Bakery Management System

Criterion E: Evaluation

Success Criteria	Status
User-friendly, intuitive Graphical User Interface	Achieved: Various functioning GUI components, self-explanatory function, placeholders to explain where and what data to input, specific error messages.
Different profiles for managers and staff with different options.	Achieved: Register and login using designation to send user to "Admin" or "Staff" pages.
One profile per person (checked through phone number)	Achieved: Checks the database for a user with the entered phone number. If there is a match, error message is displayed.Two users cannot also have the same username; checks database, if match, error message is displayed.
Login page with username and password	Achieved: Checks if username, password and designation match a record in the database. Sends user to next page based in designation.Forgot username/password is also an option. User inputs personal data, program checks against records in database, if there is a match, displays username and password.
Link to database to store and manipulate data	Achieved: Database created in phpMyAdmin, through the XAMPP server, data is stored and manipulated using MySQL queries, connection is made by creating a class and using java.sql.* - (Connection, DriverManager)
Table showing orders for a searched date	Achieved: Date is entered (validation checks performed). Database is checked for all orders and ones with a matching date are displayed in the table.
Table showing customers' information for ones who haven't returned dish to the admin.	Achieved: When entering order details, Yes/No is picked for whether the order is using returnable dish and the status of the order is also selected. The program checks the database for records

	where dish is 'Yes' and Status is anything but 'Returned'
Table showing attendance for a searched date for the admin.	Achieved: All records in database are checked against the entered date and the attendance records are shown for that matching date.
Manage inventory by inputting product name, type, store, quantity bought/used, price per unit and date bought.	Achieved: User can enter item detailsto store in database and also enter quantity to record quantity of items used.
Keep track of inventory for each product by entering product name.	Achieved: Can select item name from dropdown listand the program calculates and displays the stock remaining for the item.
Keep track of orders search by month.	Achieved: Can select month from drop down list and the program checks the database and all matches are displayed in the table.
Manage orders by inputting customer name, contact, item, amount, delivery date and time, whether a dish is also given and status.	Achieved: Can enter order and customer details to add, update (edit) and delete orders. The program takes the entered data as input and performs the action based on the button pressed.
Addingother expenses	Achieved: Can enter expense details to add, update (edit) and delete expenses. The program takes the entered data as input and performs the action based on the button pressed.
Chart displaying popular orders – order statistics.	Achieved: Can select month and a pie chart showing the quantity of each dish ordered during that month.
Calculating expenses for inventory and staff by month.	Achieved: Algorithm calculates totalsalaries and inventory (for each month)and displays them in the table.
Calculating revenue for orders and kulfis separately.	Achieved: Checks the item name in the orders table and calculates the total for kulfis and the total for all other items and displays them in the table
Manage staff details	Achieved: Can enter staff details to add, update (edit) and delete staff details. The program takes the entered data as input and performs the action based on the button pressed.
Manage staff attendance with start	Achieved:

time and end time	Can enter attendance details to add, update (edit) and delete attendance. The program takes the entered data as input and performs the action based on the button pressed. End time is originally set as null and can be updated based on when the staff finishes working that day.
Calculate overtime so that admin can decide how to compensate them	Achieved: Program calculates hours worked by the staff based on start and end time and compares this with the number of hours they are supposed to work based data in the staff details table.
Validation checks for input data	Achieved: Each input field is checked with appropriate validation checks and error messages are displayed if there is an error in the entry.

Future Development (refer to Appendix I and II for client and staff evaluation interviews):

The client was satisfied with the application and confirmed that it was fully functional, easy to use and helped make their business more professional and organised by creating a singular space to store all their data and also automatically perform all the calculations, almost completely eliminating human errors like forgetting to consider some data, mathematical errors and misplaced records. And so, this application has increased the client's efficiency by reducing time worked and errors made. However, after the evaluation, it was determined that there is room for extensions for the objectives of the app.

To start with, the client said that it could be made into an online application so that customers can order directly and view the status of the orders. This could also include a message board option for the customers to give specific directions or feedback and also a private messaging option between the admin and the staff for instructions. They also wished to turn it into a mobile app for ease of access and portability. The client said they would also like a way to enter data that eliminates the risk of typing errors. And so, another improvement that could be made is the use of barcode scanning to scan the inventory and update the database with price and quantity, removing the risk of typing errors.

Furthermore, encryption can be used to increase data security. Data backup options, such as downloading and exporting or making copies of the database through the program, can be added to protect the data from system crashes and accidental deletion.