

Paradigms of Programming – Project Report

E-Commerce Framework

AYUSH MITTAL PARTH PATEL RAVINDRA KUMAR SUMAN SAURAV PANDA

Aim: -

Building a design for an e-commerce platform, which can serve as a framework for future implementations of the same with possible feature extensions.

Languages and Framework: -

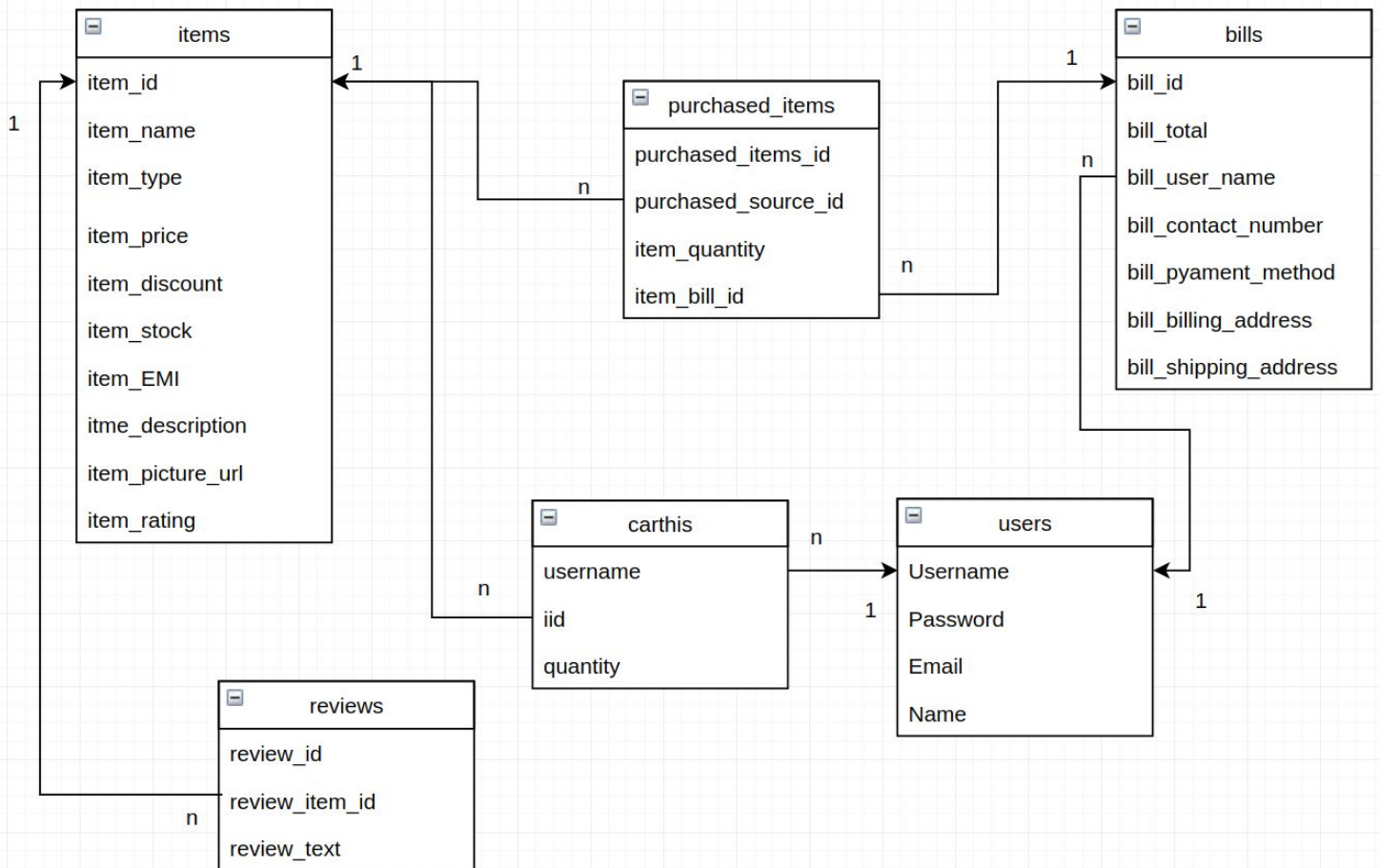
1. Frontend: - Bootstrap Framework, Font awesome toolkit
2. Backend: - PHP
3. Database: - MySQL

Design Choice: -

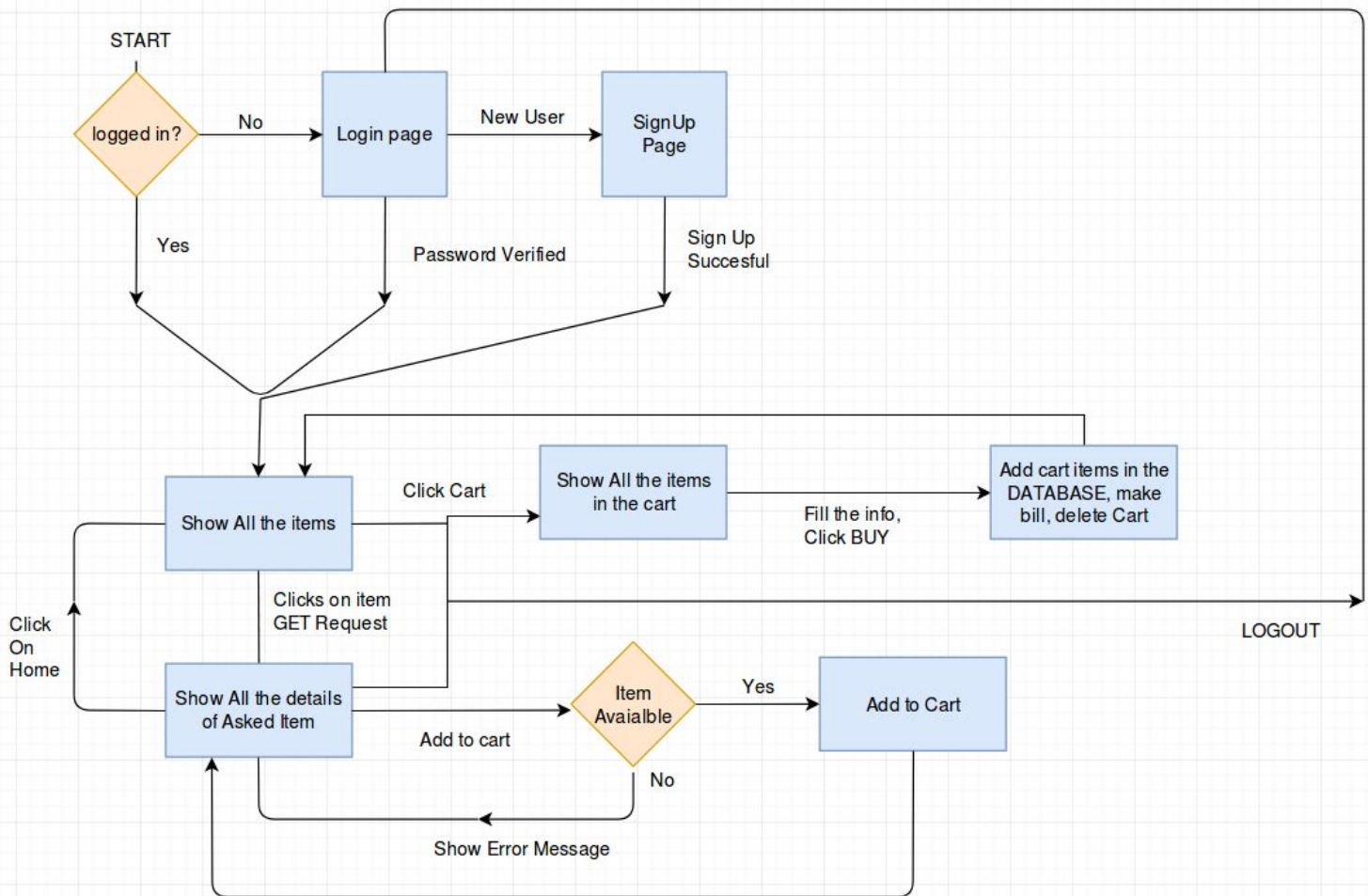
1. Frontend: -
 - a. Bootstrap Framework: - Framework provides better security and relative ease. Bootstrap provides basis for **responsive design**⁺ and is highly efficient.
 - b. Font Awesome toolkit: - Handy since it's already integrated with bootstrap. This toolkit provides **675** pictographic icons which help create **vector graphics**.
2. Backend: -
 - a. PHP: - It provides **easy, fast** and **efficient** integration with web based applications and **database** system like MySQL. It is also **scalable** and is more **object oriented**.
3. Database: -
 - a. MySQL: - Simplicity, and **predefined** database schema⁺.

1. Adaptability to user's choice and screen. So the website can now automatically adapt its viewport policies based on device size. (Mobiles-various sizes, Laptop).
2. In our usage we have fixed structure for database models(tables) and hence don't need to resort to other database system which would compromise simplicity.

Database Schema: -



FLOW CHART



Features: -

1. Membership Only Access: - The plan was to incorporate a private e-commerce platform for a group of users, for e.g. IIT PKD merchandise only available to IIT PKD students.
2. Session Save: - Sessions are saved over both cookies and database. Cookies help preserve the login information over events such as tab, browser close event. Database helps preserve the same information over even logout.
3. Summary/Full View: - Initially the products appear in summarized form then each item's details can also be viewed.
4. An easy-to-use checkout – Checkout Portal is really simple to use.
5. Purchase History
6. Quantity Check : - When adding to cart stock quantity would be checked
7. Reviews
8. Security – Protection layer has been added to prevent SQL Injection.

Improvements/Additions: -

1. Promotion and discount code tools
2. An integrated blog or articles section
3. Catalog Management – Searching and sorting by categories, reviews, rating , bestselling, relevance
4. Related Item display (Item Recommendations)
5. Mass Email Policies
6. Payment portal using payment gates like Paytm,UPI , then this could have been realized as e-commerce website for canteen running on mobiles.
7. Loan Option: - Not immediately pay money but add in ledger.
8. Wish-Lists
9. User Customizable backgrounds.
10. Price Comparison graphs and charts.

Development Cycle: -

1. We planned out to use python for backend and its module TKINTER for content display.
2. But then the inherent difficulty of python's TKINTER and usability of e-commerce software as compared to a website, we started to think of implementing a website based system rather than a software platform.
- 3.

We then chose PHP for backend instead of python then using python for rendering the HTML pages is because PHP is more freely available on free web-hosts online and thus give chance to people to try out the framework using free web hosting services.

4. After we implemented it with PHP and Bootstrap we realized we were right to use PHP instead of Python as it allowed easy bootstrap access which reduced our development time and increased our site's responsiveness.

Some Challenges: -

1. The design and database schema to start with was a bit problematic as we were then unaware of the situations that might arise like stock quantity, session saves.
2. The Updating of cart items in database when items added to cart also left us perplexed as normally AJAX is used for it, but then we created a workaround wrapper to implement it with PHP.

Insights: -

PHP is inherently more inclined towards object oriented. PHP means creating an integral code object that has certain properties and executes certain actions depending on the user input. We explore the concept of OOP as we form classes for items, and then pass its objects/instances around to transfer information.

Our implementation contains three classes for items, based upon the attributes associated with different items in different scenarios (Front page items, Single page item, Cart Item). So when a query is made array of such class objects is passed around which have an associated function to set their variables.

Contributions: -

1. Design: - Everyone equally contributed their inputs in how the database schema and website design should be.
2. Coding part was divided as per the person's familiarity with the concepts needed. In our model, then we had
 - A. UI: - handled by Parth Patel.
 - B. Backend and Integration: - Ayush Mittal, Parth, Suman Saurav Panda.
 - C. Database Formation: - Ayush, Parth, Suman, Ravindra Kumar.
 - D. Report: - Ayush, Suman.
 - E. Database Entry⁺: - Ravindra Kumar

Then on the need basis the other members helped the core member in their task. This model we believed achieved us high efficiency.