**AIO Tools**

Website providing dashboard and tools to simplify tracking for small vendors.

**Abstract**

Provides a simple set of tools in form of an online dashboard. It would be able to help small and medium vendors to use and track their daily, monthly and yearly sales and growth easily in form of a graph and charts in just few clicks.

We would be following the MERN stack for the implementation of the project with Amazon AWS and a free domain using freedns.afraid.org. The MERN stack constitutes of the following components:

1. MongoDB
2. Express JS
3. React
4. Node JS

The Mongo DB is used for the online database, whereas Node JS is used for the implementation and java run-time environment for development. React, its components and libraries are used for functionality and user interfaces. Express JS is being used for the web application framework. We will providing latest UI Elements to provide the customer or user with an attractive and easy to use Dashboard and web application for easier access and work.

**Keywords**

**Generic Technology keywords**

Operating System – Windows, Linux

Databases – MongoDB

Programming – HTML, CSS, React.

Network and Middleware

Software Engineering

**Specific Technology keywords**

MERN Stack

Mongo DB

Express JS

React

Node JS

**Project type keywords**

Application development, retail, sales, online store, expense tracking.

**Functional components of the project**

Store –

* Sells different goods.
* Tracks Products available and their popularity

Customer -

* Buys Goods and products available
* Register for benefits

Store Manager -

* Monitors sales of store.
* Tracking of popular items and products
* Tracking of geographical locations of orders

Finance -

* Keeping a record of admins on the dashboard
* Keeping a record of performance of employees.

1 – Transaction of the customer would be recorded in the Database with following details

Name

Delivery Address

Product

Phone no.

Email id

2 – Store Manager can keep a track of the daily sales and increase in customers.

3 – He can also look at the performance of the employees

4 – He can keep track of the employees attendance

5 – He can keep track of the products.

**Steps to start-off the project**

Learning the following :

1. HTML
2. CSS
3. Node JS
4. Express JS
5. React
6. SQL

**Requirements**

**Hardware requirements**

|  |  |  |
| --- | --- | --- |
| **Number** | **Description** | **Alternatives (If available)** |
| 1 | PC with 2 GB hard-disk and 256 MB RAM |  |
|  |  |  |

**Software requirements**

|  |  |  |
| --- | --- | --- |
| **Number** | **Description** | **Alternatives (If available)** |
| 1 | Windows 98 or higher | Ubuntu or any debian distro |
| 2 | Tomcat server | Any freely available server |
| 3 | Any editor to write code | VS Code |
| 4 | Any RDBMS | MongoDB |

**Manpower requirements**

2 to 3 students can complete this in 4 – 6 months if they work full time on it

**Milestones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Milestone Name** | **Milestone Description** | **Timeline**  Week no.  from the start  of the project | **Remarks** |
| 1 | Requirements Specification | Complete specification of the system (with appropriate assumptions) | 2-3 | Attempt should be made to add some more relevant functionality other than those that are listed in this document. |
| 2 | Technology familiarization | Understanding of the technology needed to implement the project. | 4-5 | The presentation should be from the point of view of being able to apply it to the project, rather than from a theoretical perspective. |
| 3 | Database creation | List of tables and attributes of each of them and a data set to check the app and flow. | 5-7 | It is important to finalize on the database at this stage itself so that development and testing can proceed with the actual database itself. |
| 4 | High-level and Detailed Design | HLD  Assumptions  Flow charts  List of screens  DLD  Pseudo code or algorithm for each the activity | 7-9 | The scenarios should map to the requirement specification (ie, for each requirement that is specified, a corresponding scenario should be there). |
| 5 | Implementation of the front-end of the system | Implementation of the main screen giving the login, screen that follows the login giving various options, screens for each of the options | 10-12 | During this milestone period, it would be a good idea for the team (or one person from the team) to start working on a test-plan for the entire system. This test-plan can be updated as and when new scenarios come to mind. |
| 6 | Integrating the front-end with the database | 1 - Screens connected to data base and updating data base as required. | 12-13 |  |
| 7 | Integration Testing | The system should be thoroughly tested by running all the testcases written for the system (from milestone 5). | 14-15 | Another 2 weeks should be there to handle any issues found during testing of the system. After that, the final demo can be arranged. |
| 8 | Final Review | Issues found during the previous milestone are fixed and the system is ready for the final review. | 16-18 | During the final review of the project, it should be checked that all the requirements specified during milestone number 1 are fulfilled (or appropriate reasons given for not fulfilling the same) |

**Guidelines and References**

SQL, HTML, Express, React, Node - [https://www.w3schools.com](http://www.w3schools.com)

Free DNS – <https://freedns.afraid.org>