Jiyun Noh

Nicole Bade

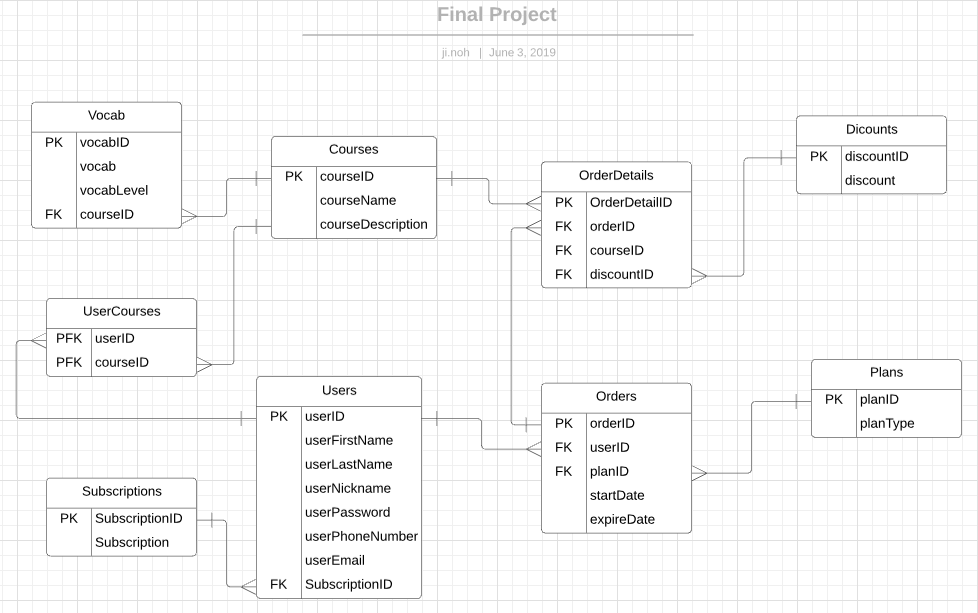
ITC 220 – Final Project

June 6, 2019

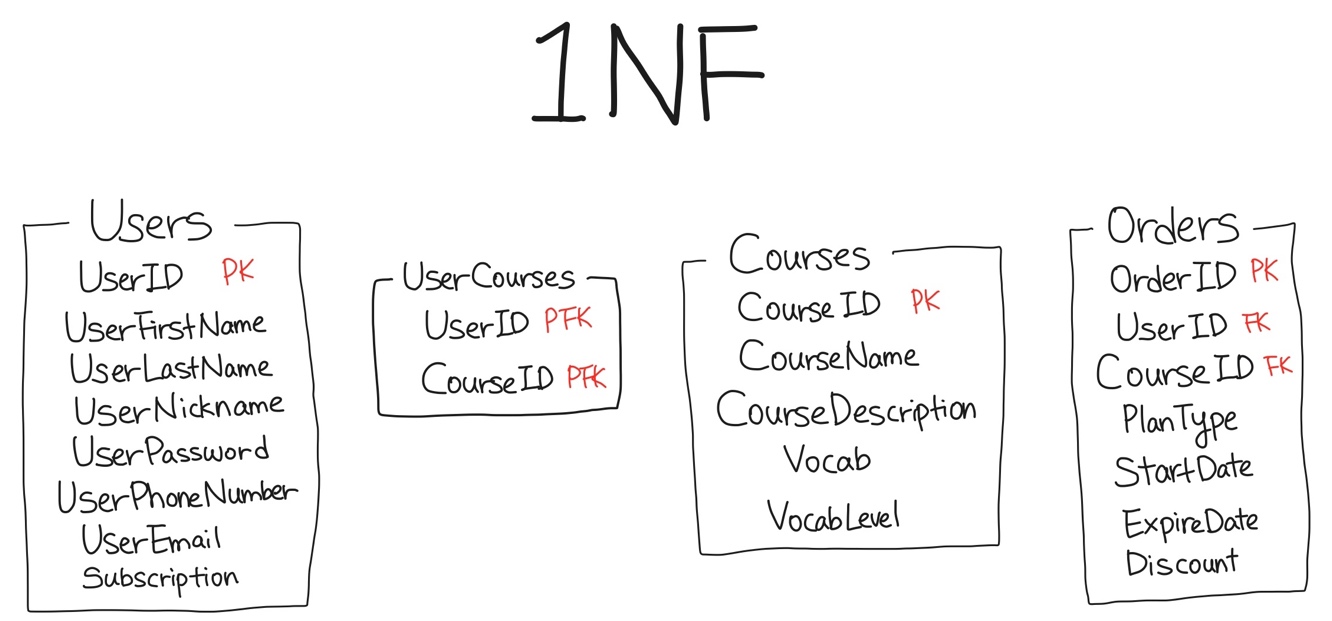
English Vocabulary Application Database

<Design>

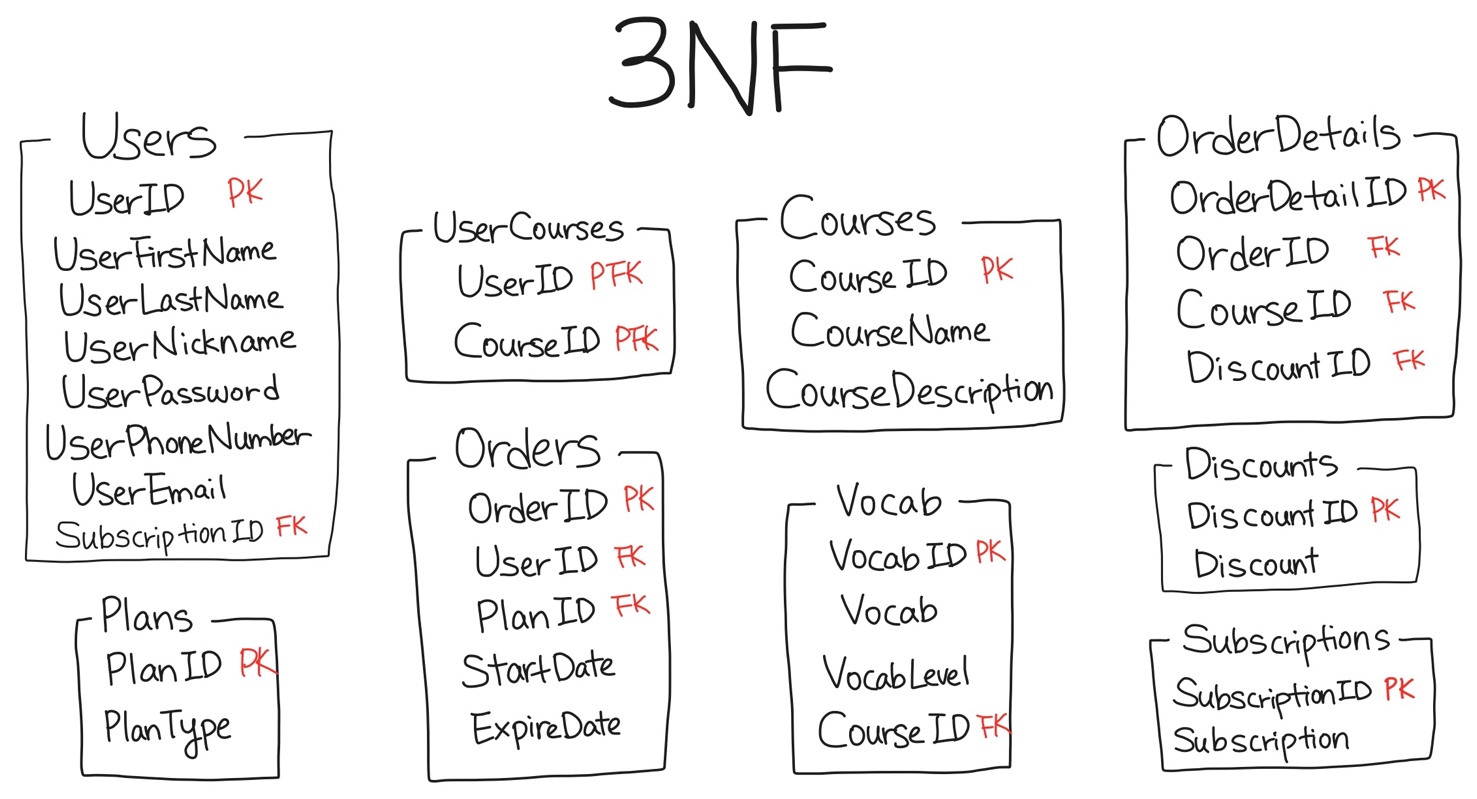
1. Mission Statement: To track users, English vocabularies, plans and courses that users take, who study English for an English test.
2. Mission Objectives:
   * Display users’ names
   * Display users’ email addresses
   * Display users’ nicknames
   * Show which plans each user chooses
   * Show which courses each user takes
   * Show the different levels of vocabularies depending on courses
3. Database Requirements:
   * Must track users’ orders
   * Must track users’ emails
   * Must track users’ subscriptions
   * Must track vocabularies depending on each course
   * Must track which plans and courses each customer takes
4. Business Rules:
   * Free trial for a week is provided for every user.
   * A longer plan should be cheaper per a month than a shorter plan.
   * A user who purchases a life-long plan can access to every course unlimitedly.
   * A user can get a 5% discount for the first month if the user subscripts the notification of this application.
5. Research: There are some other databases that match my database. They have similar tables and relationships that mine has. I learned how to make a linking table in a many-many relationship, and it does not require a new primary key. For example, a user-course table, which links a user table and a course table in many-many relationship, includes userID from user table and courseID from course table. So one user can take many courses and also one course can have many students.
6. Stakeholders: Me(App Developer), and Users
7. Preliminary field list: userFirstName, userLastName, userNickname, userPassword, userPhoneNumber, userEmail, courseName, courseDescription, planType, vocab, vocabLevel, discount, startDate, expireDate, subscription
8. Preliminary entity list: Users, Courses, Plans, Vocab, Discounts, Orders, OrderDetails, UserCourses, Subscriptions
9. Identify primary keys: UserID(Users), CourseID(Courses), PlanID(Plan), OrderID(Orders), VocabID(Vocab), OrderDetailID(OrderDetails), DiscountID(Discounts), SubscriptionID(Subscriptions)
10. Create relationships:



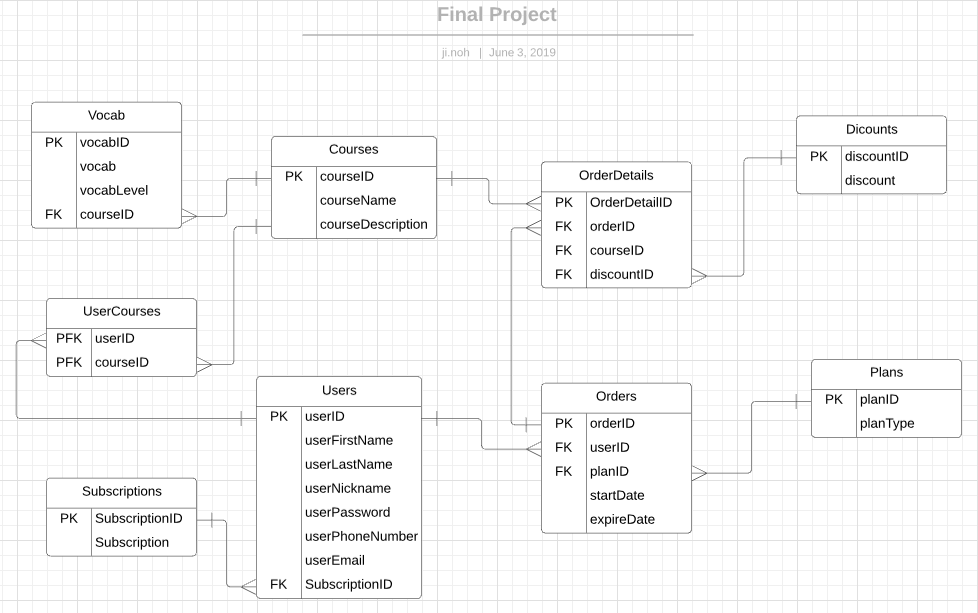
1. ERD through 1NF, 2NF, 3NF:



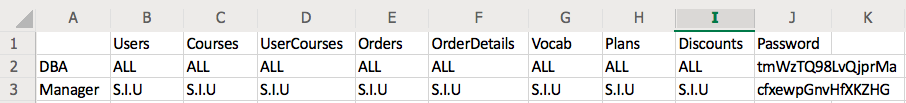
2NF- There is a composite key in my DB but it does not have a partial dependency.



1. Define data views:
   * SELECT UserNickName, UserPassword FROM Users;
   * SELECT \* FROM Courses WHERE CourseName=”TOEFL”;
   * SELECT OrderDetails.OrderDetailID, Orders.StartDate, Orders.ExpireDate FROM OrderDetails INNER JOIN Orders ON OrderDetails.OrderID = Orders.OrderID
2. Review data integrity:
   * Table-Level: I reviewed each table to make certain that I have established table-level integrity.
   * Field-Level: I reviewed that I have defined a set of field specifications for each field.
   * Relationship-Level: I made sure that I have properly established relationship-level integrity.
3. Final ERD:

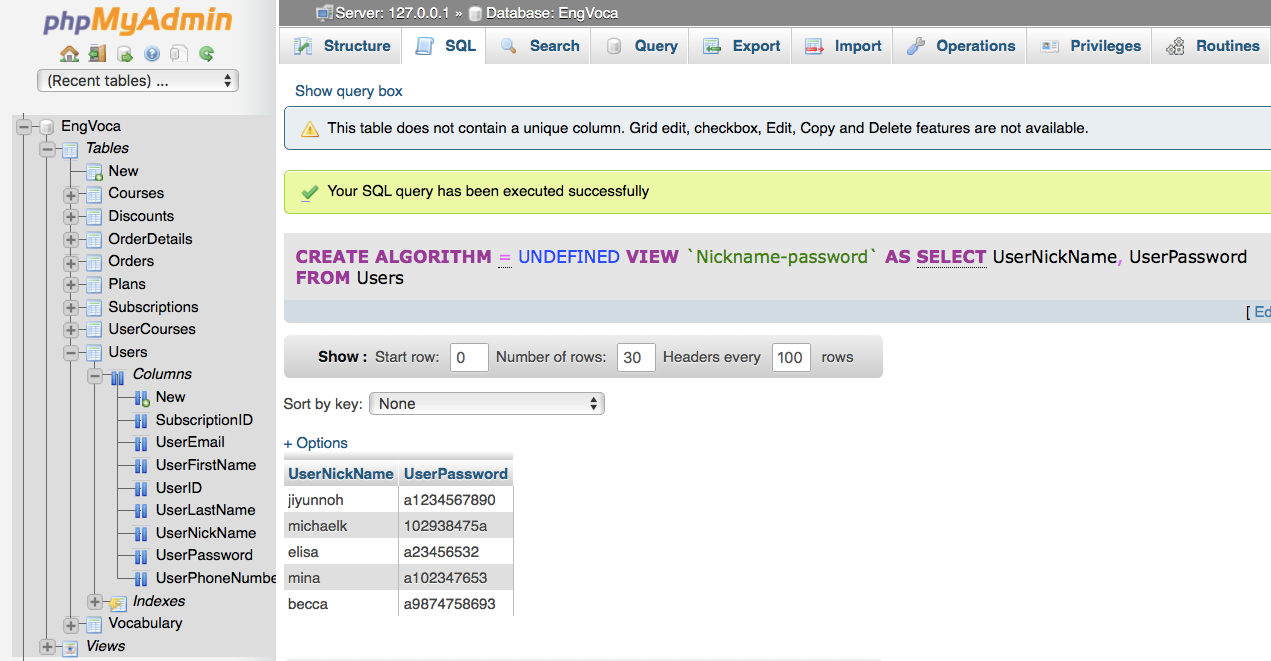


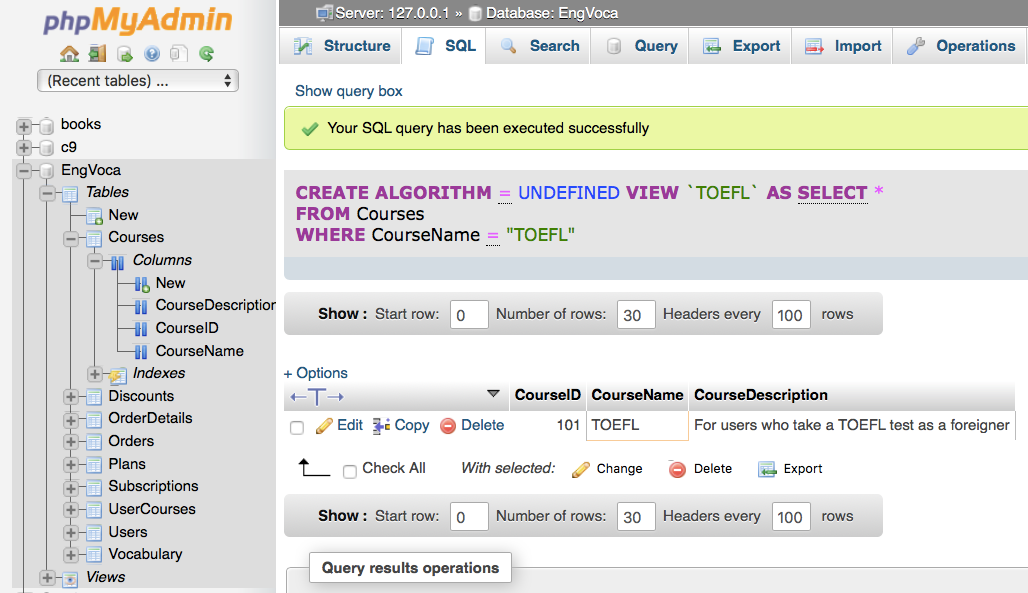
1. Security diagram:

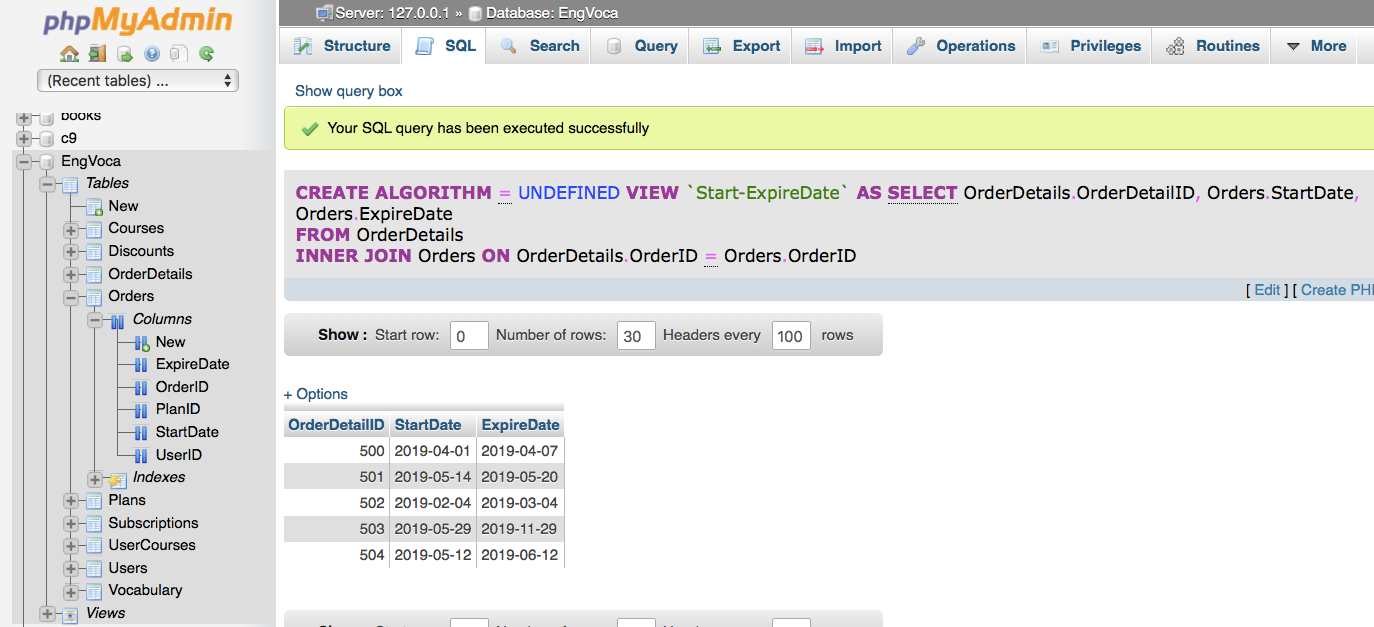


<Implementation>

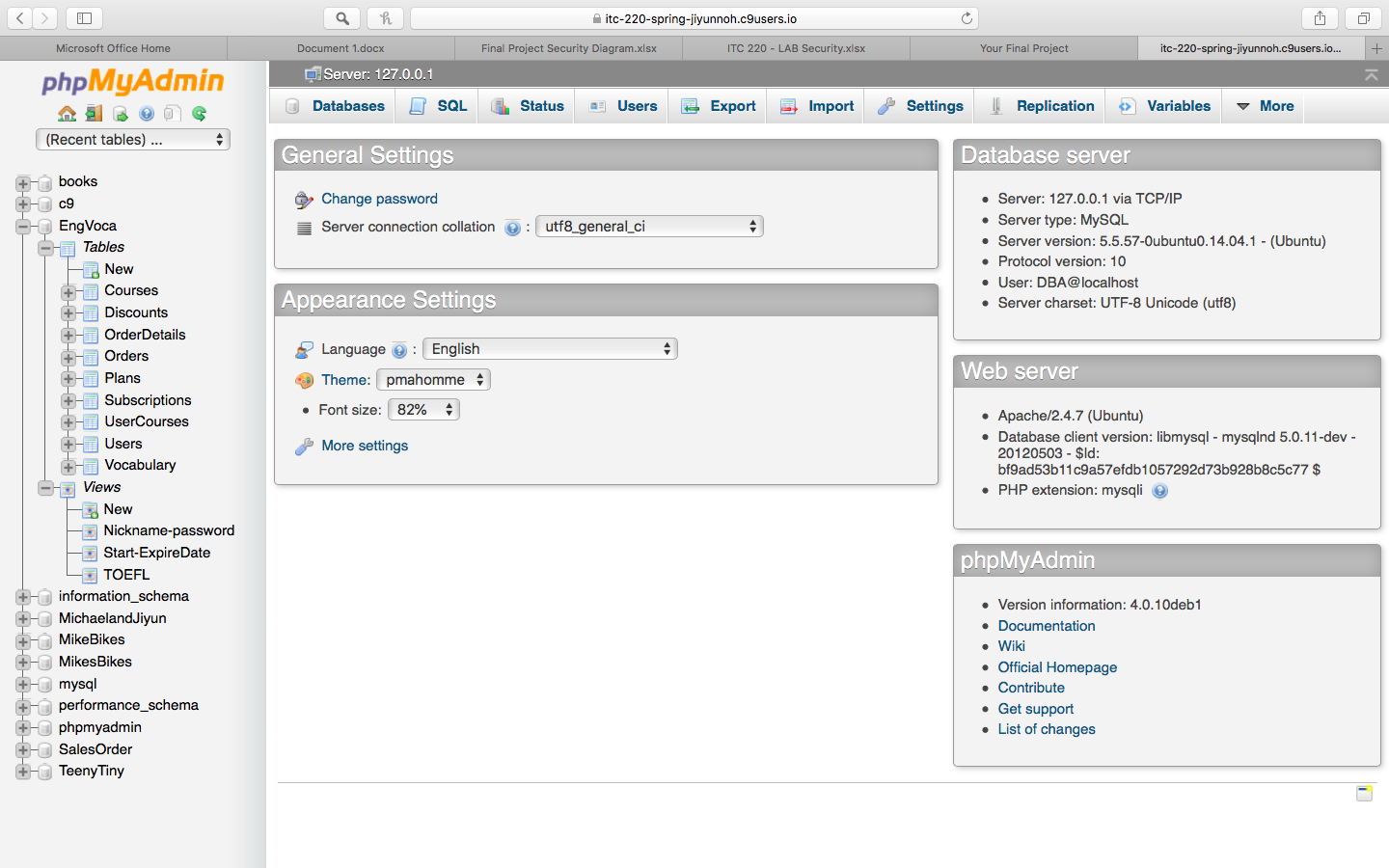
* Use MSSQL server or MySQL to implement your database design
* Enter at least five rows of data in each table for testing purposes
* Create at least one view
* Create users or roles for security (Make sure to have at least one DBA)
* Execute (and screenshot) three queries that answer questions you have about how your DB is working. At least one of these should be a join.

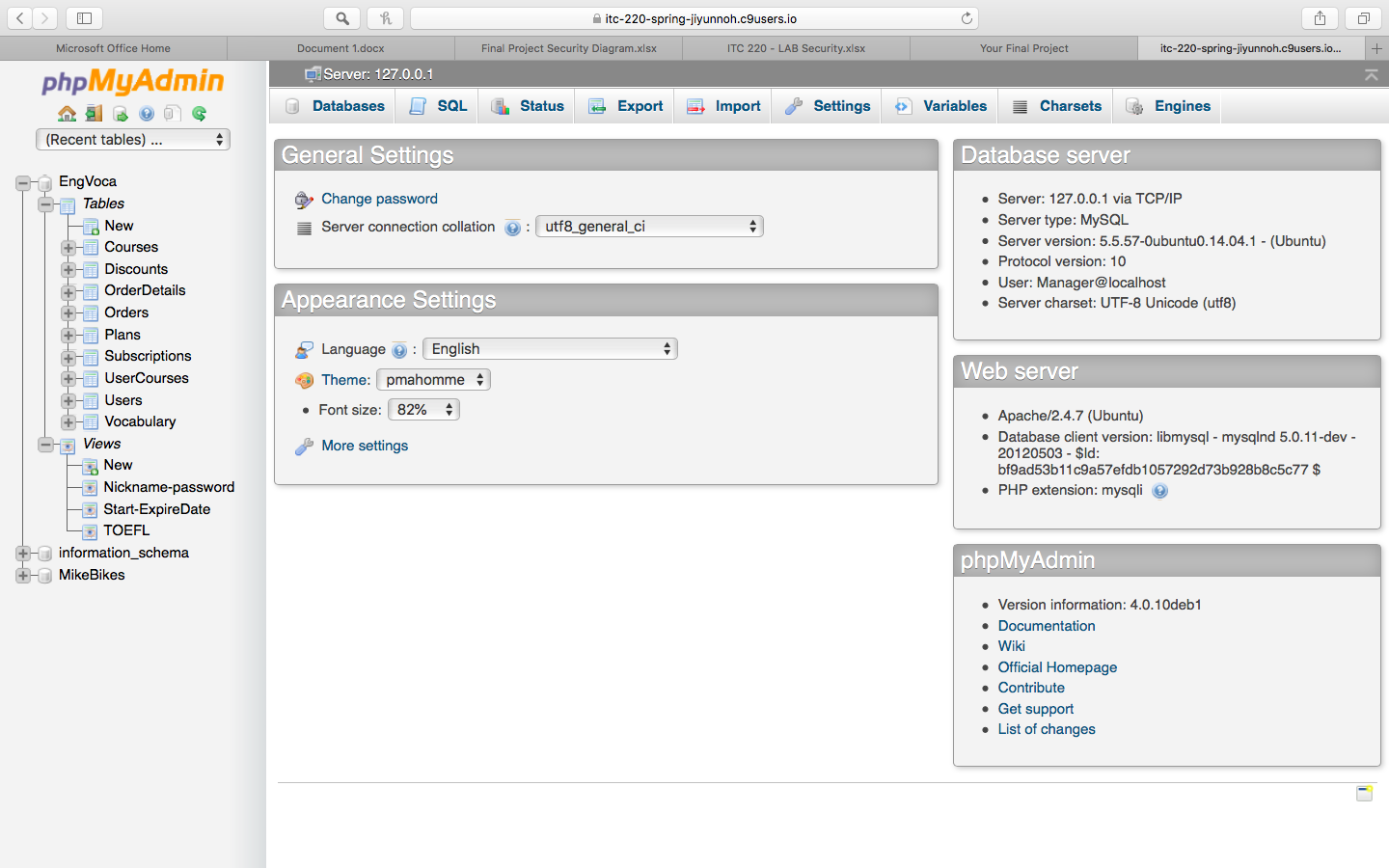






* Test your DB users/roles by logging in as each user/role and performing tasks





* Test your Business rules:
  + SELECT Orders.OrderID, Plans.PlanType FROM Orders INNER JOIN Plans ON Orders.PlanID = Plans.PlanID;
  + SELECT Users.UserID, Subscriptions.Subscription FROM Users INNER JOIN Subscriptions ON Users.SubscriptionID = Subscriptions.SubscriptionID;
* Test your mission objectives:
  + SELECT UserNickName FROM Users;
  + SELECT UserEmail FROM Users;
  + SELECT UserFirstName, UserLastName FROM Users;
* Test your database requirements
  + SELECT \* FROM Orders;
  + SELECT UserID, CourseID FROM UserCourses;
  + SELECT Vocab, VocabLevel, CourseID FROM Vocabulary;