Database Management System

Mini Project Report

On

" **FARMER MANAGMENT SYSTEM** "

Submitted in the partial fulfillment of requirements of Savitribai Phule Pune University For the degree of **T.E (Computer Engineering)**

By

**Name Roll NO**

Shyam Lokhande 036

Aditi Joshi 020

Abhishek Dhamdhere 022

*Under the guidance of*

Prof. B.C.Julme

### Department of Computer Engineering

**PVG's College of Engineering and Technology and G.K. Pate (Wani) Institute of Management**

**SAVITRIBAI PHULE PUNE UNIVERSITY 2021-2022**



# PVG's College of Engineering and Technology and G.K. Pate (Wani) Institute of Management

# CERTIFICATE

This is to Certify that

**Name Roll NO**

Shyam Lokhande 036

Aditi Joshi 020

Abhishek Dhamdhere 022

Has satisfactorily completed the requirements of DBMS Mini Project For the degree of

**T.E (Computer Engineering)**

**On**

### ‘Farmer Management System’

Under Guidance of

Prof. B.C.Julme

Prof. D.D. Sapkal

Head of Department

Dr. M. R. Taramble

Principal

**ABSTRACT**

Discussing the highlights of this Farm Management System in PHP, the administrator can deal with each activity in the event that they need. Here so as to utilize the framework, you need to sign in to the framework.

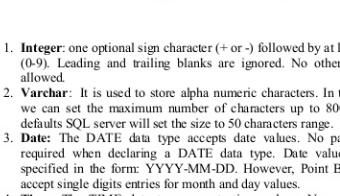
You can either sign in as a rancher or a purchaser. For the time being, this venture comprises of just the rancher side. With the utilization of this super framework, the ranchers can deal with their items without breaking a sweat. They can likewise compose a blog and distribute them.

Keywords : Php, html, SQL , Javascript ,website etc.

1. **Introduction**

The administrator can deal with each activity in the event that they need. Here so as to utilize the framework, you need to sign in to the framework. You can either sign in as a rancher or a purchaser. For the time being, this task comprises of just the rancher side. With the utilization of this super framework, the ranchers can deal with their items without any difficulty. They can likewise compose a blog and distribute them. Plan of this undertaking is quite basic with the goal that the client won’t discover any troubles while taking a shot at it. This framework in PHP helps the client in overseeing stock and exchanges. To run this undertaking you more likely than not introduced virtual server i.e XAMPP on your PC (for Windows).

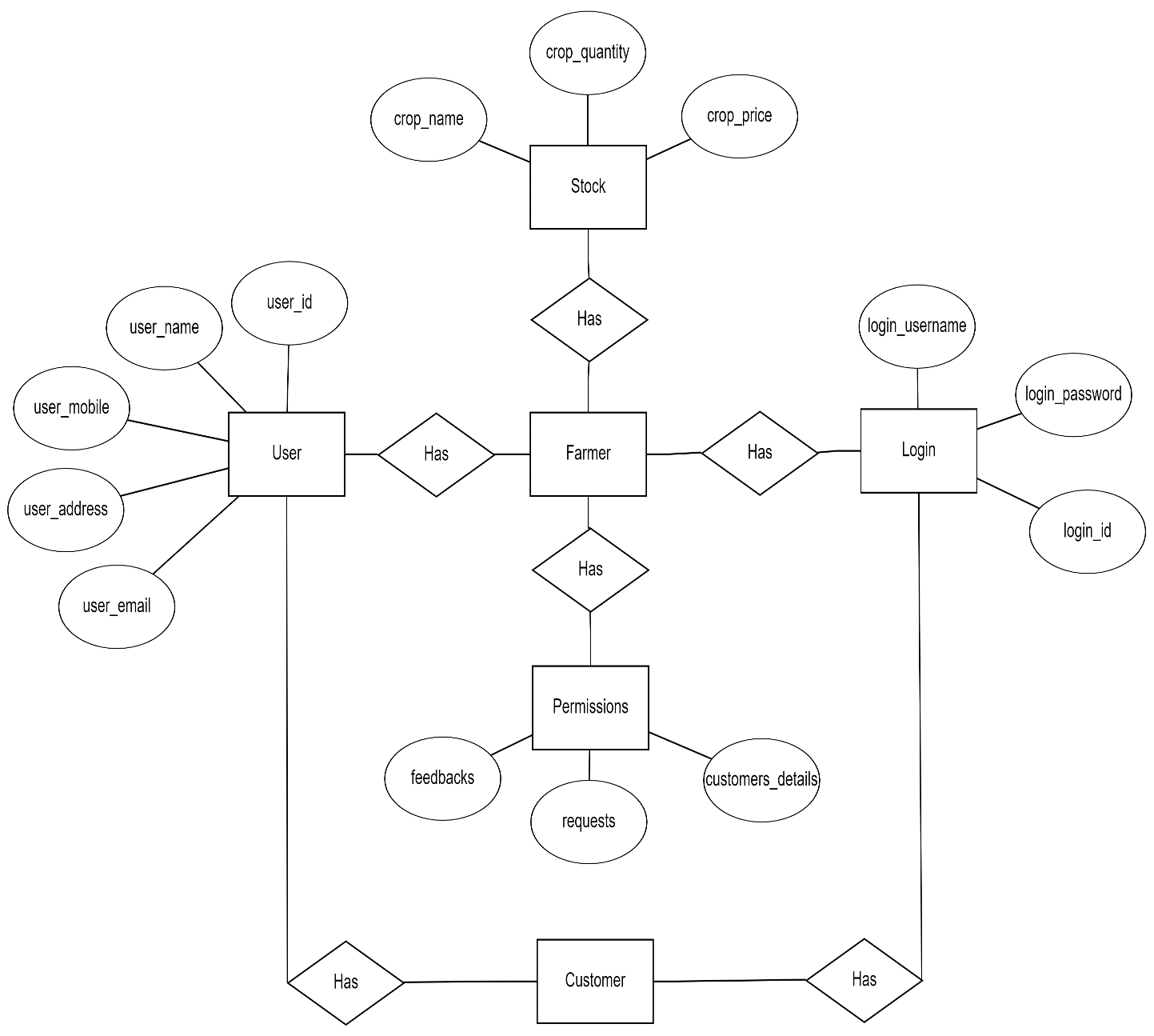
## Data Types



## E R Diagram

An entity–relationship model (ER model) describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types. In software engineering an ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure that can be implemented in a database, typically a relational database. Entity–relationship modeling was developed for database design by Peter Chen and published in a 1976 paper. However, variants of the idea existed previously, some ER modelers show super and subtype entities connected by generalization-specialization relationships, and an ER model can be used also in the specification of domain-specific ontology. An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or foreign key; in the table of another entity There is a tradition for ER/data models to be built at two or three levels of abstraction. Note that the conceptual-logical- physical hierarchy below is used in other kinds of specification, and is different from the three schema approach to software engineering.

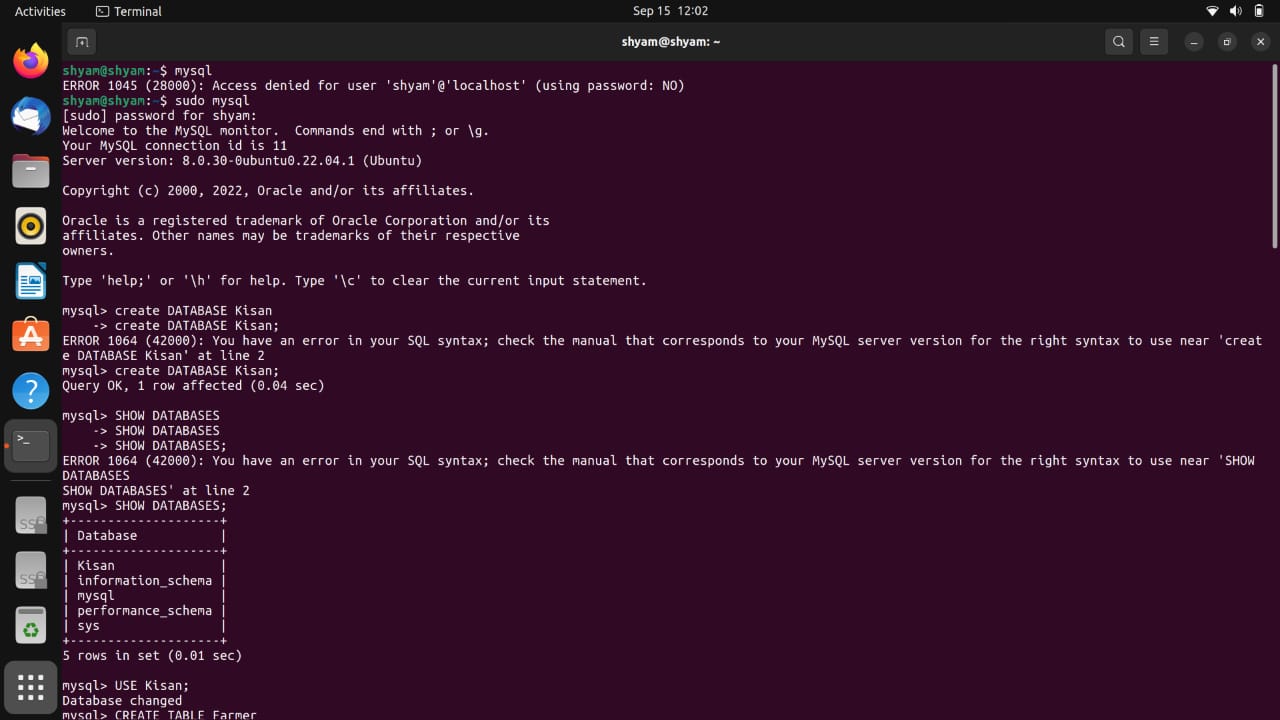
Fig A. The ER conceptual schema diagram for the database.

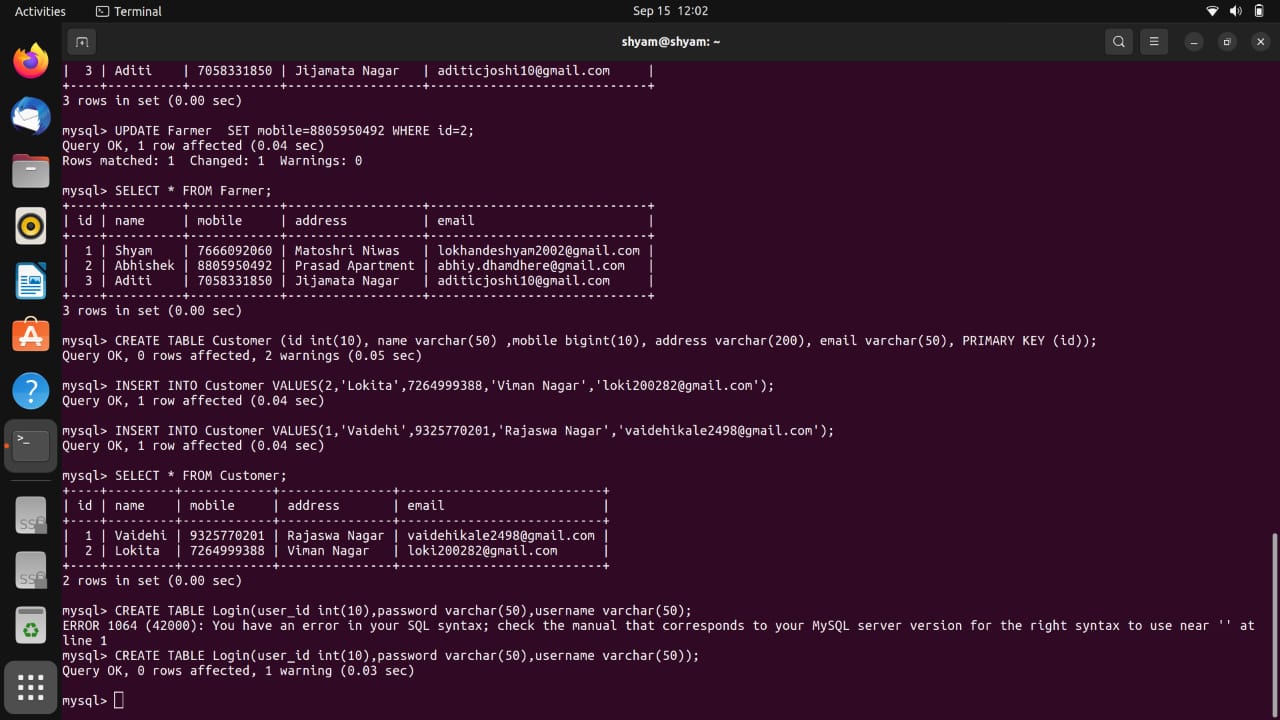
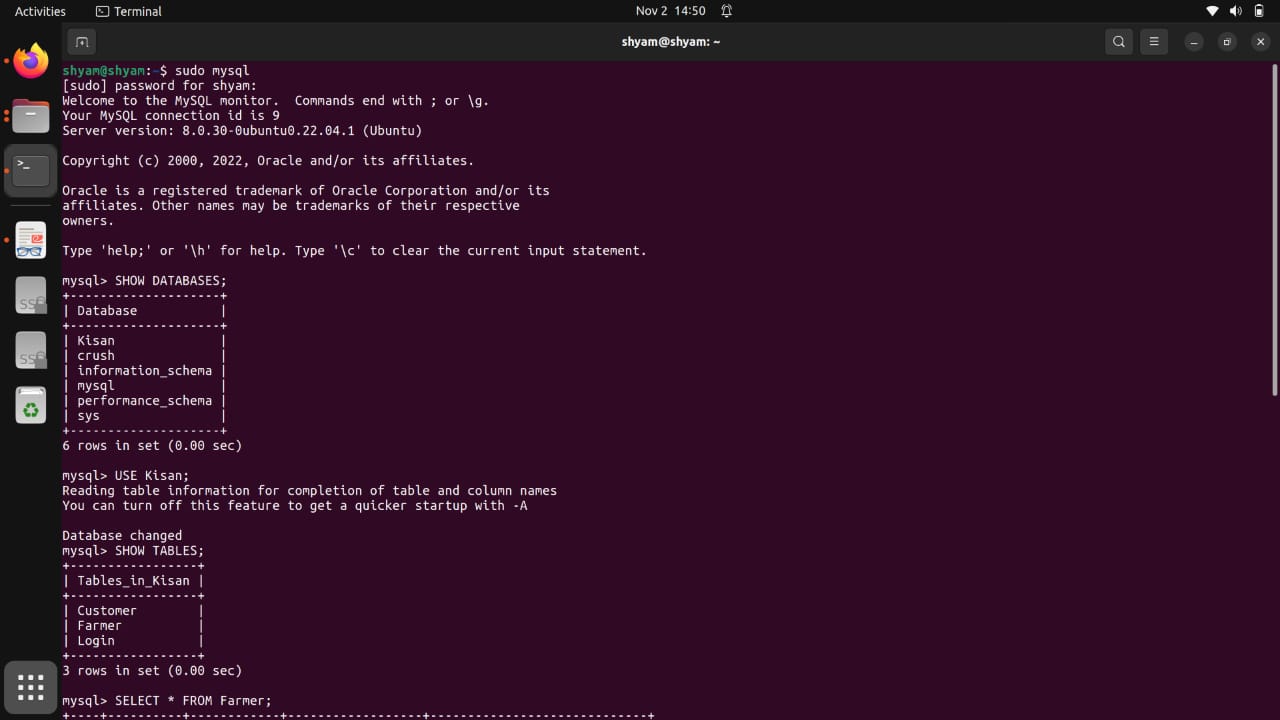


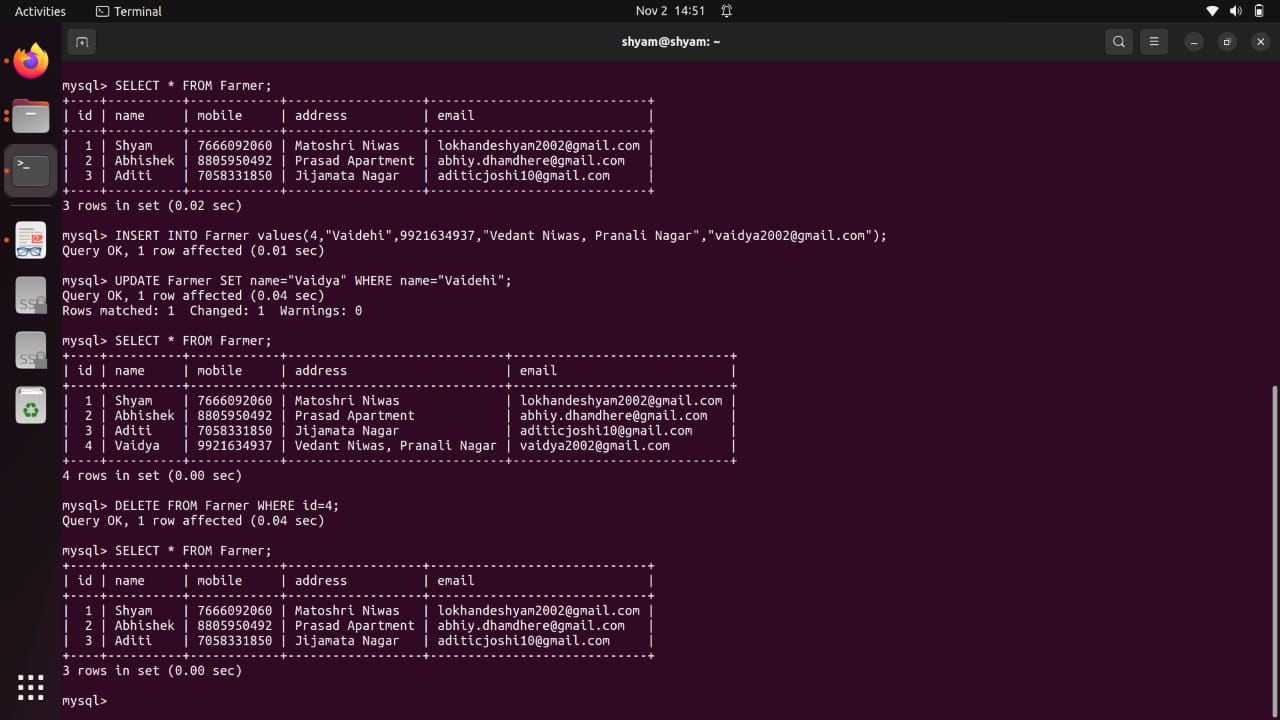
## Schema Diagram

Relational Database Design Using ER-to-Relational Mapping

The ER schema is shown in Figure A, and the corresponding relational database schema is shown in Figure B.







ER-to-Relational Mapping Algorithm

The relational model constraints, which include primary keys, unique keys (if any), and referential integrity constraints on the relations, will also be specified in the mapping results.

Step 1: Mapping of Regular Entity Types. For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes of E. Include only the simple component attributes of a composite attribute. Choose one of the key attributes of E as the primary key for R. If the chosen key of E is a composite, then the set of simple attributes that form it will together form the primary key of R. If multiple keys were identified for E during the conceptual design, the information describing the attributes that form each additional key is kept in order to specify secondary (unique) keys of relation R. Knowledge about keys is also kept for indexing purposes and other types of analyses. In our example, we create the relations CUSTOMER , CAR ,CARTYPE,LOCATION and DISCOUNT in Figure A. The foreign key and relationship attributes, if any, are not included yet; they will be added during subsequent steps. These include the attributes DL NUMBER and PICKUP,DROP OFF LOCATION. In our example, we choose CUSTOMER ID as primary keys for the relations, BILLING, and CAR, respectively. Knowledge secondary keys is kept for possible use later in the design. The relations that are created from the mapping of entity types are sometimes called entity relations because each tuple represents an entity instance. The result after this mapping step is shown in Figure 9.3(a).

Step 2: Mapping of Weak Entity Types. For each weak entity type W in the ER schema with owner entity type E, create a relation R and include all simple attributes (or simple components of composite attributes) of W as attributes of R. In addition, include as foreign key attributes of R, the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s); this takes care of mapping the identifying relationship type of W. The primary key of R is the combination of the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any. If there is a weak entity type E2 whose owner is also a weak entity type E1, then E1 should be mapped before E2 to determine its primary key first.

Step 3: Mapping of Binary 1:1 Relationship Types. For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating

in R. There are three possible approaches: (1) the foreign key approach, (2) the merged relationship approach, and (3) the cross reference or relationship relation approach. The first approach is the most useful and should be followed unless special conditions exist, as we discuss below.

1. Foreign key approach: Choose one of the relations—S, say—and include as a foreign key in S the primary key of T. It is better to choose an entity type with total participation in R in the role of S. Include all the simple attributes (or simple components of composite attributes) of the 1:1 relationship type R as attributes of S.

If only 2 percent of employees manage a department, then 98 percent of the foreign keys would be NULL in this case. Another possibility is to have foreign keys in both relations S and T redundantly, but this creates redundancy and incurs a penalty for consistency maintenance

Step 4: Mapping of Binary 1:N Relationship Types.

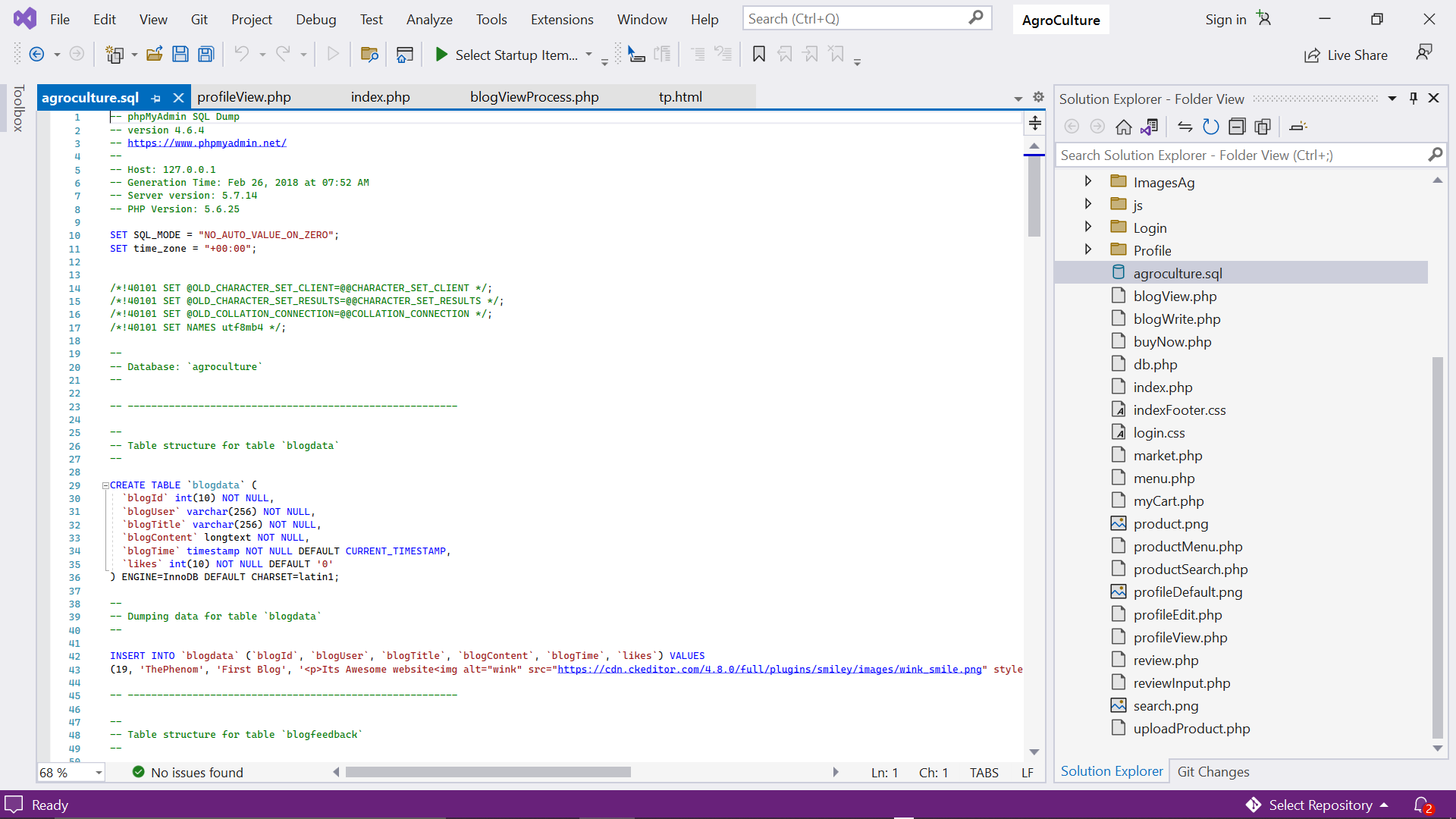
For each regular binary 1:N relationship type R, identify the relation S that represents the participating entity type at the N-side of the relationship type. Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R; we do this because each entity instance on the N-side is related to at most one entity instance on the 1-side of the relationship type. Include any simple attributes (or simple components of composite attributes) of the 1:N relationship type as attributes of S.. These foreign keys are shown in Figure 9.2. An alternative approach is to use the relationship relation (cross- reference) option as in the third option for binary 1:1 relationships. We create a separate relation R whose attributes are the primary keys of S and T, which will also be foreign keys to S and T. The primary key of R is the same as the primary key of S. This option can be used if few tuples in S participate in the relationship to avoid excessive NULL values in the foreign key

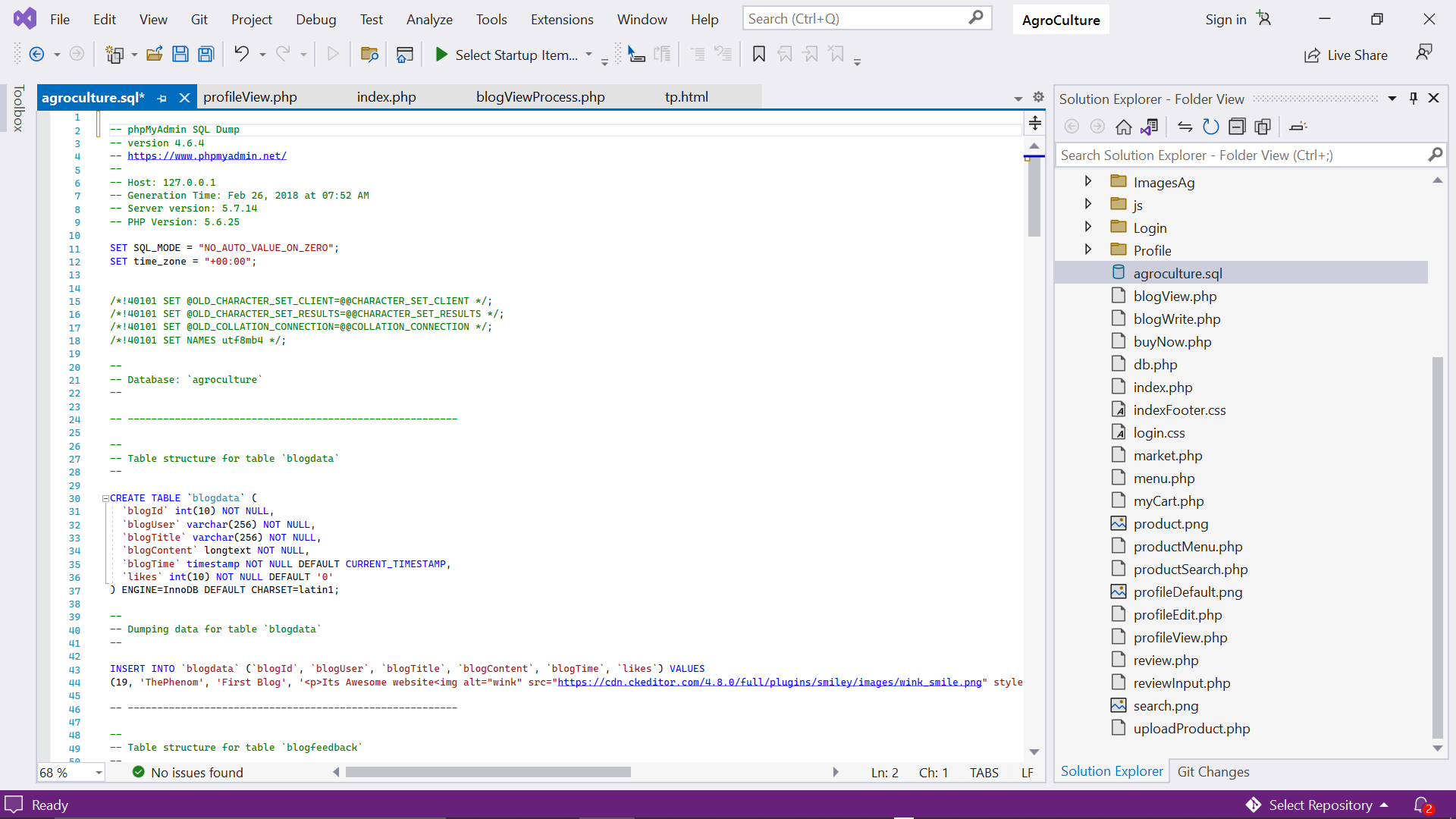
Step 5: Mapping of Binary M:N Relationship Types. For each binary M:N relationship type R, create a new relation S to represent R. Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; their combination will form the primary key of S. Also include any simple attributes of the M:N relationship type (or simple components of composite attributes) as attributes of S. Notice that we cannot represent

an M:N relationship type by a single foreign key attribute in one of the participating relations (as we did for 1:1 or 1:N relationship types) because of the M:N cardinality ratio; we must create a separate relationship relation S. In our project, we map the M:N relationship type WORKS\_ON from Figure 9.1 by creating the relation WORKS\_ON in Figure 9.2. We include the primary keys of the PROJECT and EMPLOYEE relations as foreign keys in WORKS\_ON and rename them Pno and Essn, respectively. We also include an attribute Hours in WORKS\_ON to represent the Hours attribute of the relationship type. The primary key of the WORKS\_ON relation is the combination of the foreign key attributes {Essn, Pno}. This relationship relation is shown in Figure 9.3(c).

Step 6: Mapping of Multivalued Attributes. For each multivalued attribute A, create a new relation R. This relation R will include an attribute corresponding to A, plus the primary key attribute K—as a foreign key in R—of the relation that represents the entity type or relationship type that has A as a multivalued attribute. The primary key of R is the combination of A and K. If the multivalued attribute is composite, we include its simple components.

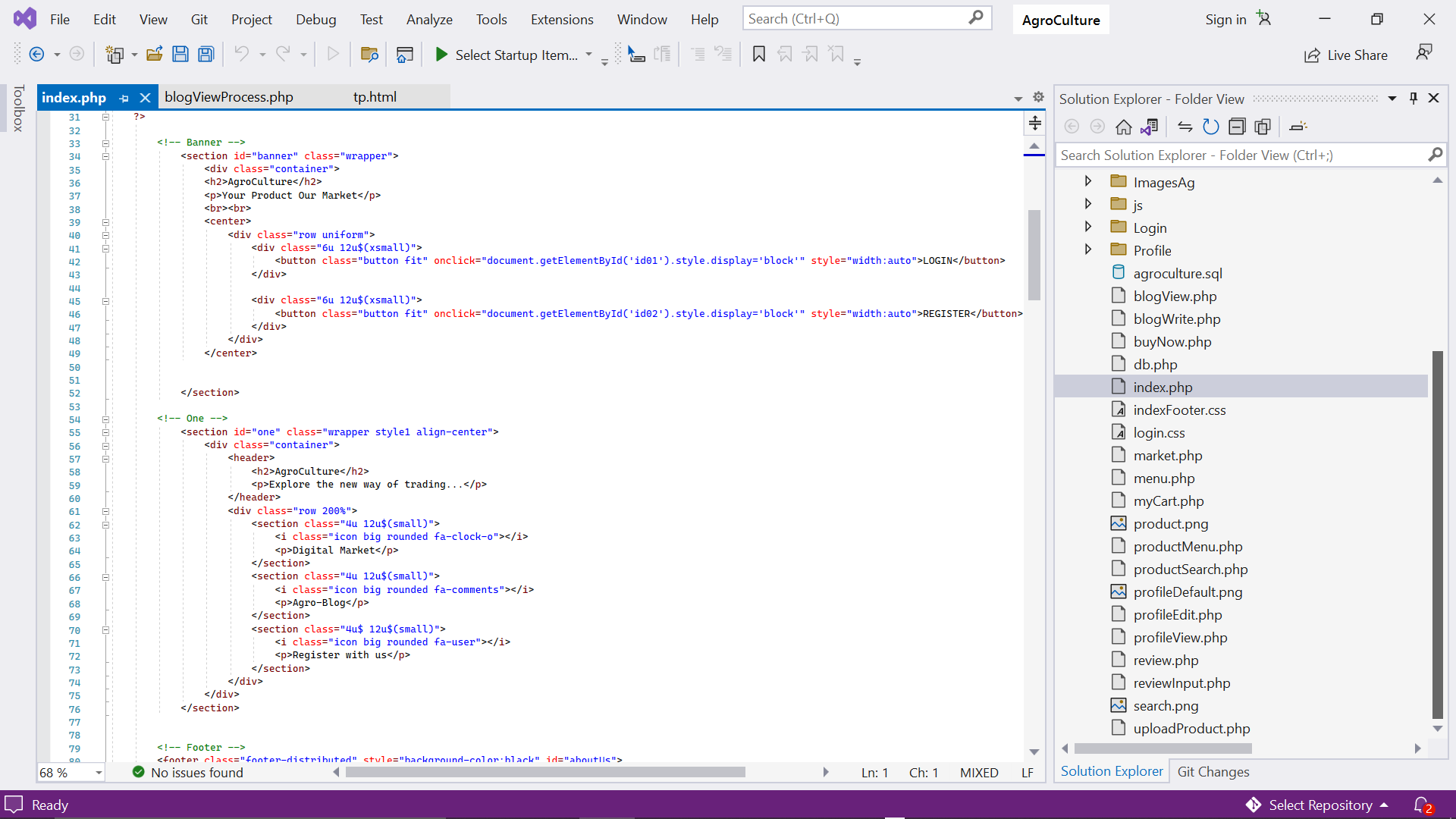
## Relational Database Design

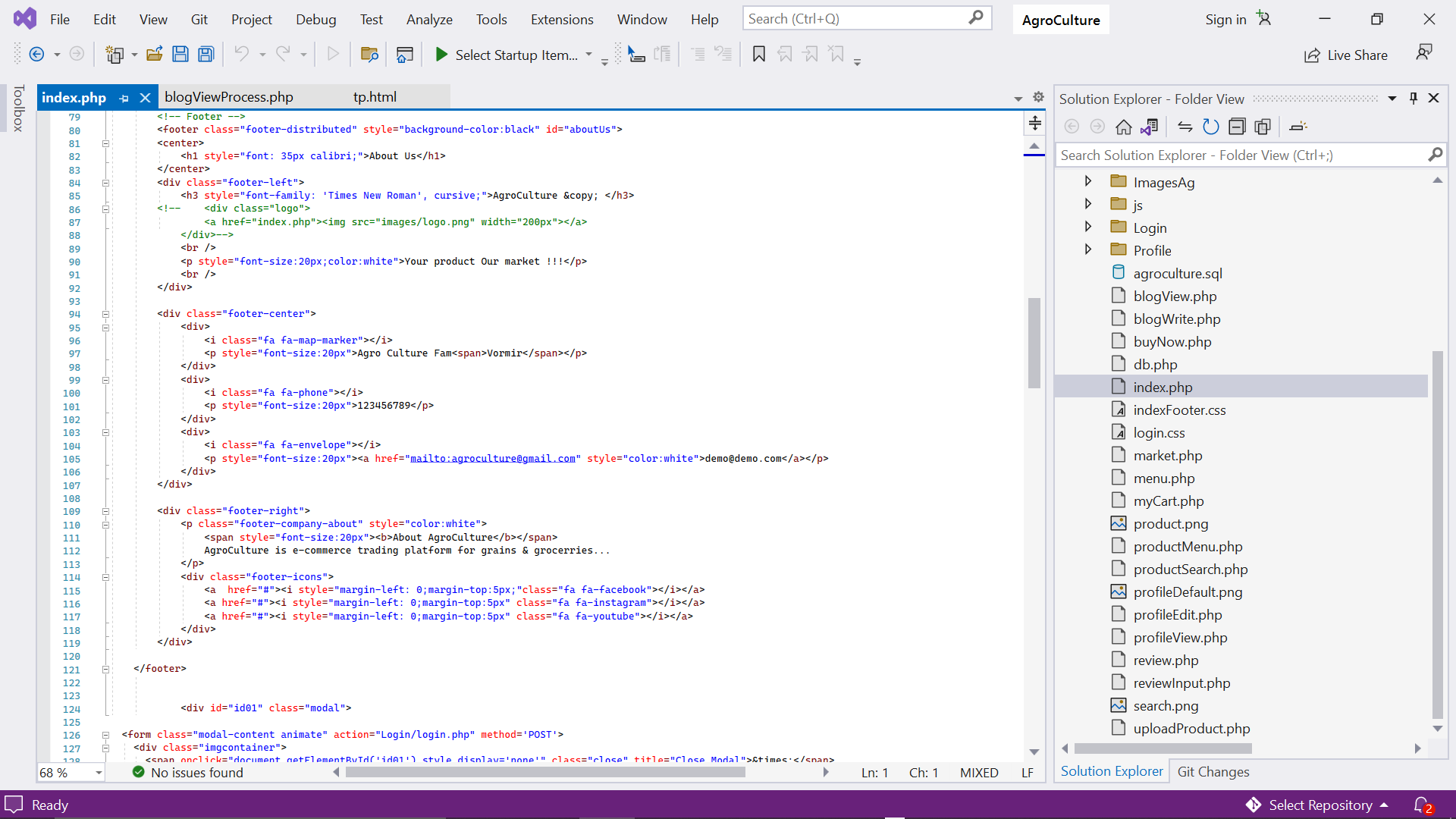


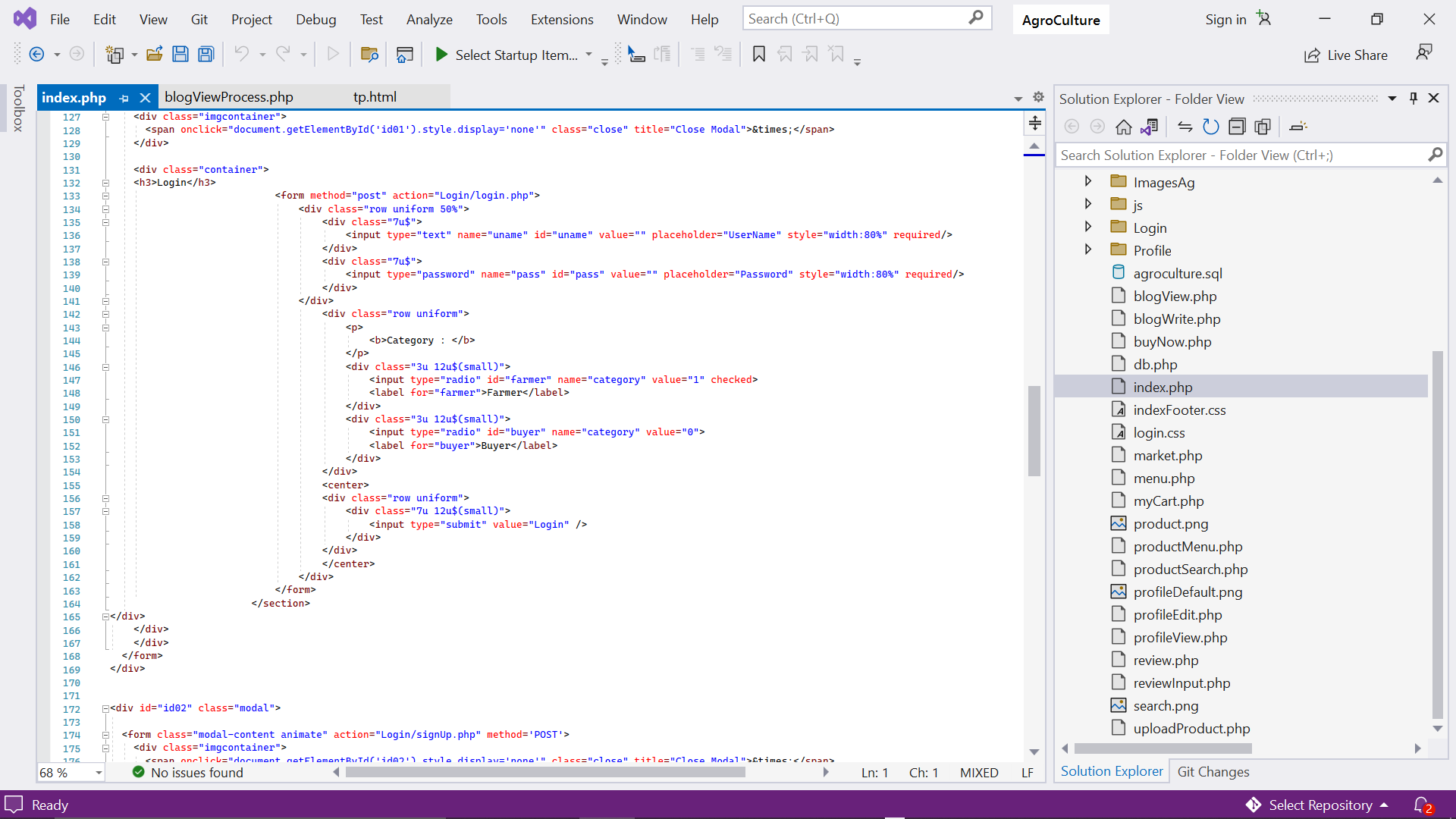


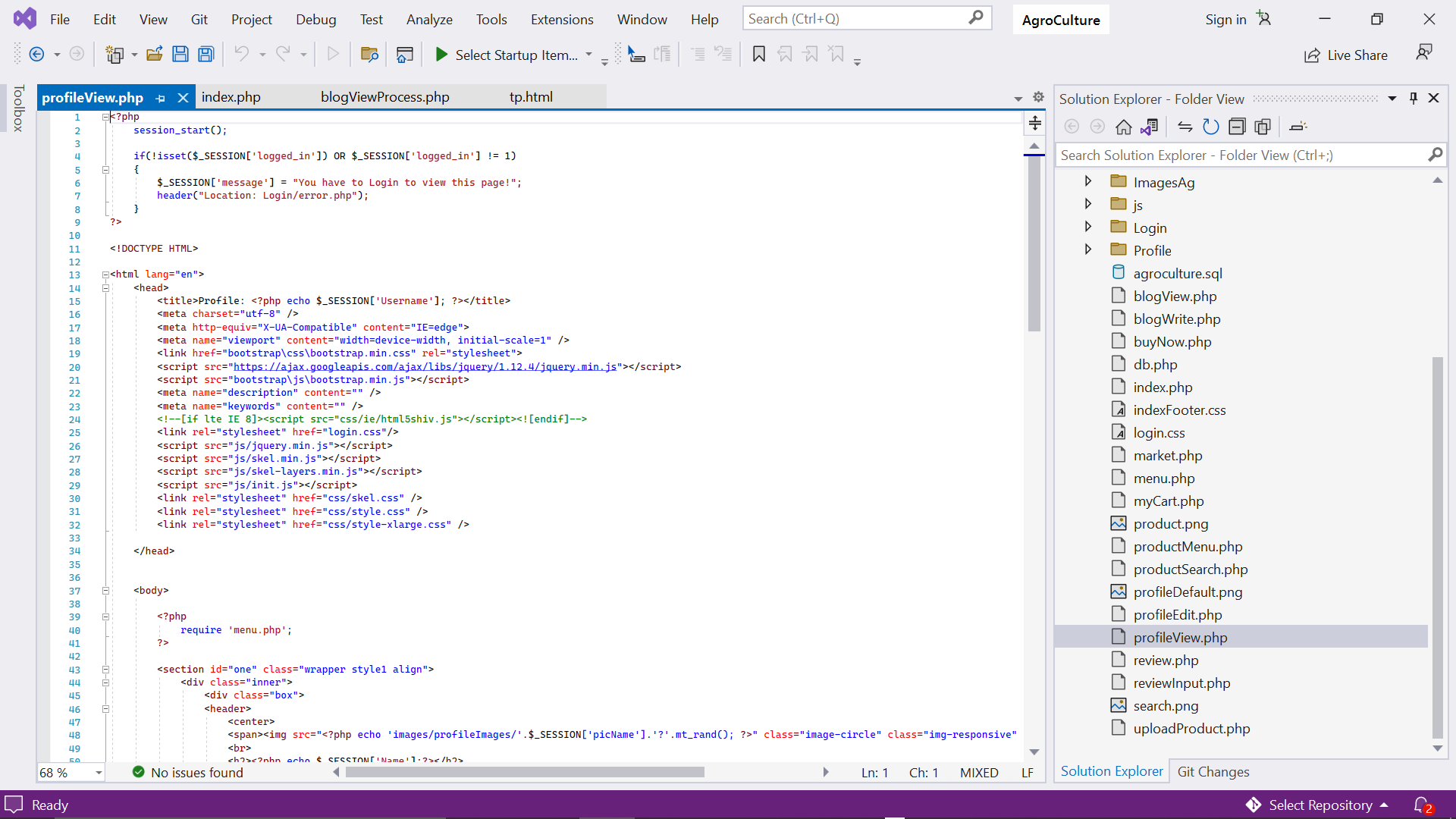
## Creating database using MySQL

An SQL schema is identified by a schema name, and includes an authorization identifier to indicate the user or account who owns the schema, as well as descriptors for each element in the schema. Schema elements include tables, constraints, views, domains, and other constructs (such as authorization grants) that describe the schema. A schema is created via the CREATE SCHEMA statement, which can include all the schema elements’ definitions. Alternatively, the schema can be assigned a name and authorization identifier, and the elements can be defined later. The CREATE TABLE command is used to specify a new relation by giving it a name and specifying its attributes and initial constraints. The attributes are specified first, and each attribute is given a name, a data type to specify its domain of values, and any attribute constraints, such as NOT NULL. The key, entity integrity, and referential integrity constraints can be specified within the CREATE TABLE statement after the attributes are declared, or they can be added later using the ALTER TABLE command.

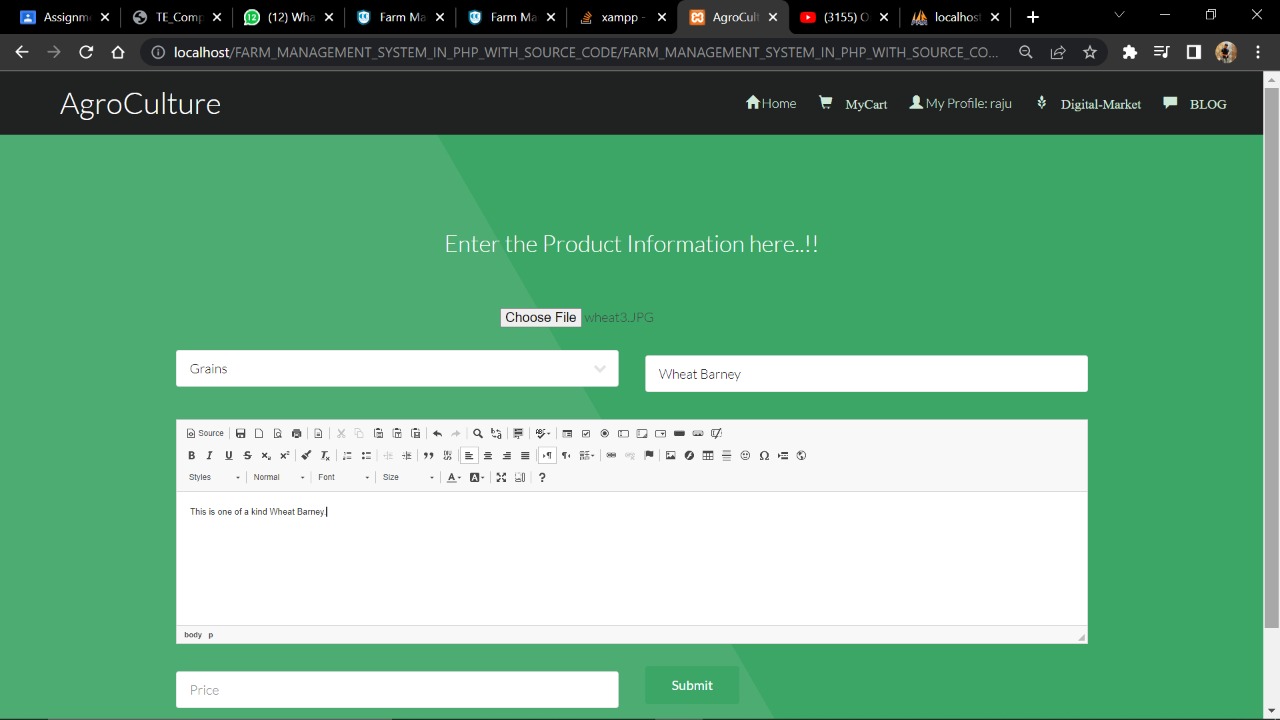


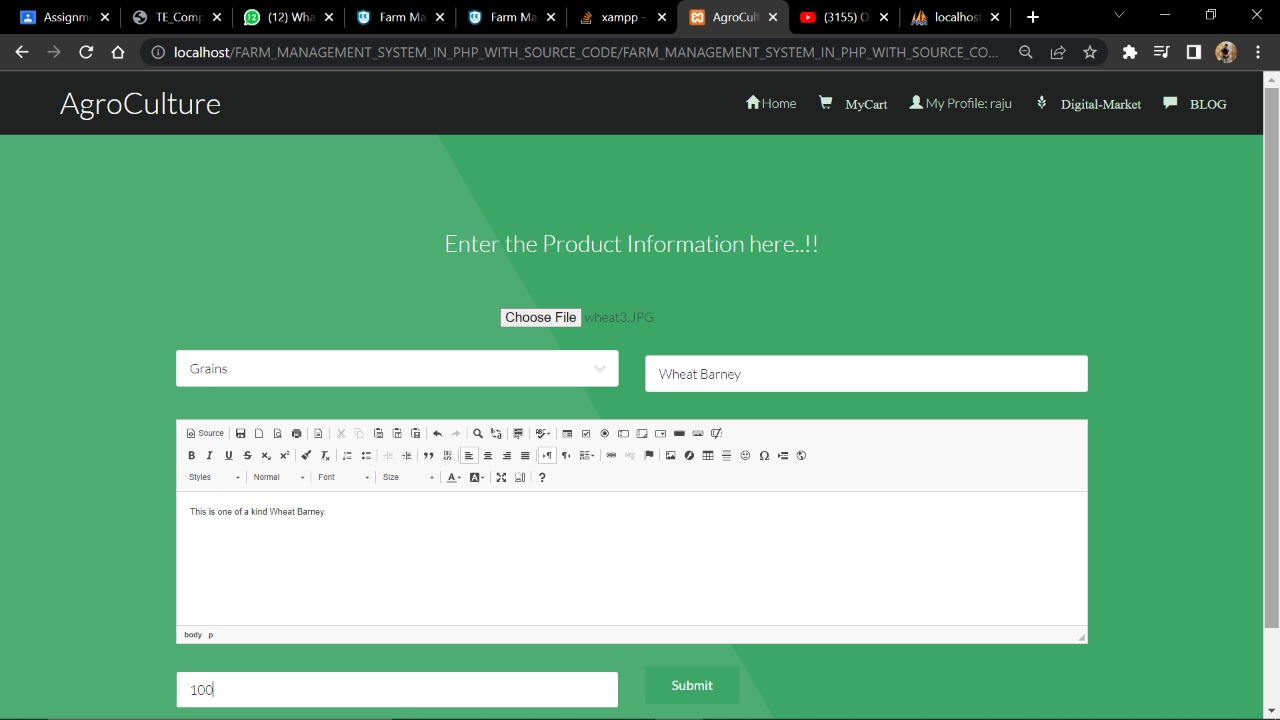
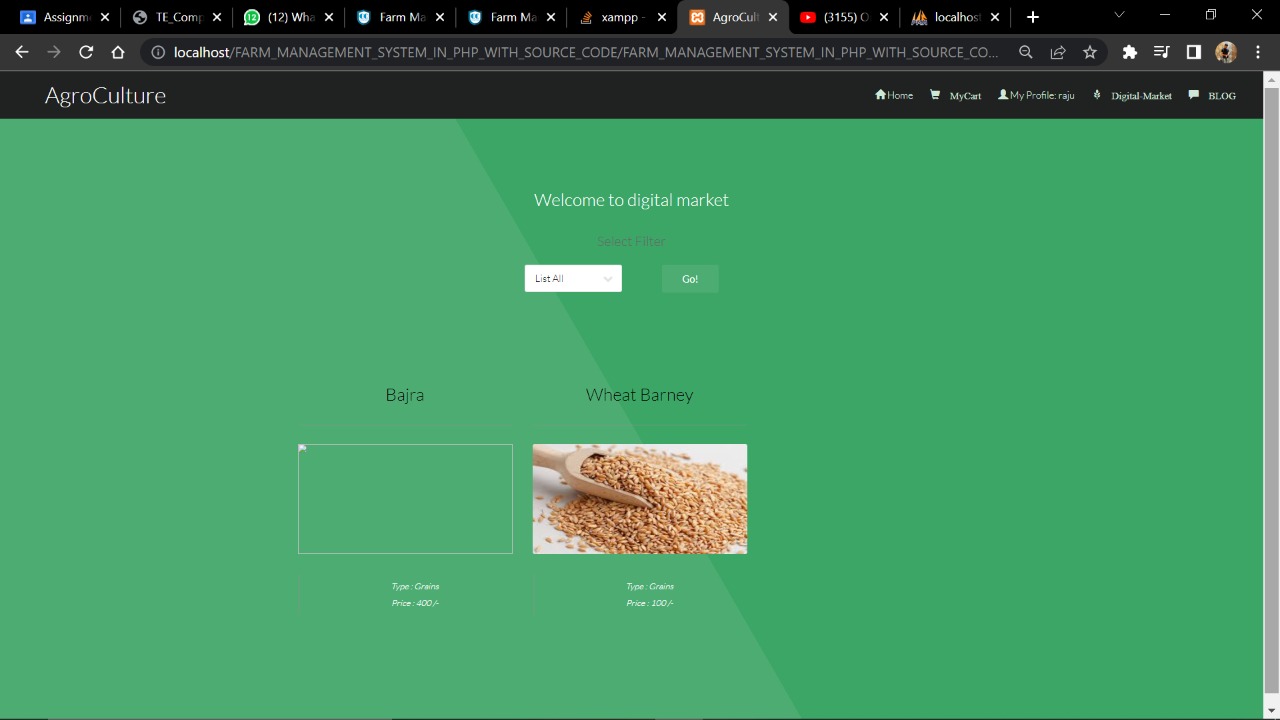


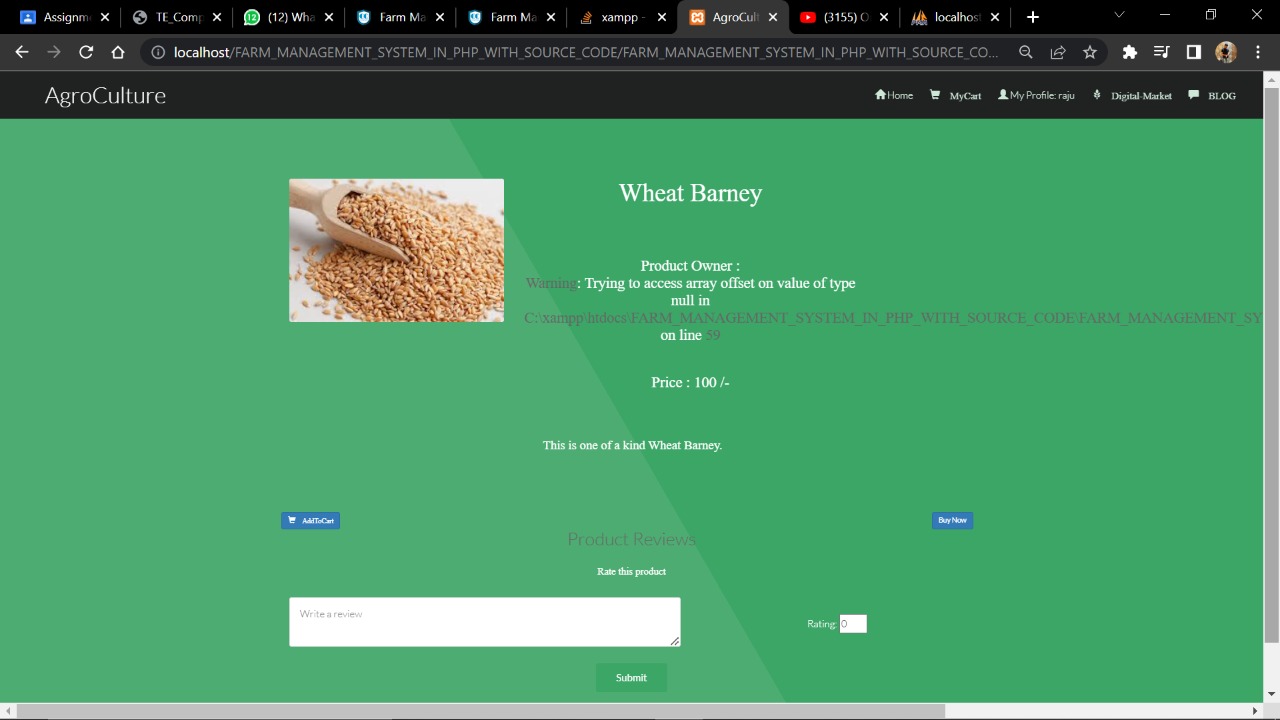
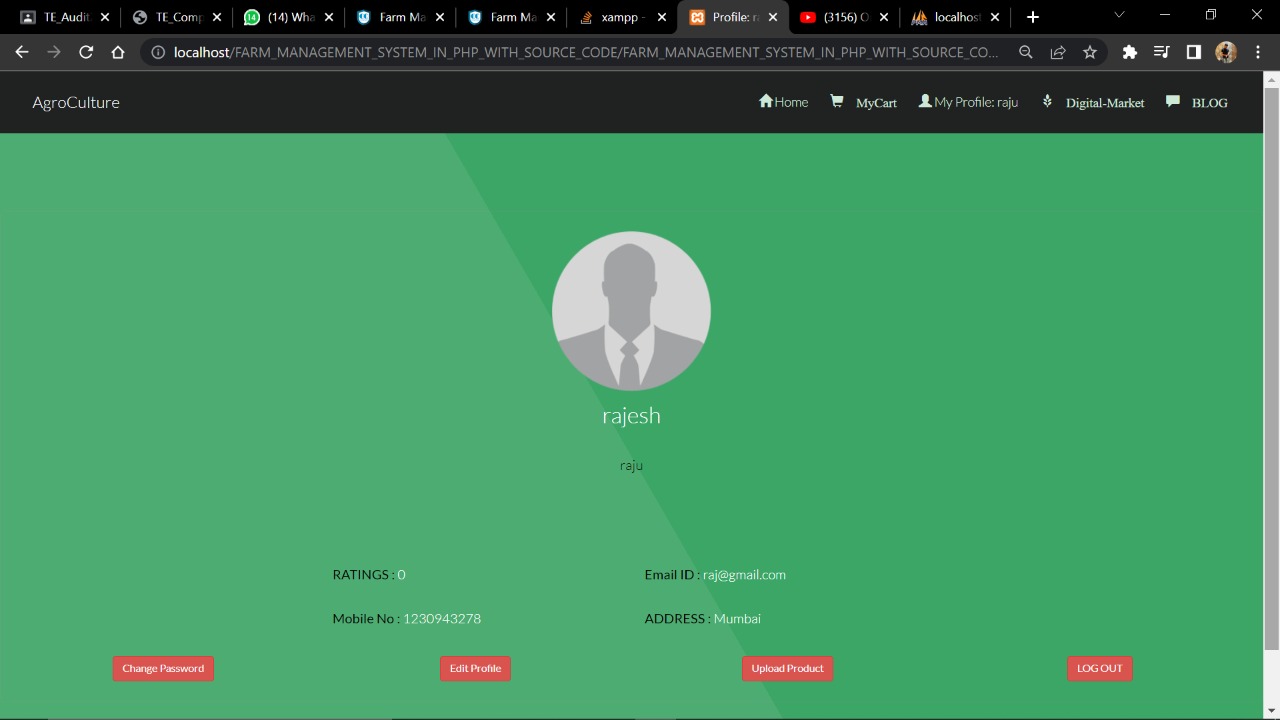




* 1. **Implementation**







## Conclusion

During the course of this project, we learnt creating a database, the rules to construct a good ER diagram, How to come up with relational schema mapping from the ER diagram, deriving the functional dependencies and how to normalize the relational schema.

In comparison to previous experiences, when every activity related to the vehicle rental business was restricted to a physical place alone, the car rental industry has emerged with new delicacies. Even if the physical location has not been completely eliminated, the internet's power has altered the nature of functions and how these tasks are accomplished. Customers may now book vehicles online, rent automobiles online, and have the car delivered to their home if they are a registered member, or they can travel to the office to pick up the car.

## References.

1. Silberschatz A., Korth H., Sudarshan S. “Database System Concepts”, 6th edition, Tata McGraw Hill Publishers
2. Dr. P. S. Deshpande, “SQL and PL/SQL for Oracle 10g Black Book”, DreamTech Wikipedia, Online. 'Use Case Diagram'. N.p., 2015. Web. 9 June 2015.
3. Wikipedia, Online. 'Activity Diagram'. N.p., 2015. Web. 9 June 2015.
4. Tutorialspoint.com, Online. 'UML - Activity Diagrams'. N.p., 2015. Web. 9 June 2015. [5]. Wikipedia, online. 'Swim Lane'. N.p., 2015. Web. 9 June 2015.

[6]. Mindtools.com, Online. 'Swim Lane/Rummler-Brache Diagrams: Mapping and Improving Processes in Your Organization'. N.p., 2015. Web. 9 June 2015.

1. Laudon, Kenneth C, and Jane Price Laudon. Management Information Systems. Upper Saddle River, NJ: Prentice Hall, 2000. Print.
2. Menkus, Belden. 'Car Rental Chain Former Owners Charged With Computer Frauds'. Computer Fraud & Security Bulletin 1993.3 (1993): 3-4. Web.
3. Li, Zhang. 'Design And Realization Of Car Rental Management System Based On AJAX+SSH'. Information Technology J. 12.14 (2013): 2756-2761. Web.