CSE-322

Software Engineering Laboratory

XAMPP & ngrok

Md. Saidul Hoque Anik
anik@cse.uiu.ac.bd

Required Software

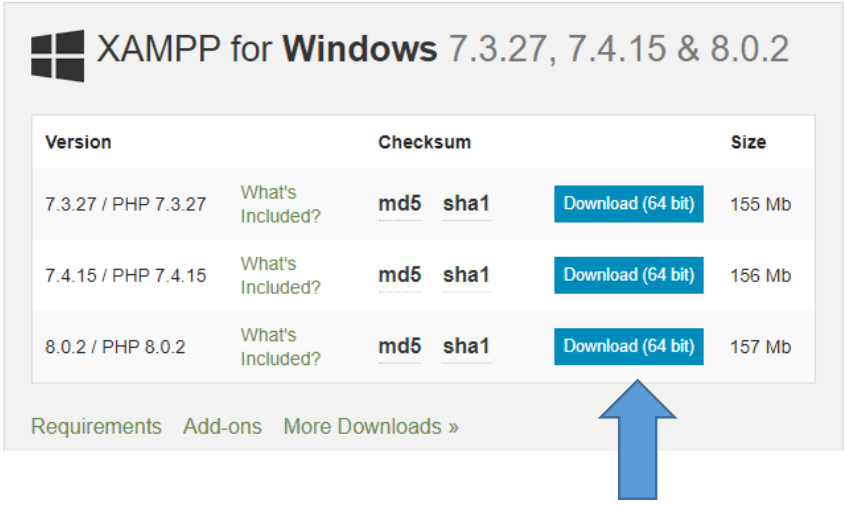
1. XAMPP

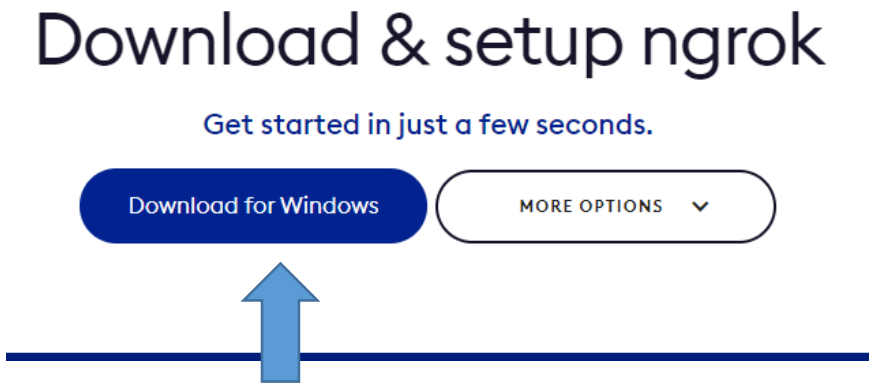
   https://www.apachefriends.org/download.html

2. ngrok

   https://ngrok.com/download

Both are uploaded in [Week 2 folder]



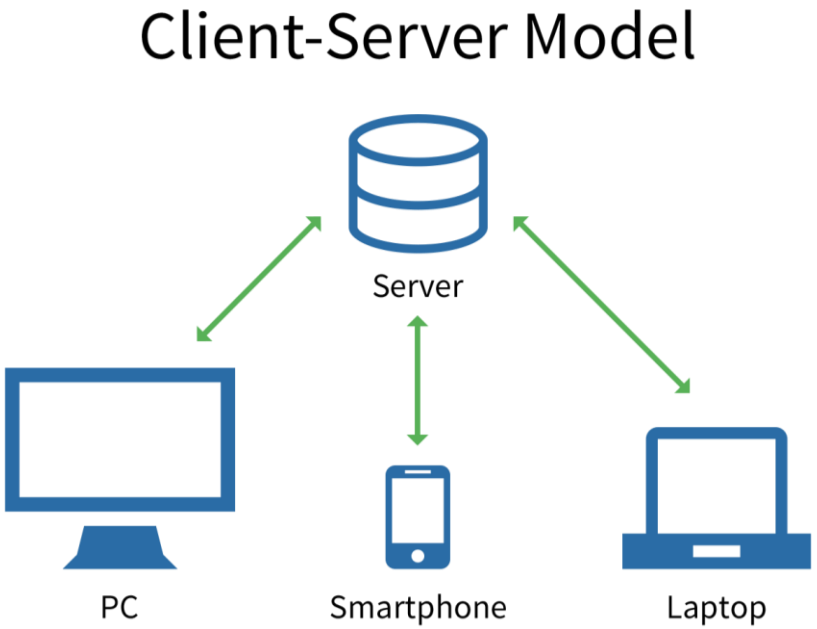


CRUD Operations

The clients will-

1. Create
2. Read
3. Update
4. Delete

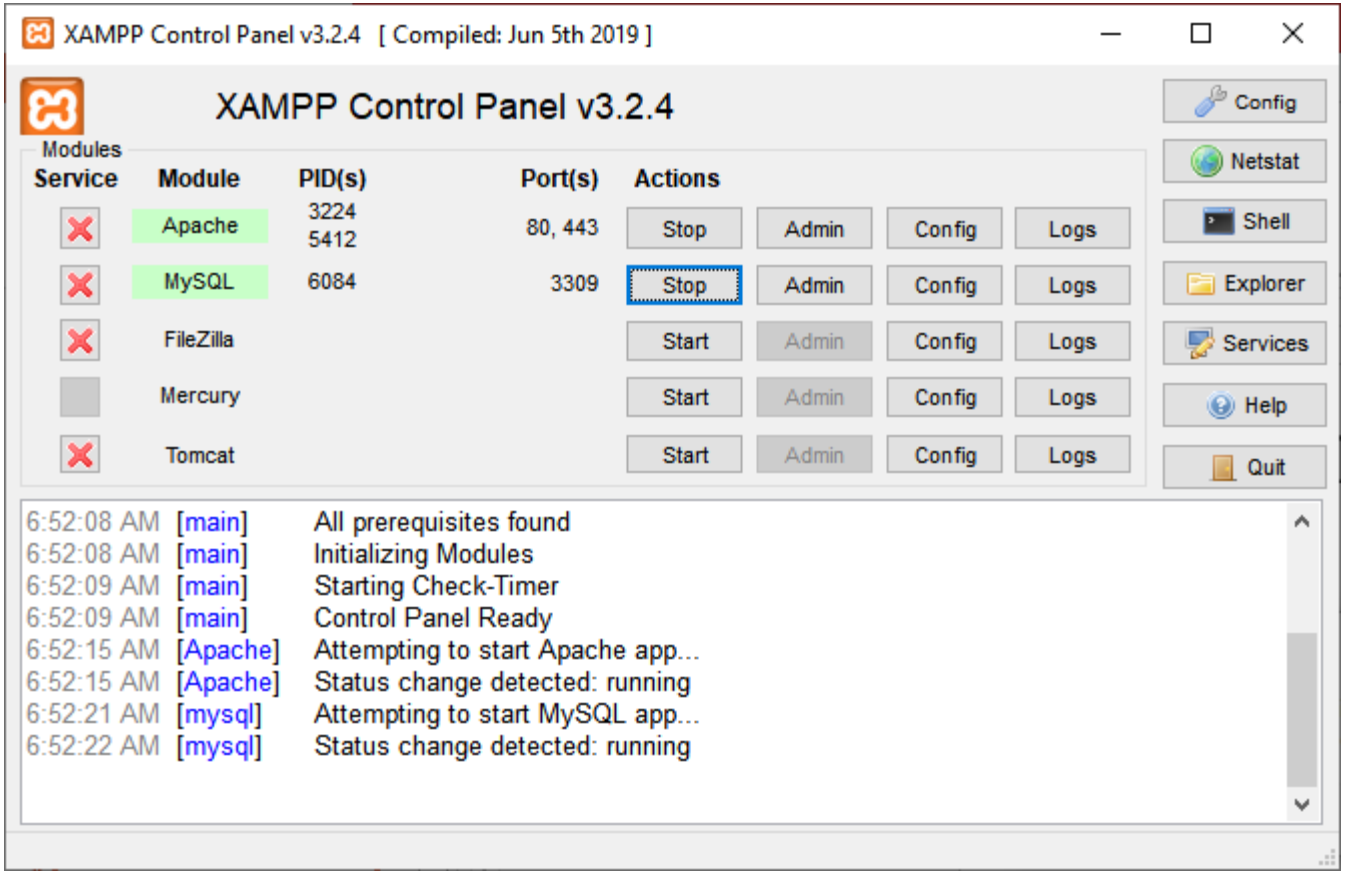
-in the server.



We shall need to create the interface for the operations.

Image Courtesy: techterms.com

Initial Setup



Database Creation

In your browser,

1. Open localhost/phpmyadmin
2. Go to SQL and
3. execute the following query-

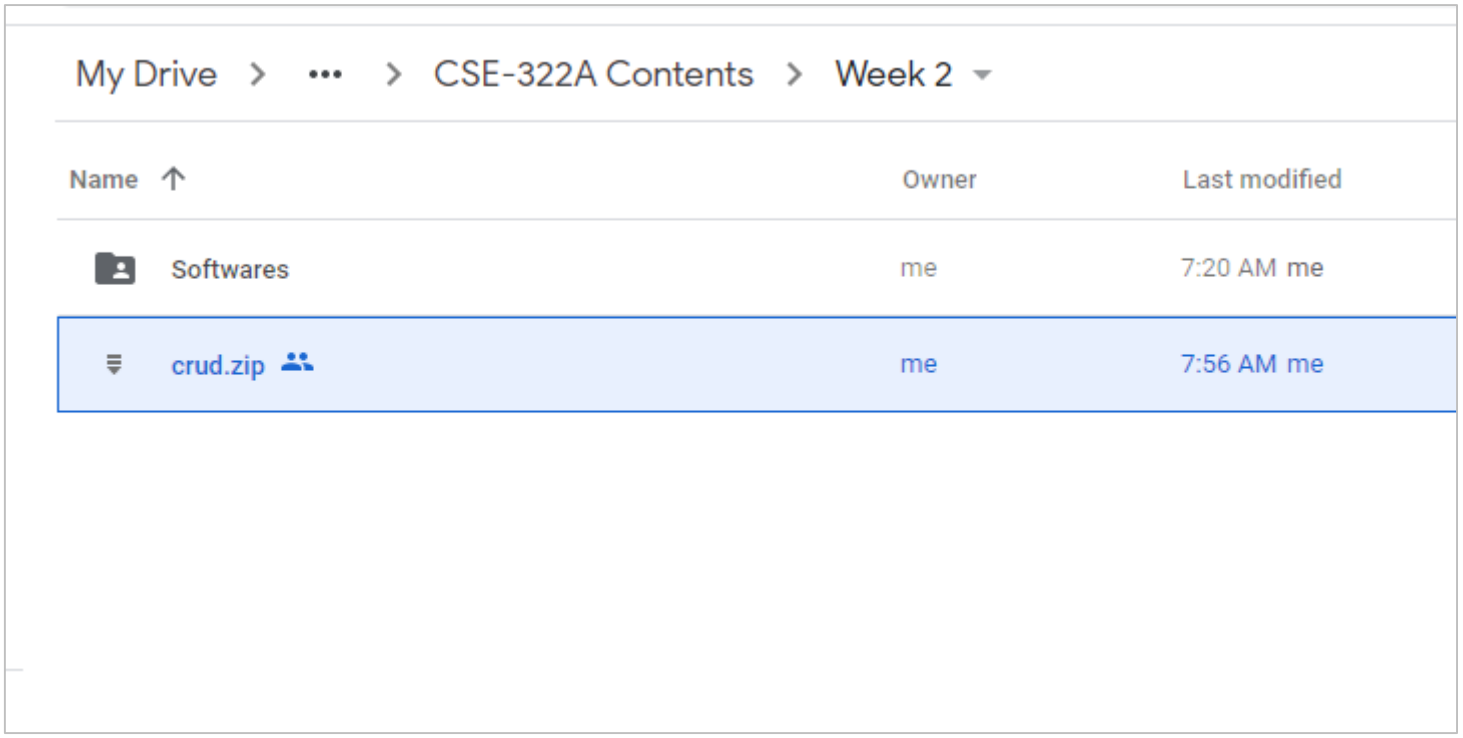
```
create database db0;
use db0;

create table t0 (
    id INT(10) AUTO_INCREMENT,
    f0 char(30),
    f1 char(30),
    PRIMARY KEY(id)
);

insert into t0 values ( '', 'v00', 'v01' );
insert into t0 values ( '', 'v10', 'v11' );
select * from t0;
```

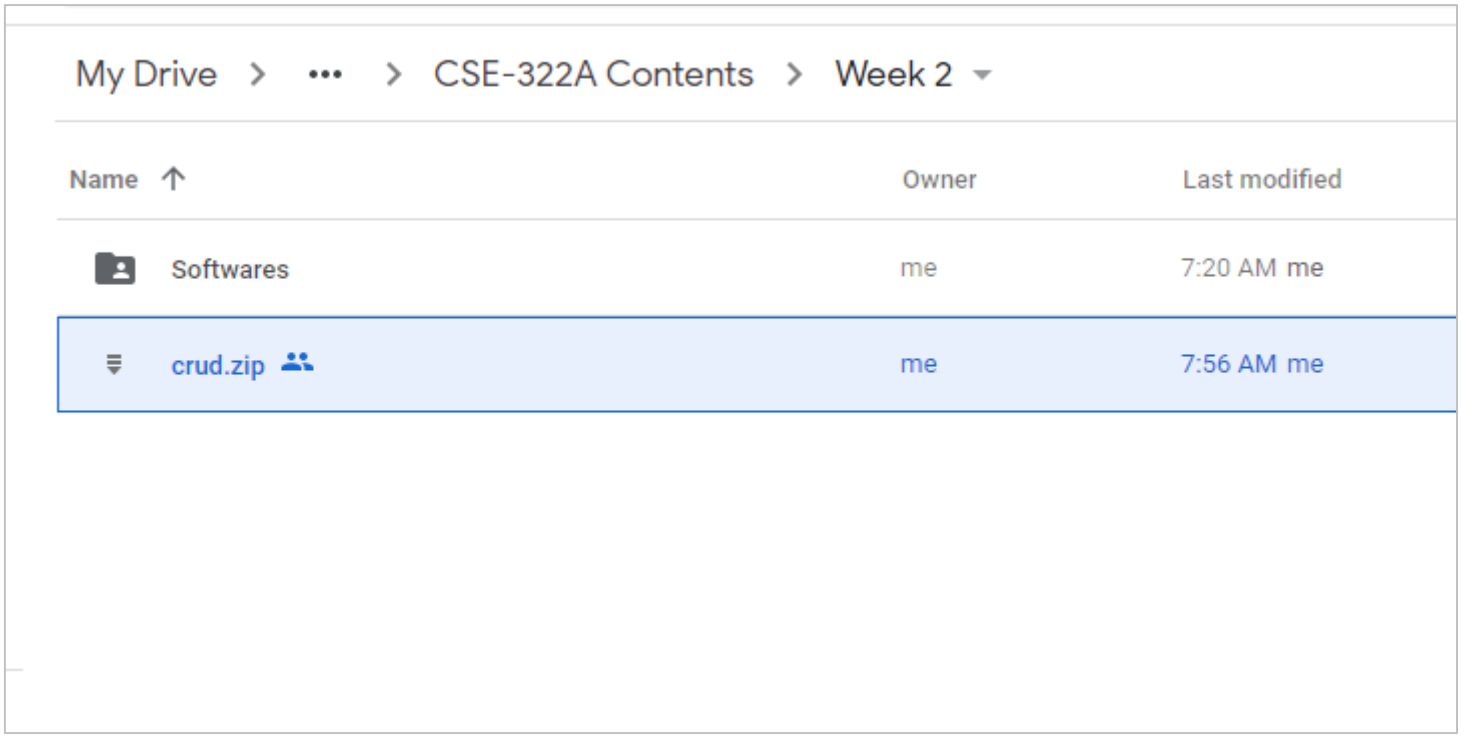
CRUD Template

1. Download crud.zip from Week 2 folder
2. Extract the files in the XAMPP htdocs folder (Usually C:\xampp\htdocs\)



Task

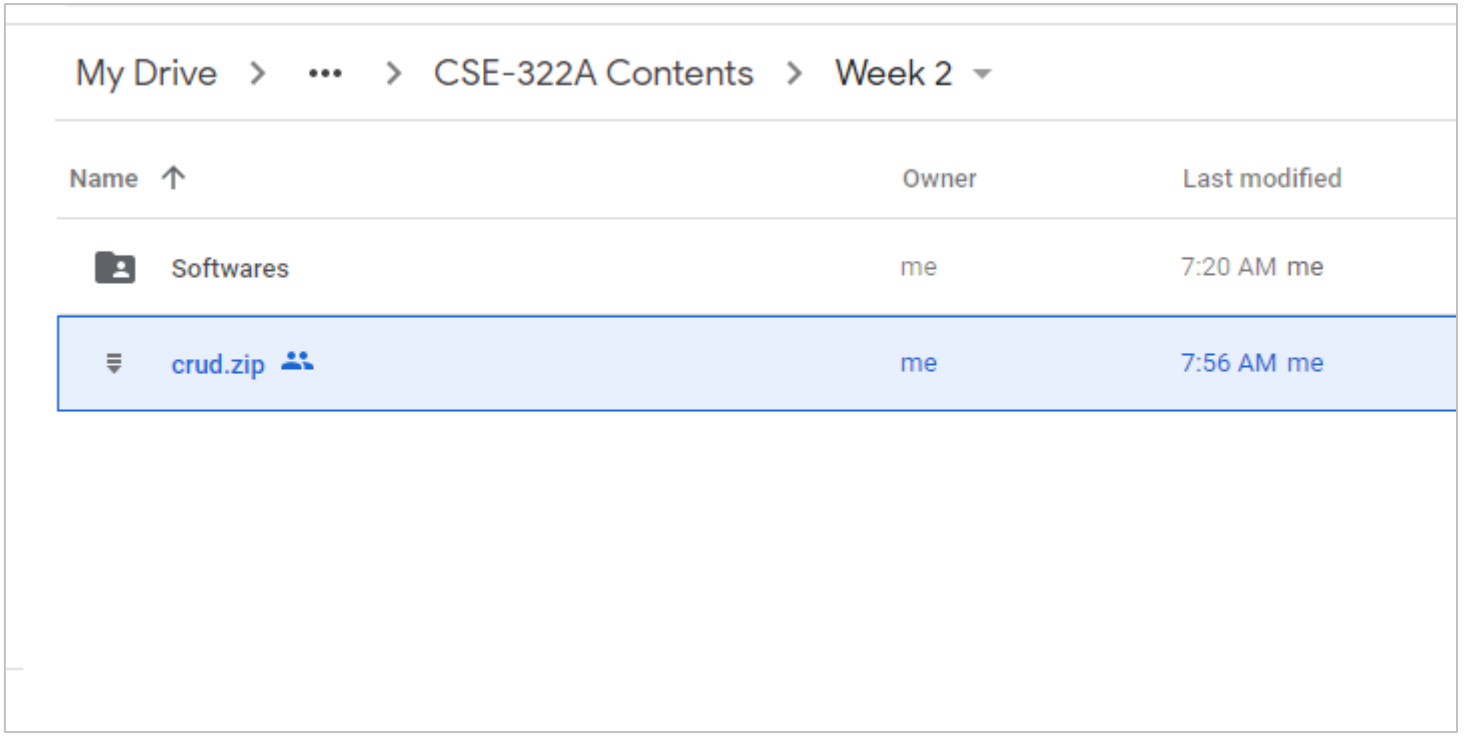
Examine the codes in the crud folder



Publish

You can quickly publish the localhost website to your client.

1. Download ngrok from the Softwares folder
2. Unzip and double-click the ngrok.exe



Publish

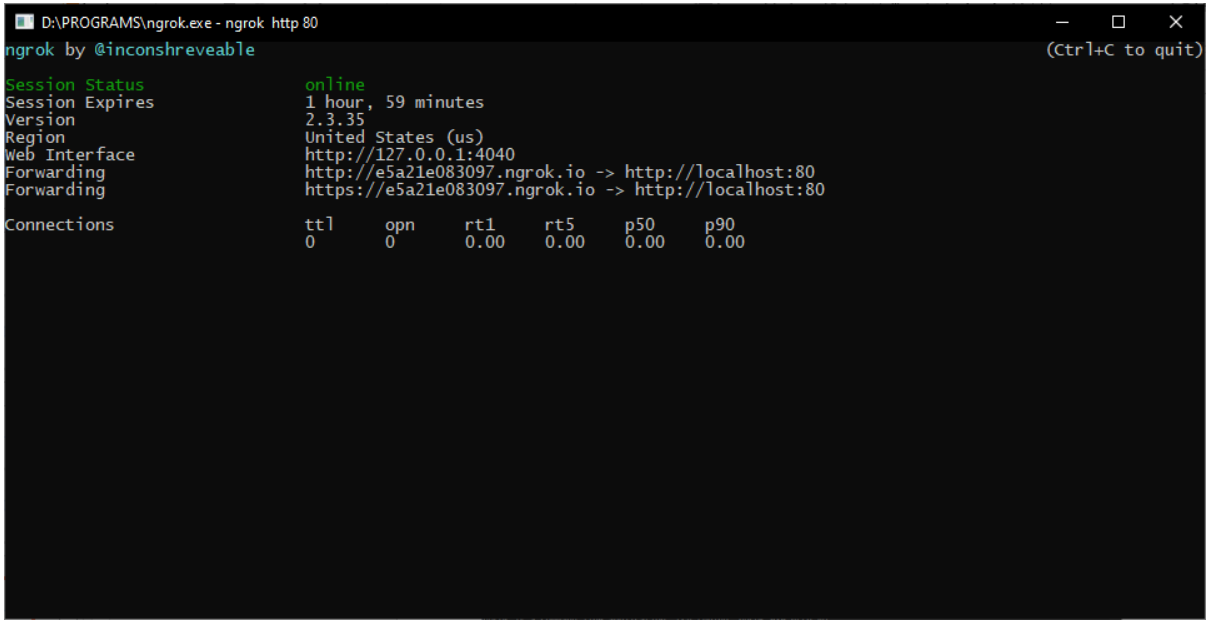
You can quickly publish the localhost website to your client.

1. Download ngrok from the Softwares folder
2. Unzip and double-click the ngrok.exe

Publish

You can quickly publish the localhost website to your client.

1. Download ngrok from the Softwares folder
2. Unzip and double-click the ngrok.exe
3. Type “ngrok http 80”, the htdocs folder will be served in the forwarding link.



Publish

You can quickly publish the localhost website to your client.

1. Download ngrok from the Softwares folder
2. Unzip and double-click the ngrok.exe
3. Type “ngrok http 80”, the htdocs folder will be served in the forwarding link.
4. The session is temporary and will terminate after a few hours (usually 2 hours)

Example Specifications

Modules:

Student, Teacher, Course, Payment

Database Specifications:

TABLE student: id, dept, name, nid, birth, address, etc.

TABLE teacher: id, dept, name, nid, birth, address, etc.

TABLE course: id, dept, title, credit, syllabus, etc.

TABLE payment: payment_id, student_id, amount, date, etc.

API Specifications

Every module has CRUD functions and some cross-module functions
CRUD: Create, Read, Update, Delete.

Example of cross-module function:

addCourseToTeacher(course_id, teacher_id)

Student

createStudent(id, dept, name, nid, birth, address)
readStudent(dept, batch)
updateStudent(id, dept, name, nid, birth, address)
deleteStudent(id)

Teacher

createTeacher(id, dept, name, nid, birth, address)
readTeacher(dept)
updateTeacher(id, dept, name, nid, birth, address)
deleteTeacher(id)

API Specifications

Course

createCourse(id, dept, title, credit, syllabus)
readCourse(dept, semester)
updateCourse(id, dept, title, credit, syllabus)
deleteCourse(id)

Payment

createPayment(payment_id, student_id, amount, date)
readPayment(student_id)
updatePayment(payment_id, student_id, amount, date)
deletePayment(payment_id)

Cross-Module API

createCourseToSemester(course_id, dept, semester)
removeCourseFromSemester(course_id, semester)
addCourseToTeacher(course_id, teacher_id)
removeCourseFromTeacher(course_id, teacher_id)

Task

Tables:

Teacher: teacher_id, name, course_id
Course: course_id, title

APIs to implement:

1. Read Teachers
2. Read Courses
3. Assign Teacher To Course