g., employee performance tracking). - Mobile application development. - Improved analytics for better decision-making.***

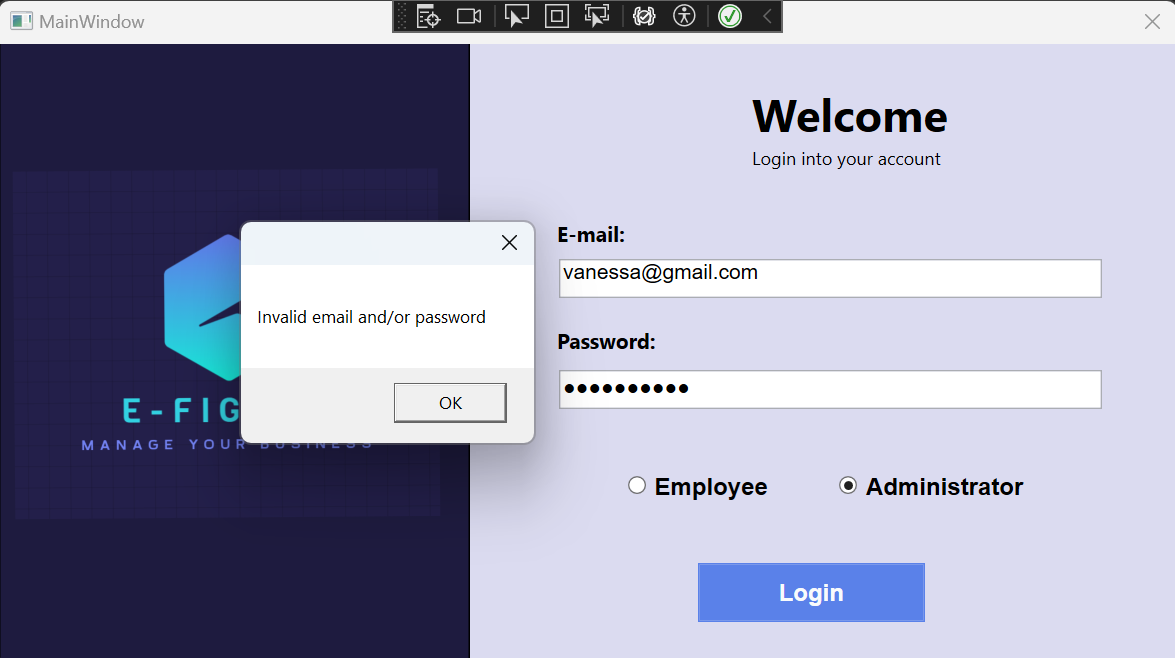
***- Upcoming Enhancements: - Continuous UI improvement. - Exploration of AI-driven features. - Collaboration with other companies for knowledge sharing.7. Conclusion - The Employee Management System project aims to provide a comprehensive solution for companies to manage their workforce efficiently. With a focus on user-friendly interfaces, secure authentication, and essential functionalities like leave tracking and payroll management, the system addresses the diverse needs of administrators and employees.8. Future Directions - The project is set to evolve with continuous improvements in UI, security enhancements, and the integration of additional modules to meet the dynamic demands of modern workforce management. The exploration of AI-driven features and collaboration with other companies will contribute to the system's growth and adaptability in the ever-changing landscape of employee management solutions.***

***Program test run + exception handling with Screenshots.***

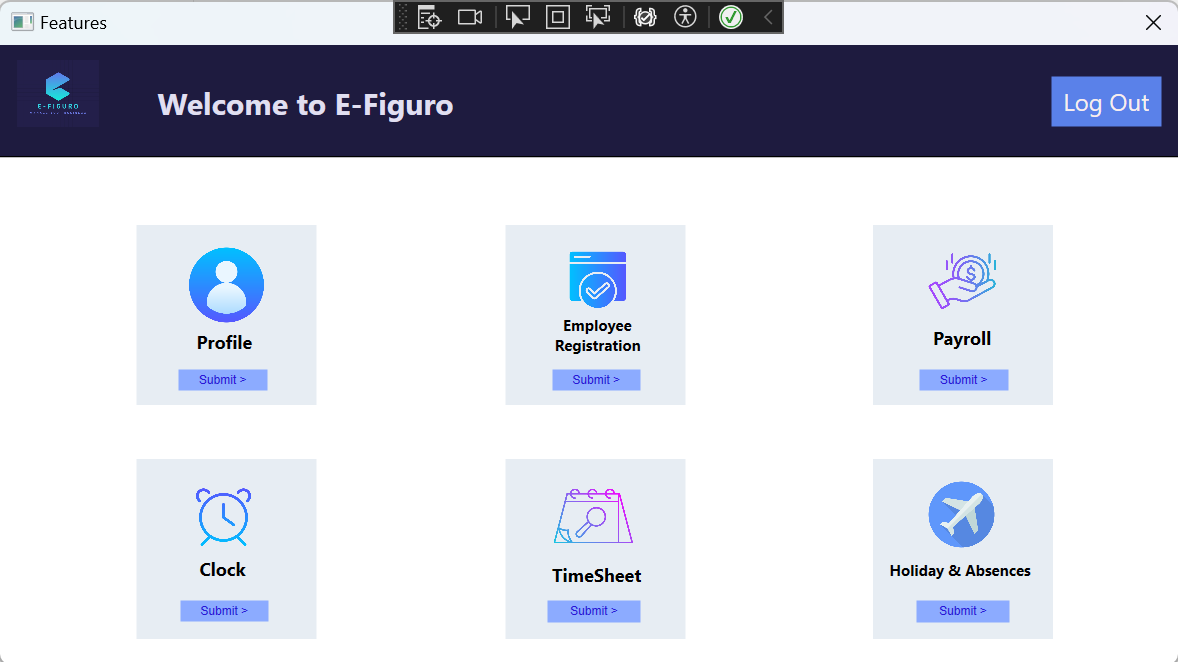
Login page



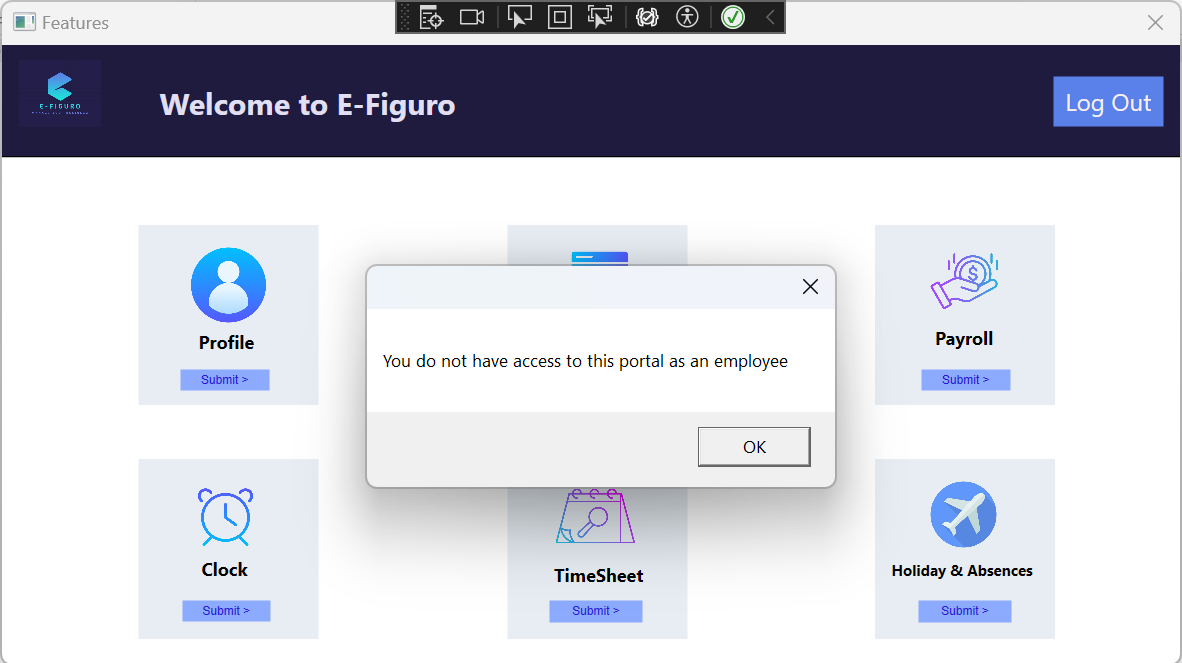
Error message if incorrect email or password is entered or wrong type of account is selected



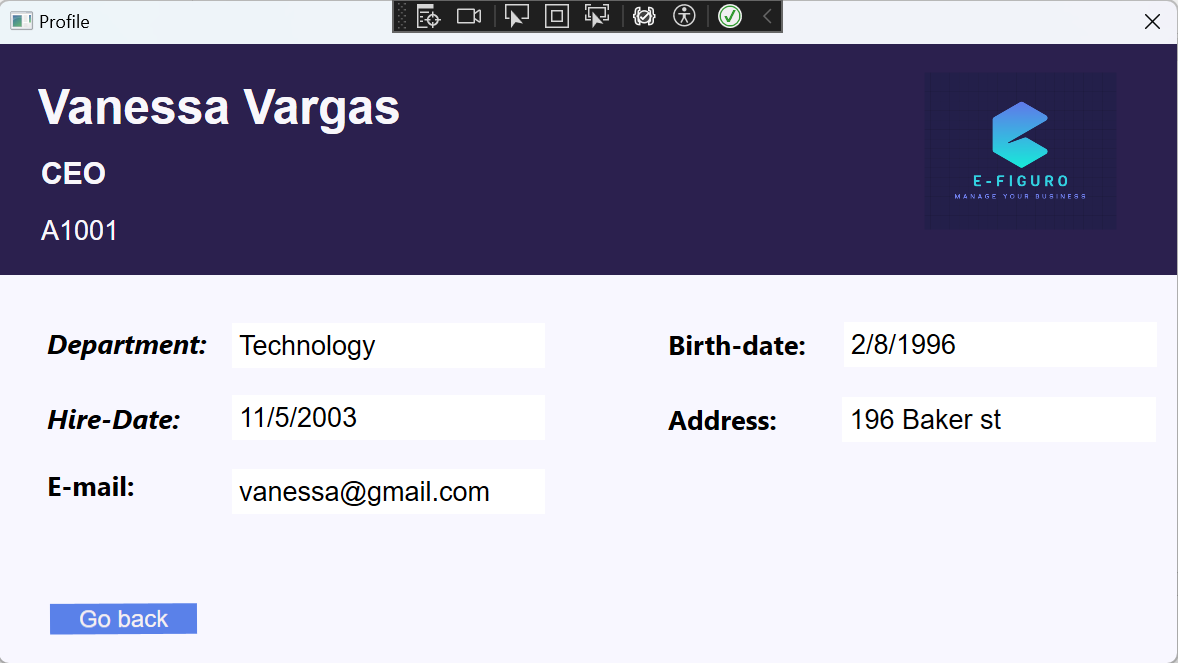
Features page – after successful login, the userID is saved from this point on until the user logs out



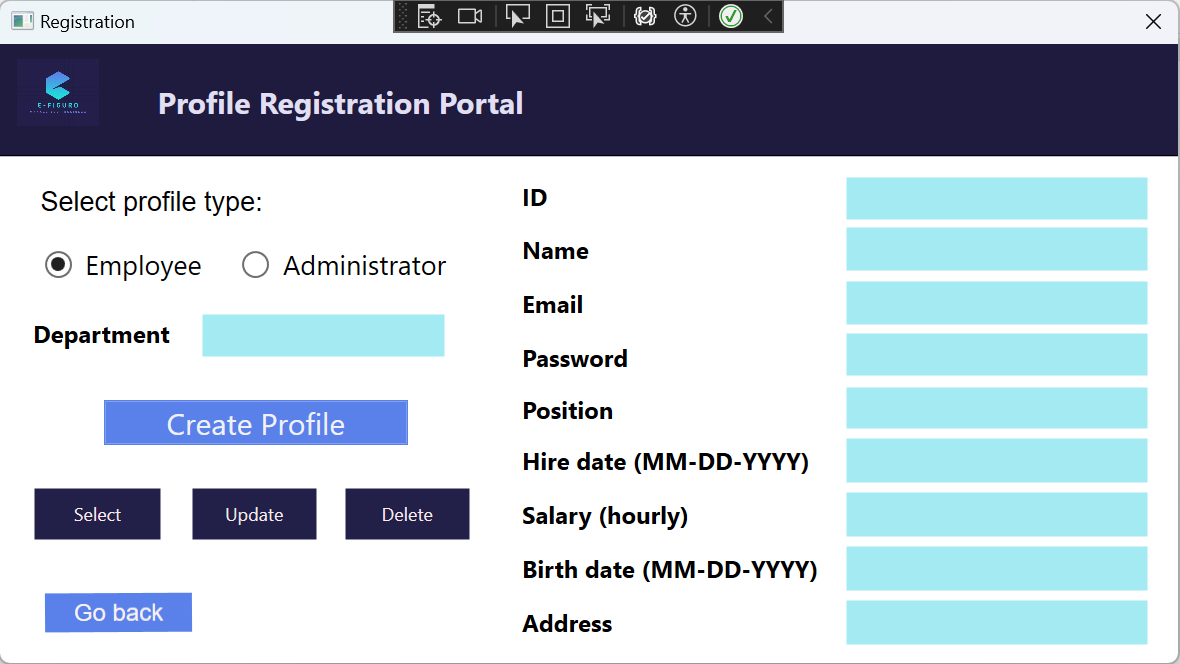
Error message if an Employee user tries to access “Employee Registration” (only administrators can create new users)



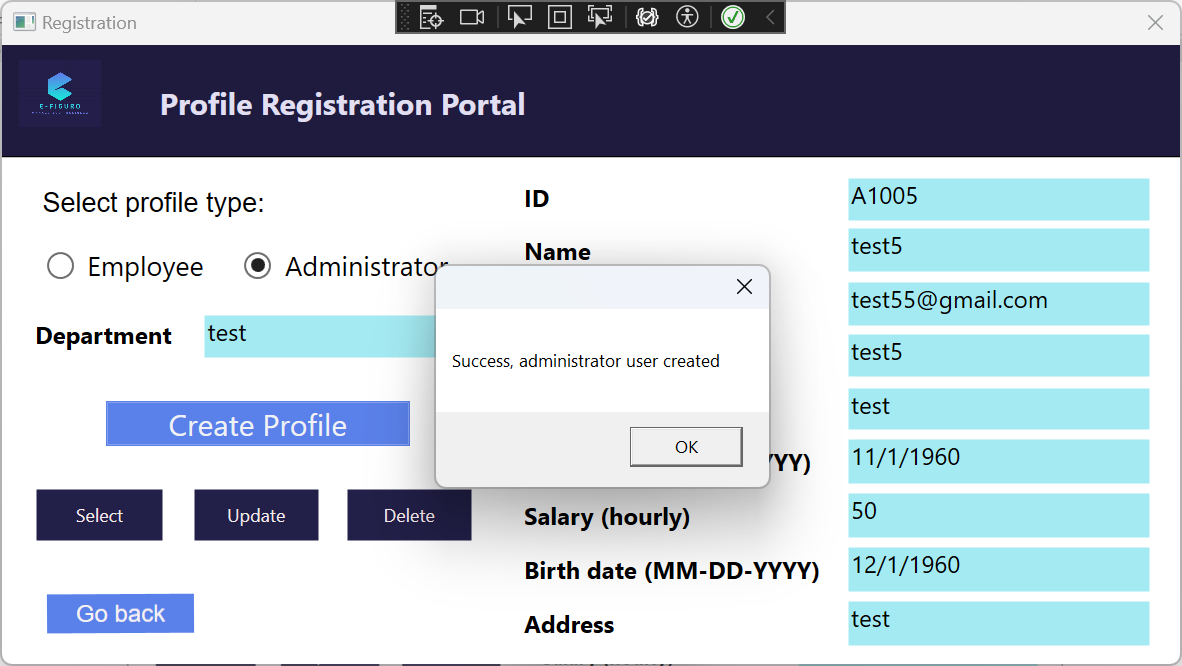
Profile portal – information is automatically pulled from database when window is opened (obtained through the userID that is saved from window to window)



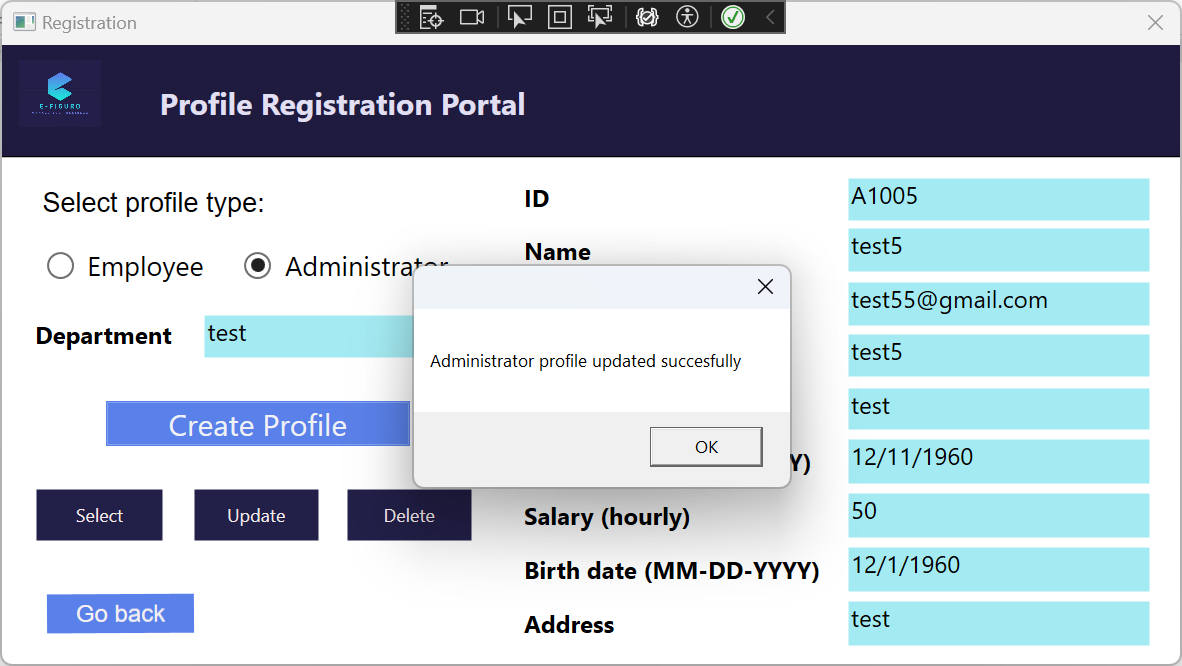
Registration portal



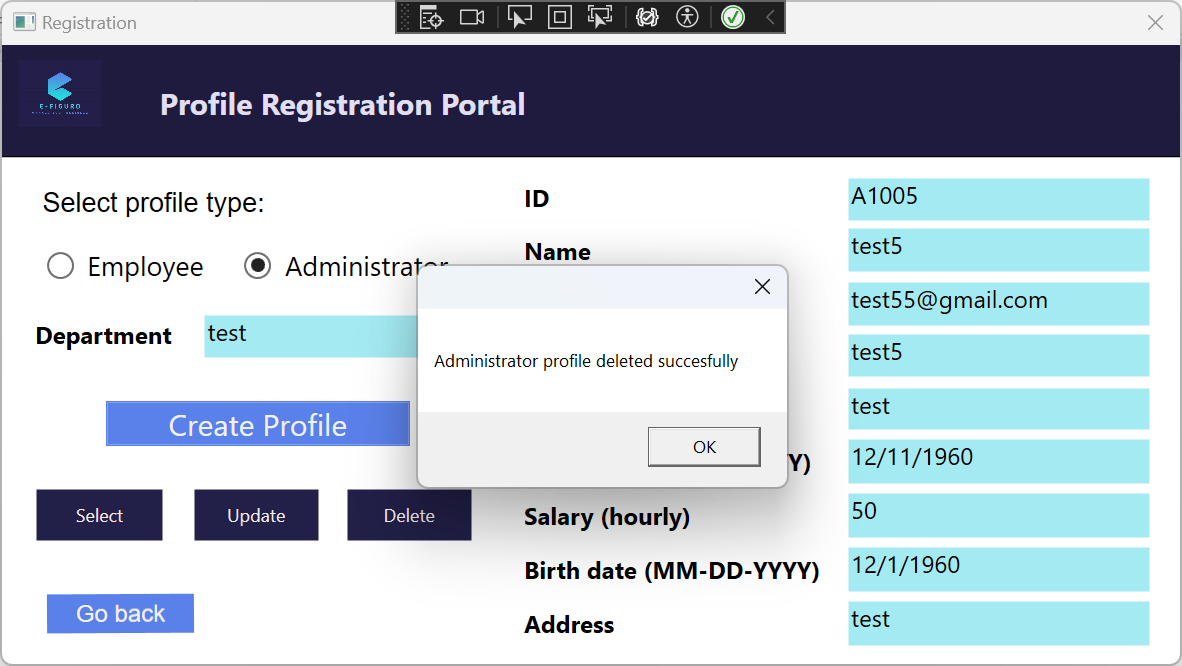
Success message if user clicks “Create Profile” and everything is okay



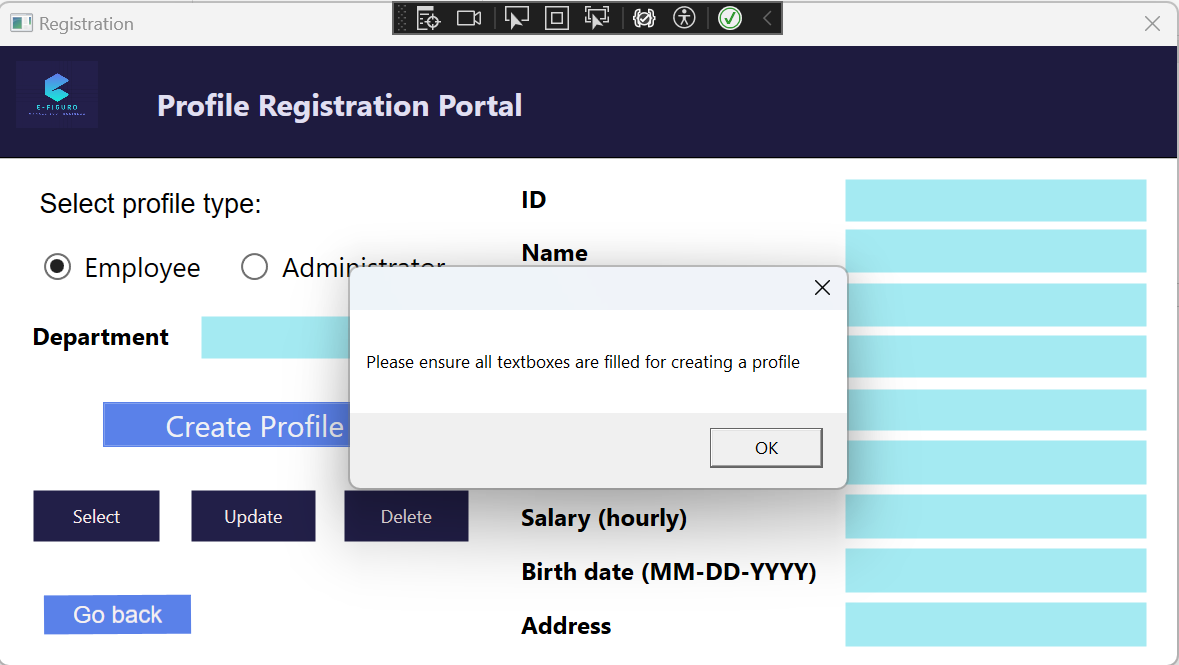
Success message if user clicks “Update” and everything is okay



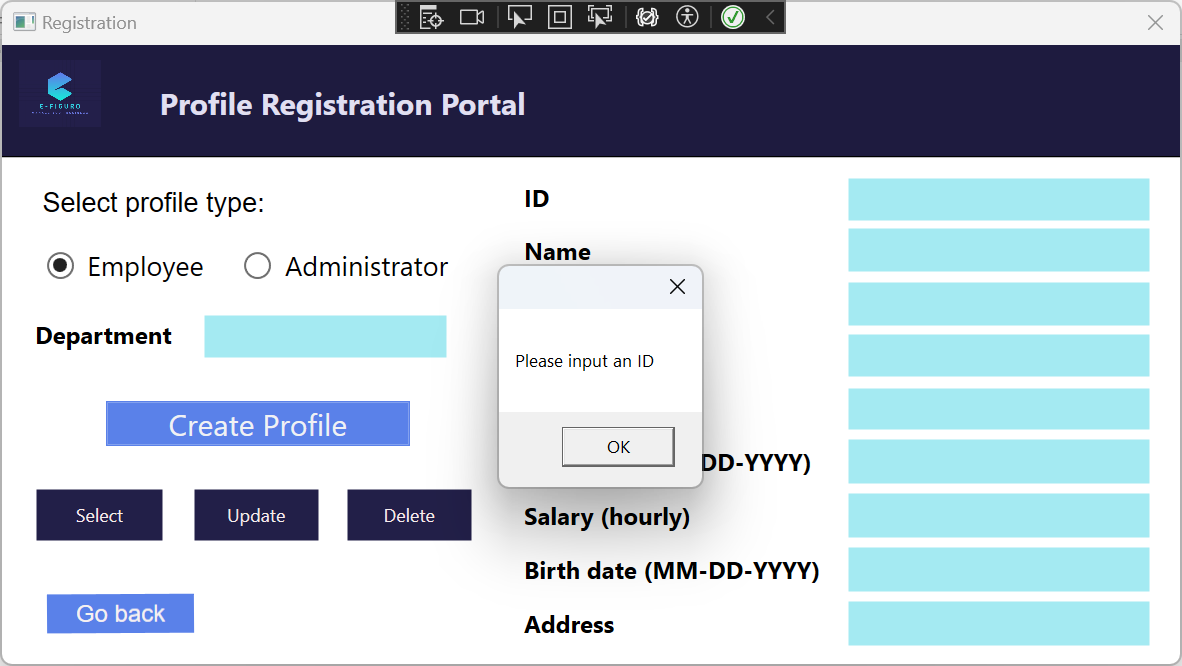
Success message if user clicks “Delete” and everything is okay



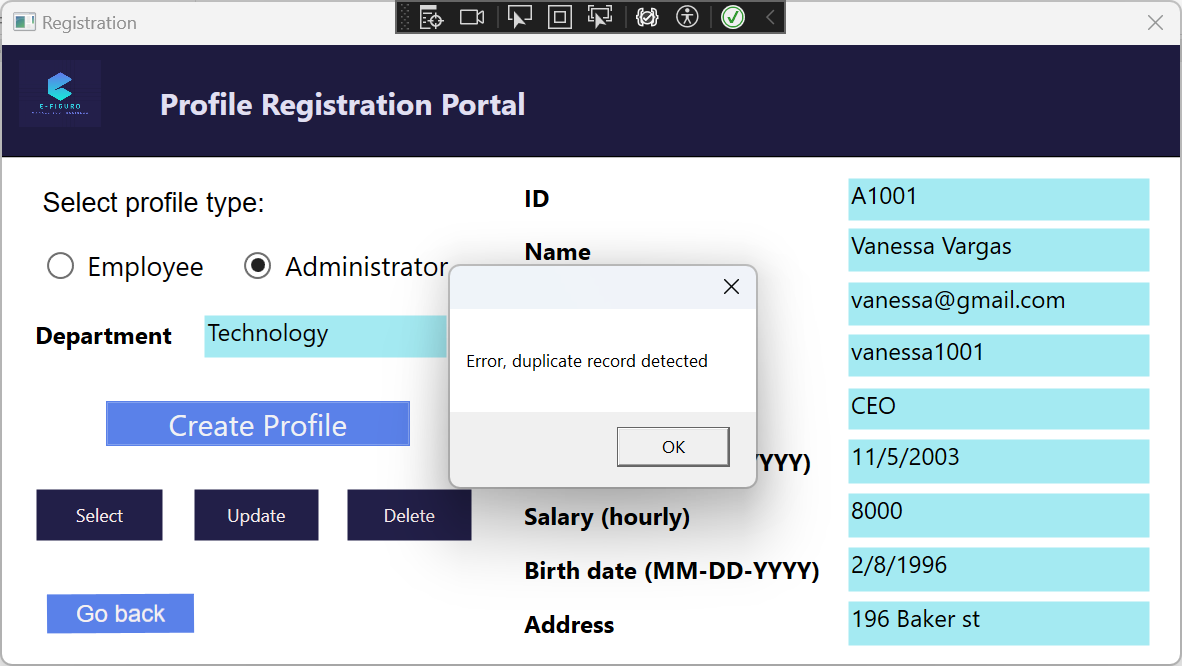
Error message if user tries clicks “Create Profile” without filling all the required fields



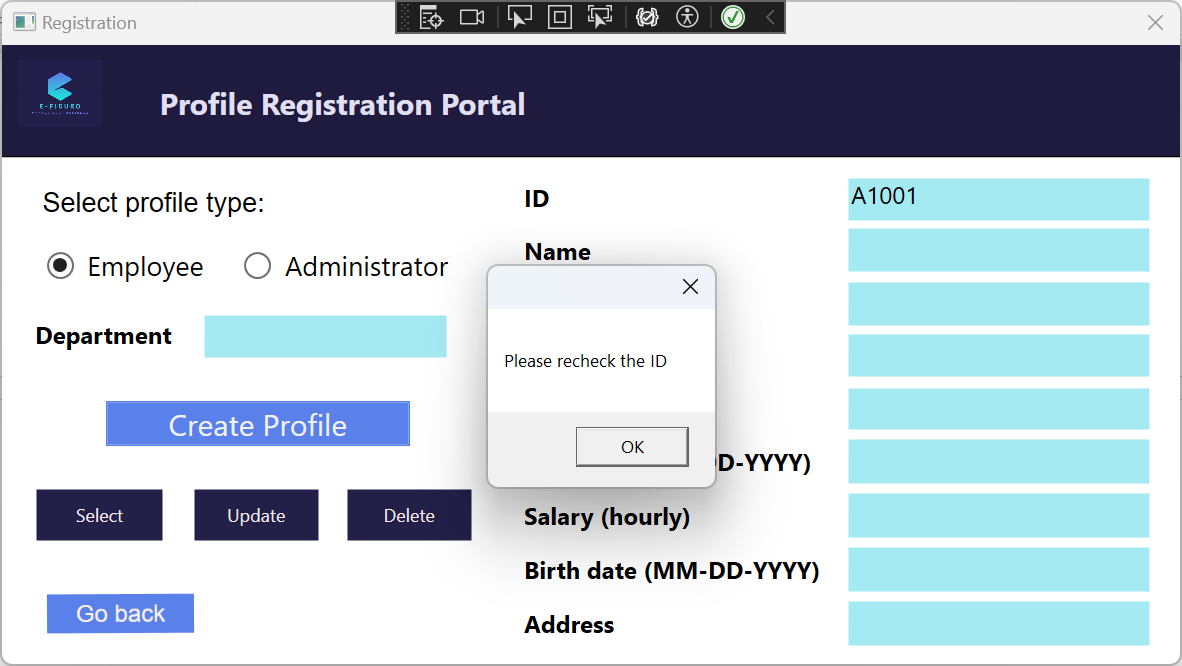
Error message if user does not input an ID when clicking “Select”



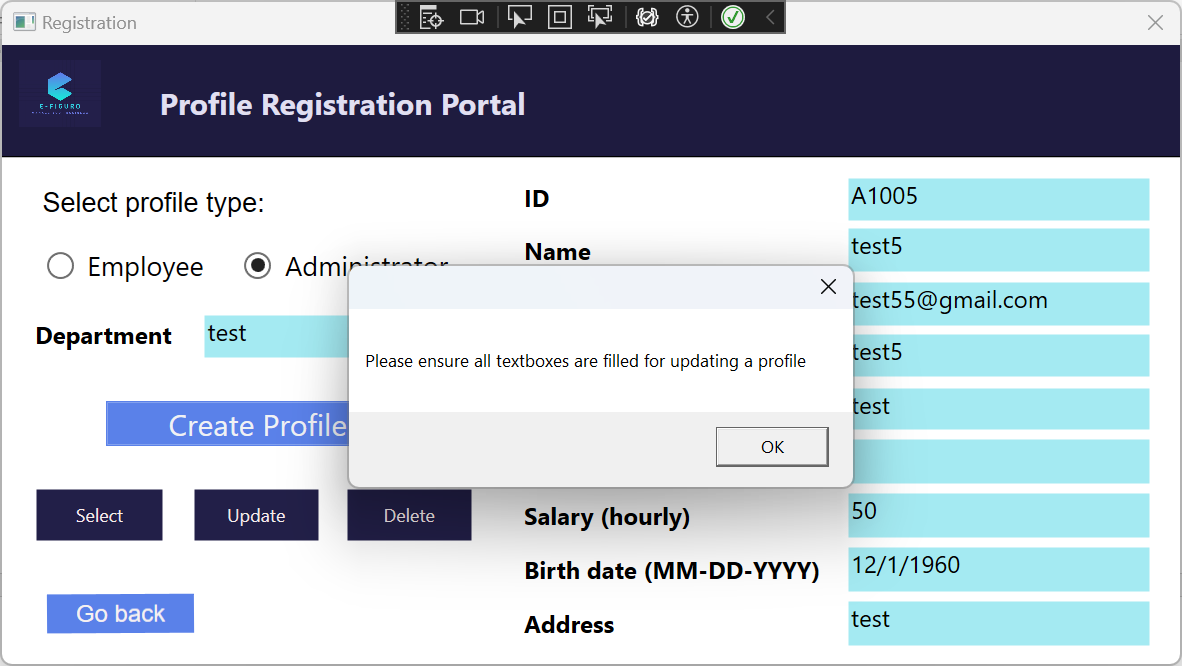
Error message if user tries to “Create Profile” but there is a conflict due to duplicate primary key



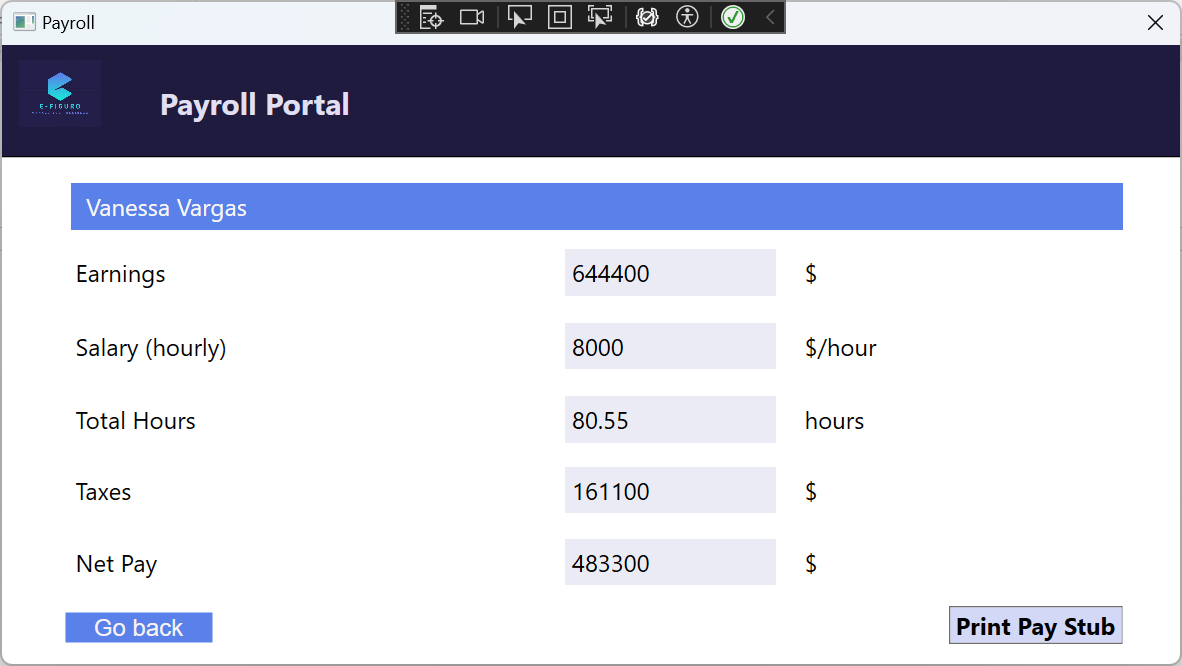
Error message if user enters an ID and clicks “Select” but they did not input the right ID or they put the wrong type of profile type for that ID



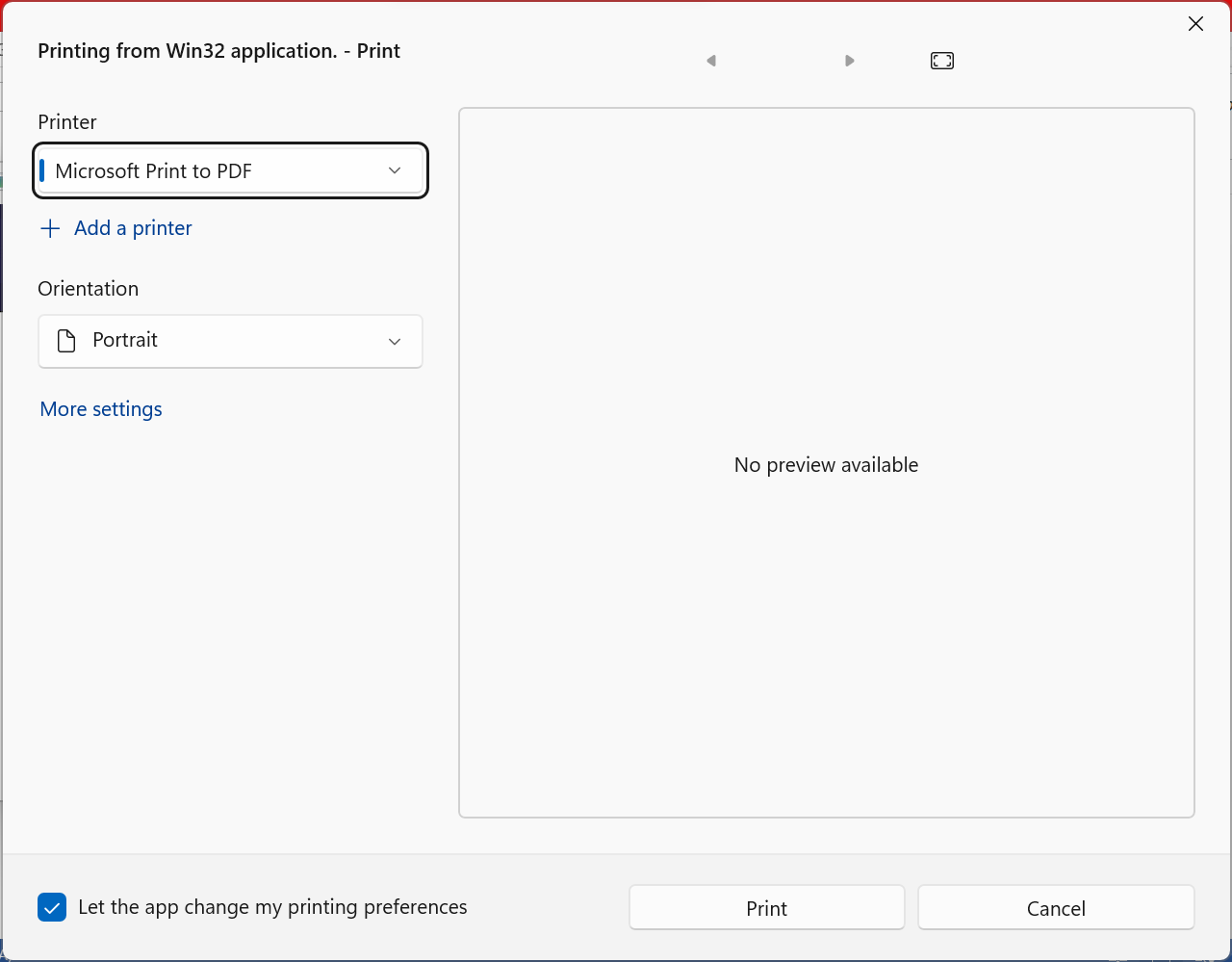
Error message if user tries to “Update” without filling all required textboxes



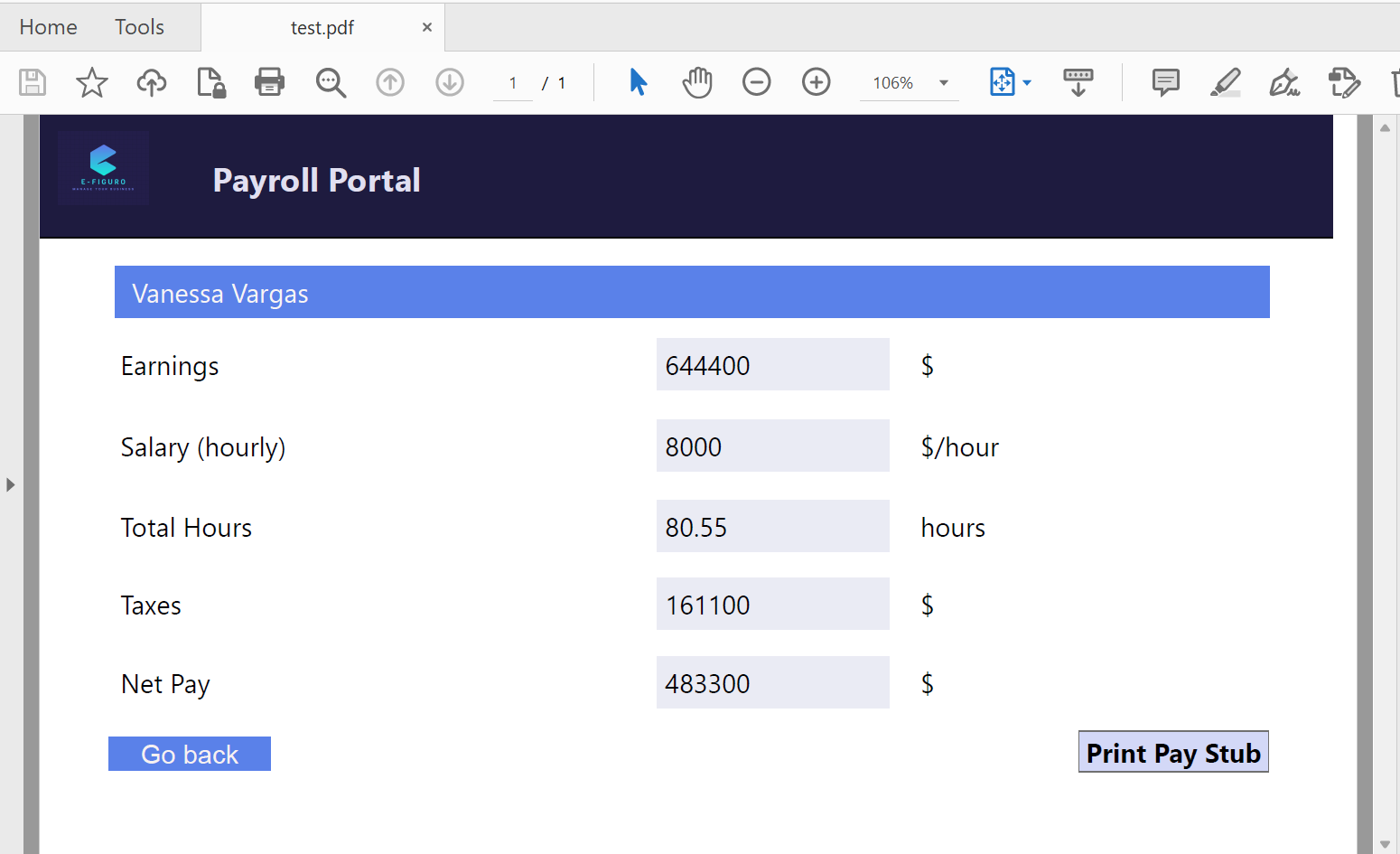
Payroll portal – information automatically pulled from database through userID and relevant calculations done once window is opened



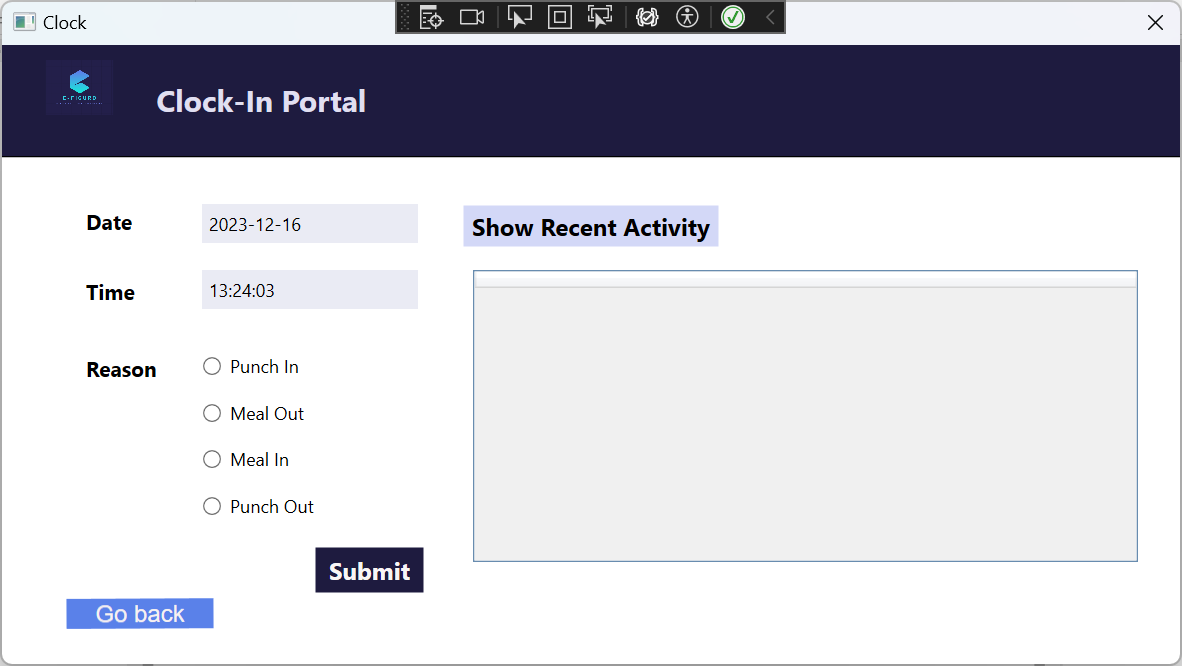
Print window that opens once “Print Pay Stub” is clicked



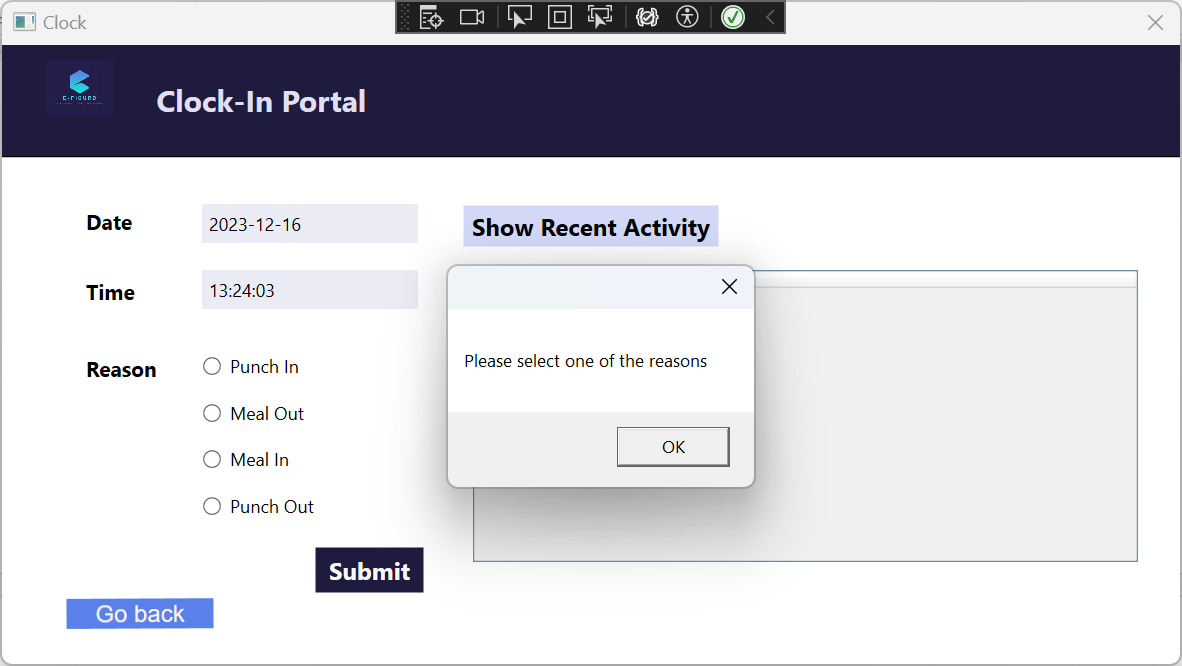
The resulting print page (saved to PDF)



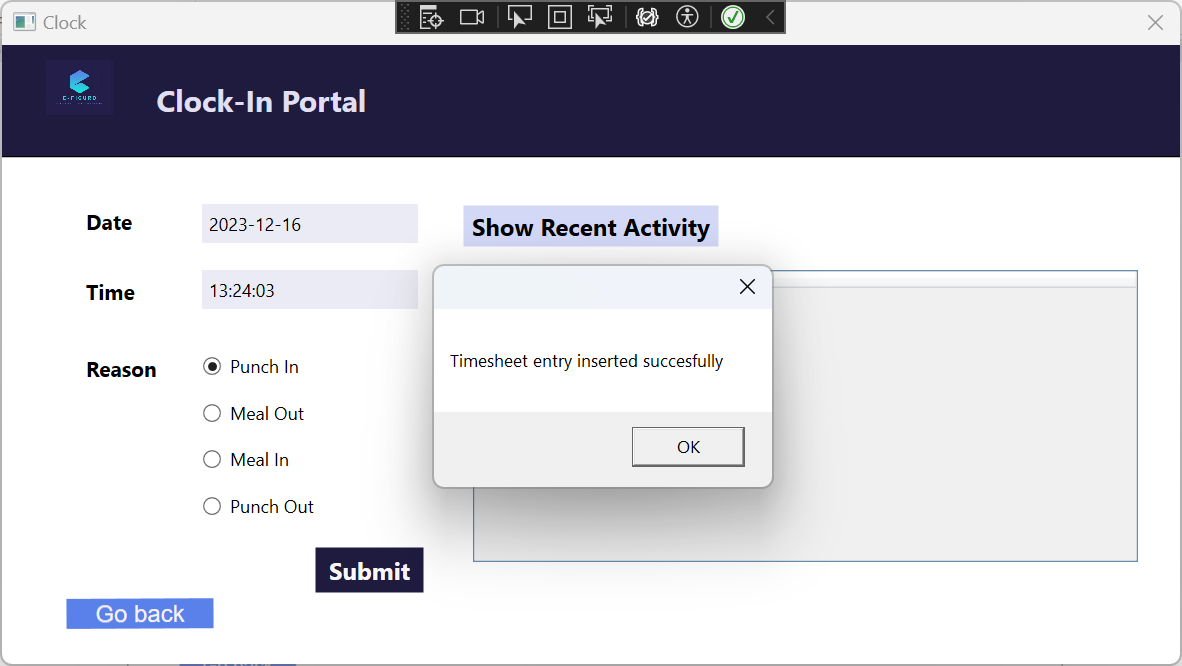
Clock portal (Date and Time automatically initialized based on when the user clicks on this portal)



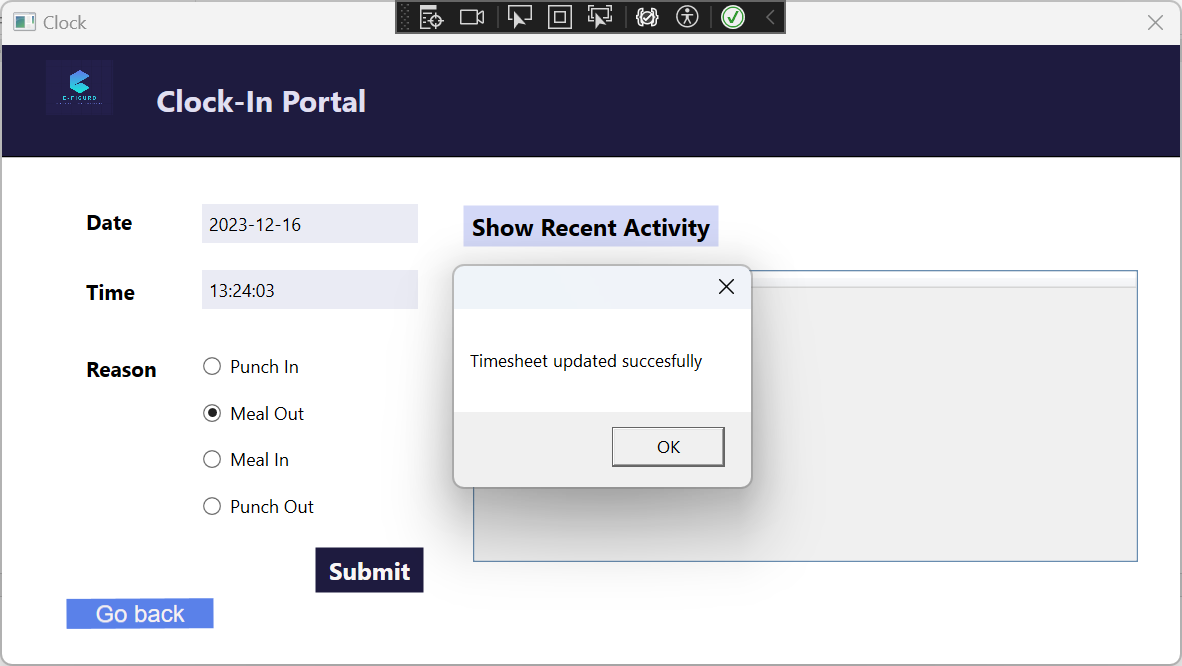
Error message if user tries to click “Submit” without selecting one of the reason radios



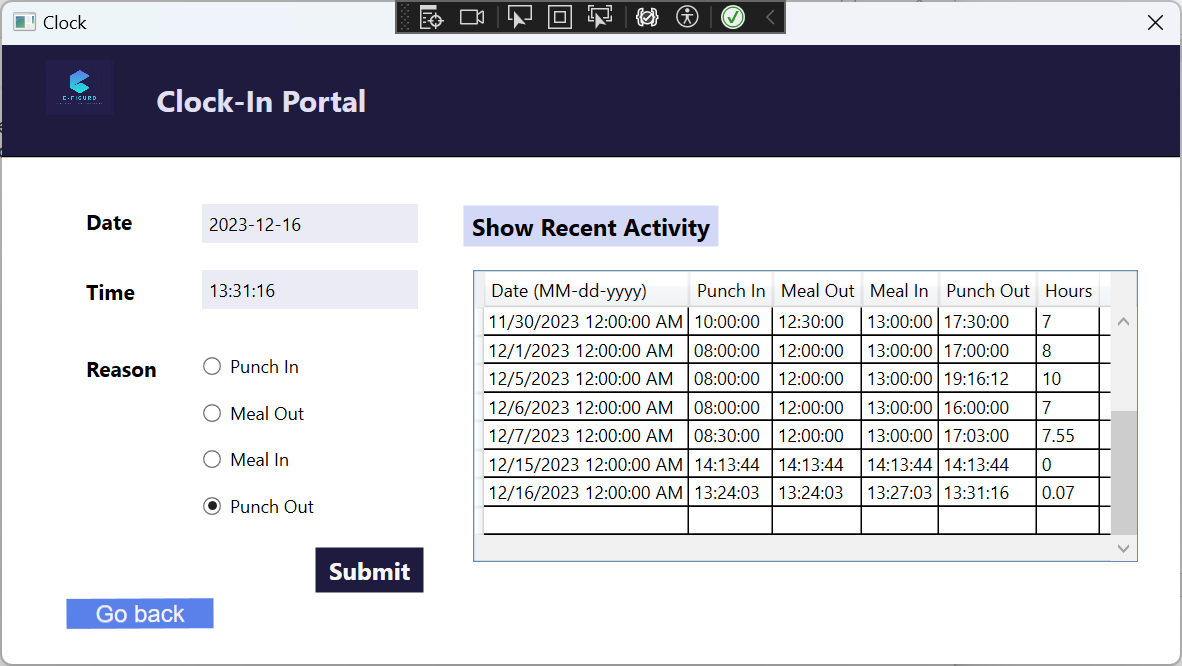
Success message if “Submit” is clicked while “Punch In” is checked



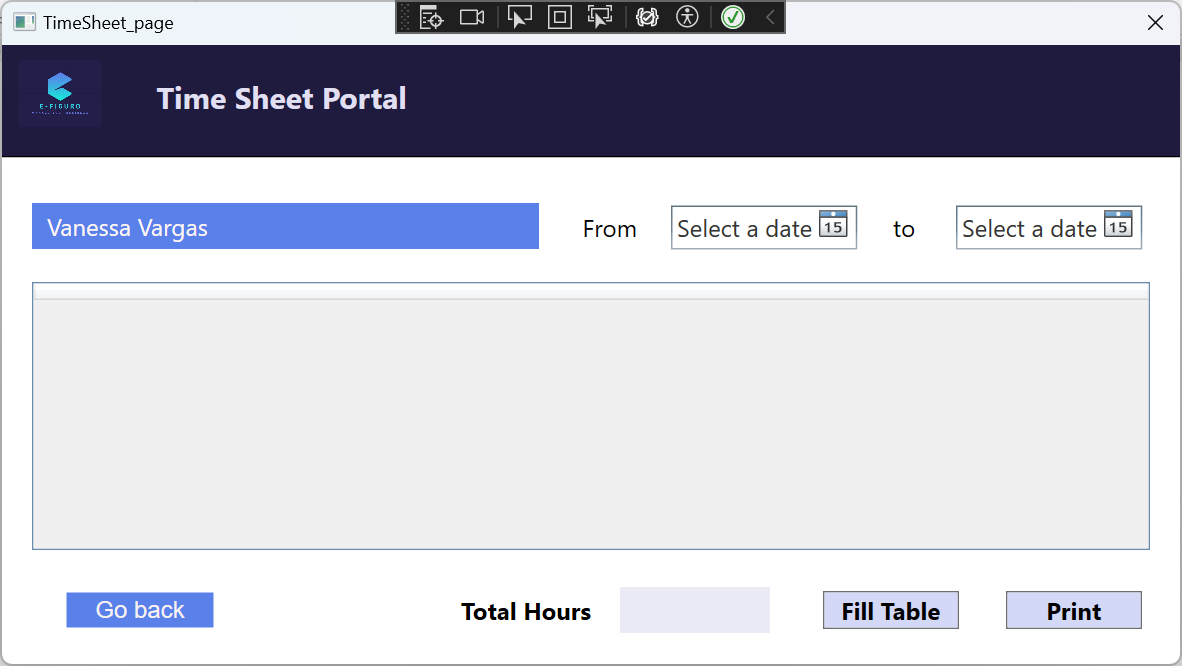
Success message if the “Submit” button is clicked while “Meal Out”, “Meal In”, or “Punch Out” is checked



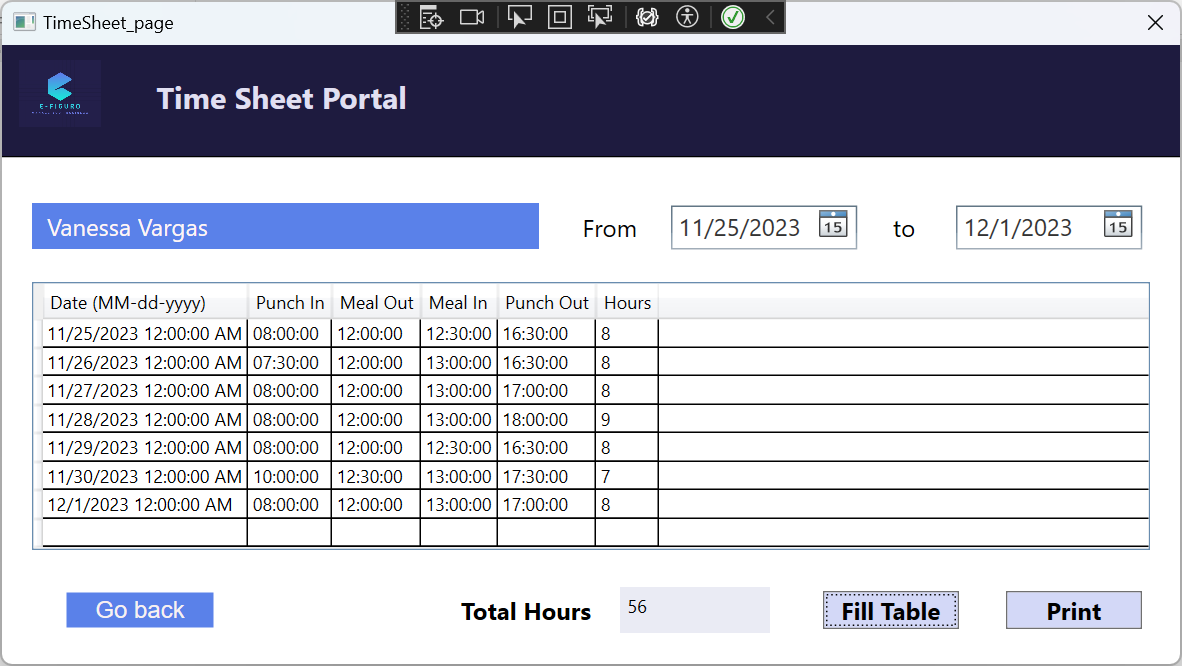
If “Show Recent Activity” is clicked, the calculation for Hours column is done once “Punch Out” + “Submit” is clicked



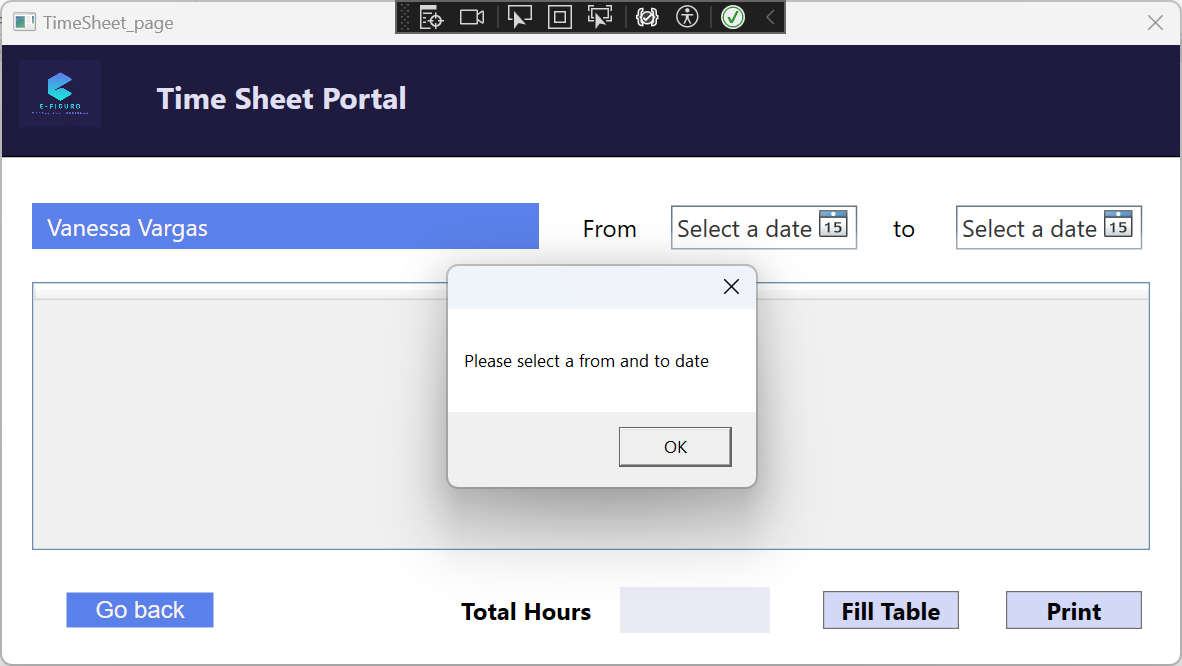
TimeSheet portal



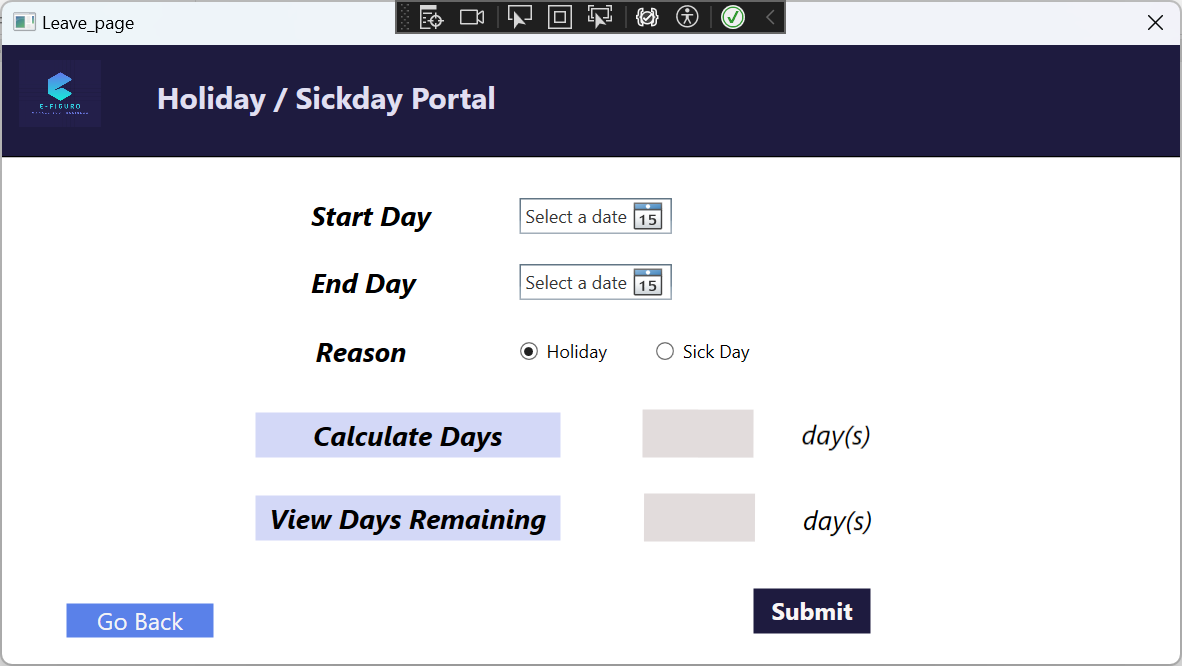
Datatable fills with entries from “DatePicker” to “DatePicker” with calculations being done and filling the label “TotalHours”



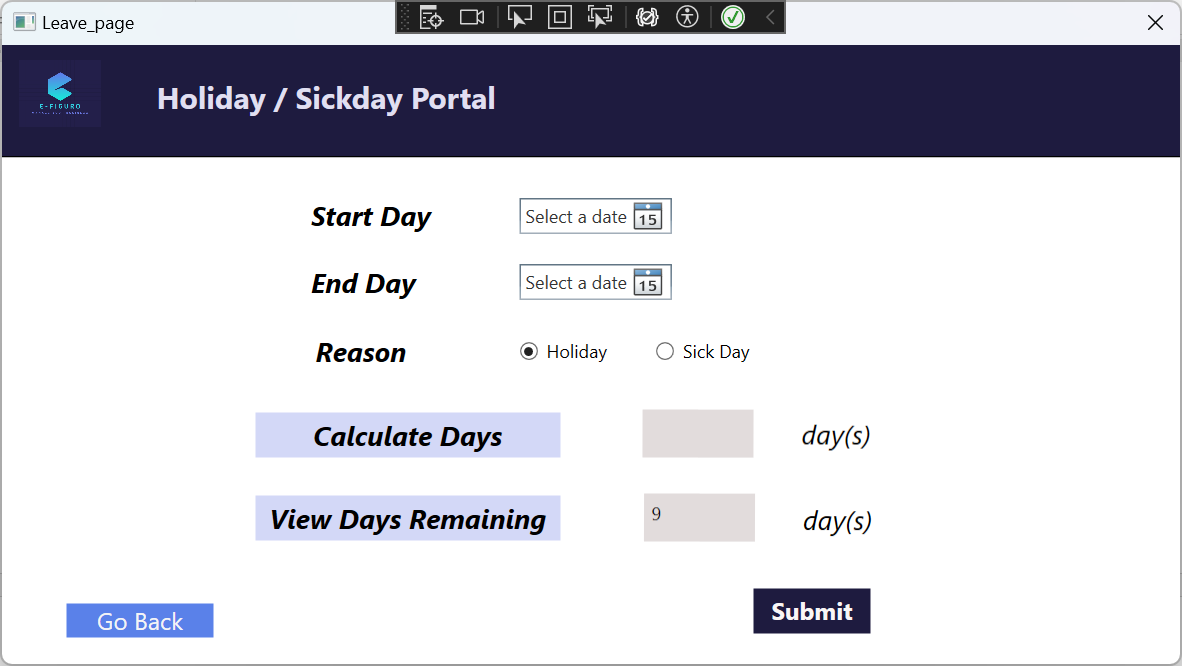
Error message if “Fill Table” button is clicked without selecting dates



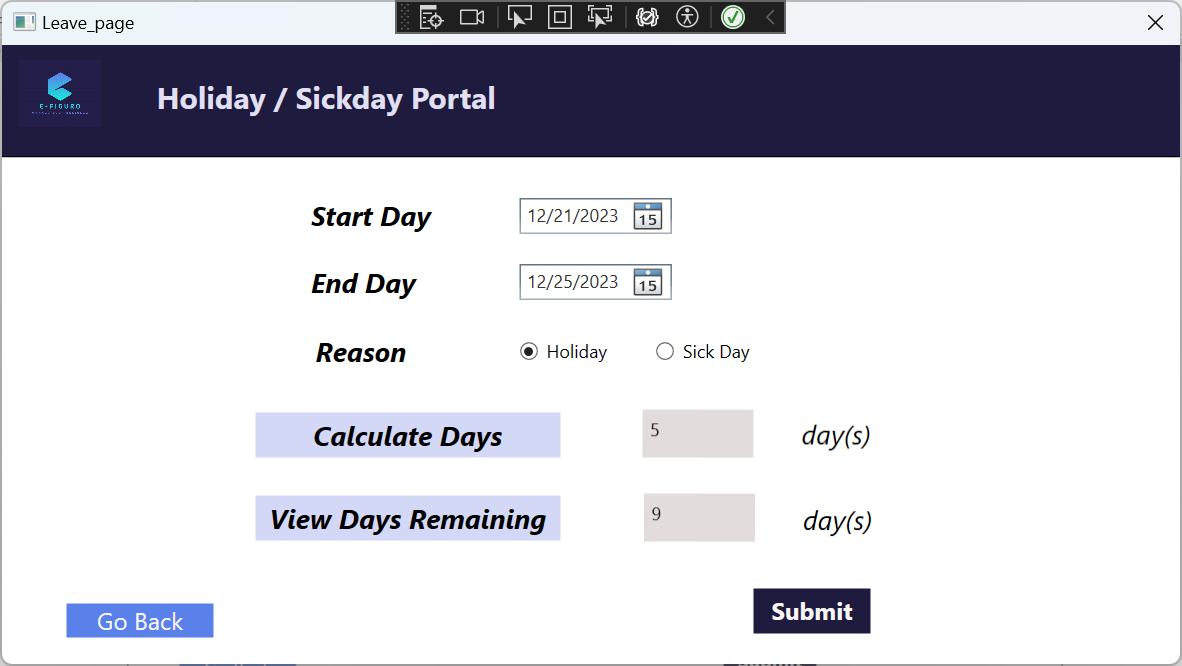
Holiday/Sickday portal



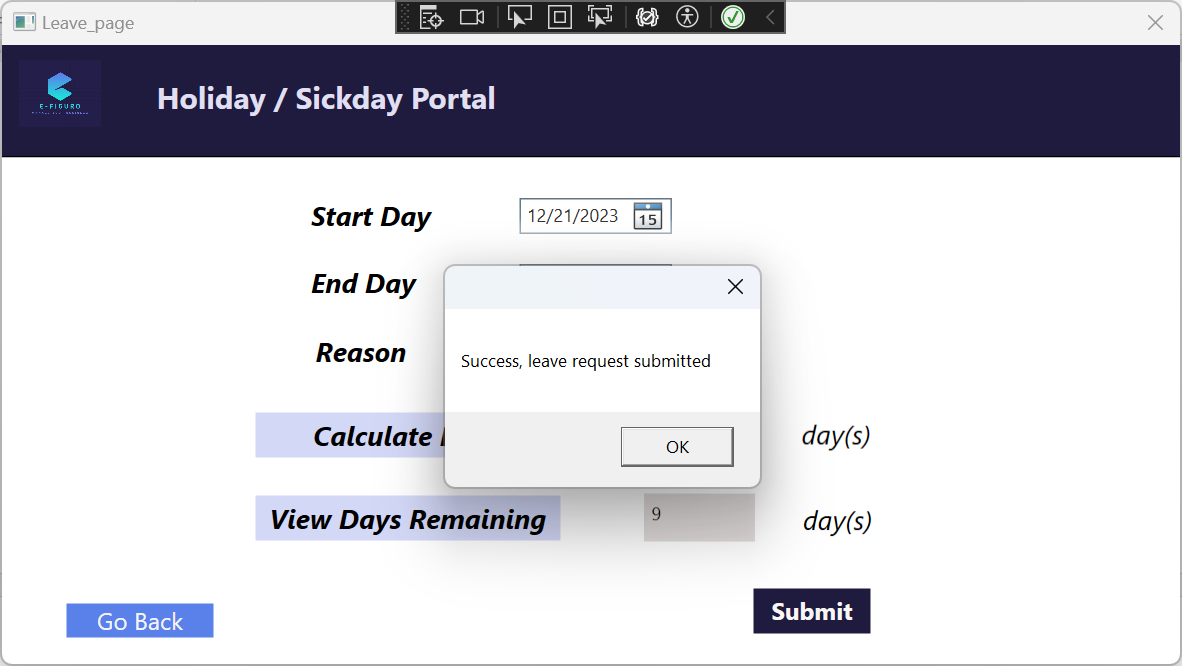
If “View Days Remaining” button is clicked, will perform a calculation to see how many “holiday” or “sick day” days are remaining for the user



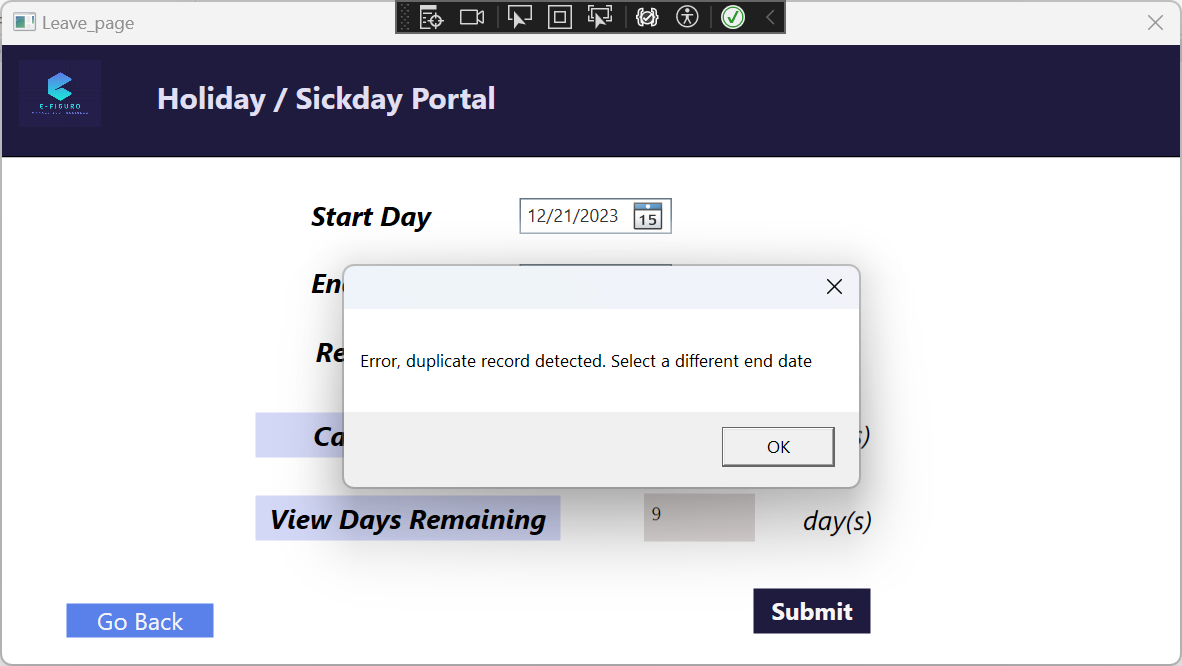
If “Calculate Days” button is clicked, performs calculation based on date of “Start Day” and “End Day” selected

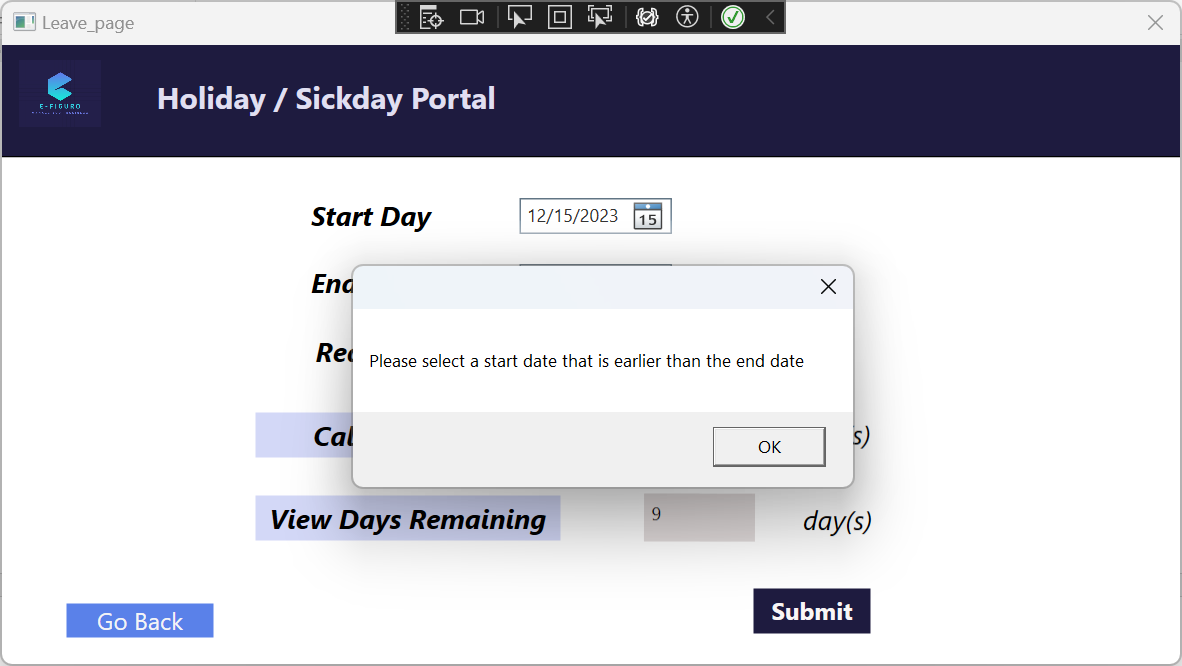


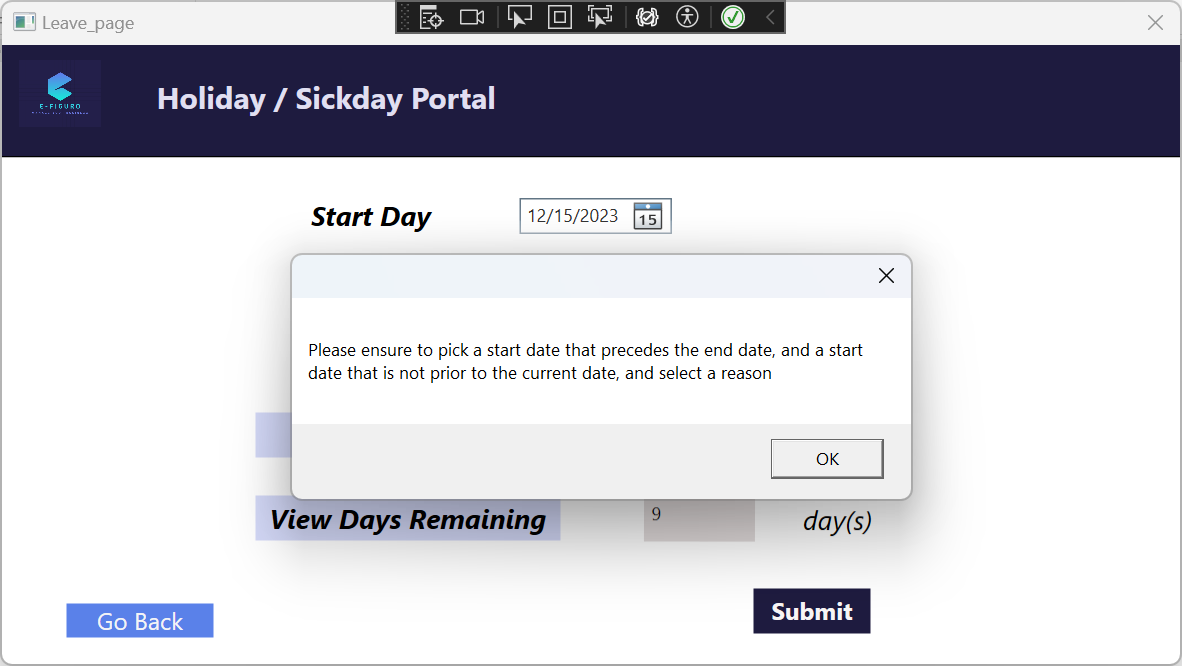
Success message if “Submit” is clicked and everything is okay



Error message if user tries to submit 2 leave requests that have the same end date



Error message if user tries to click “Calculate Days” when selecting a “Start Day” that is later than “End Day”  


Error message if user tries to click “Submit” when selecting a “Start Day” that is later than “End Day”  


**Missing in the current version**:

* Add exception handling for TimeSheet portal: if a start date that is later than end date is selected and “Fill Table” is clicked
* Add print preview when clicking on “Print page” (Payroll and TimeSheet portals)
* Add exception handling for Registration portal: if clicking on “Delete” and the userID doesn't exist already
* Add exception handling for Leave portal: if the user does not select a start or end date and then clicks on “Submit”