**Human Computer Interaction HCI : Theory ISE 1 of 2**

**Development Activity**

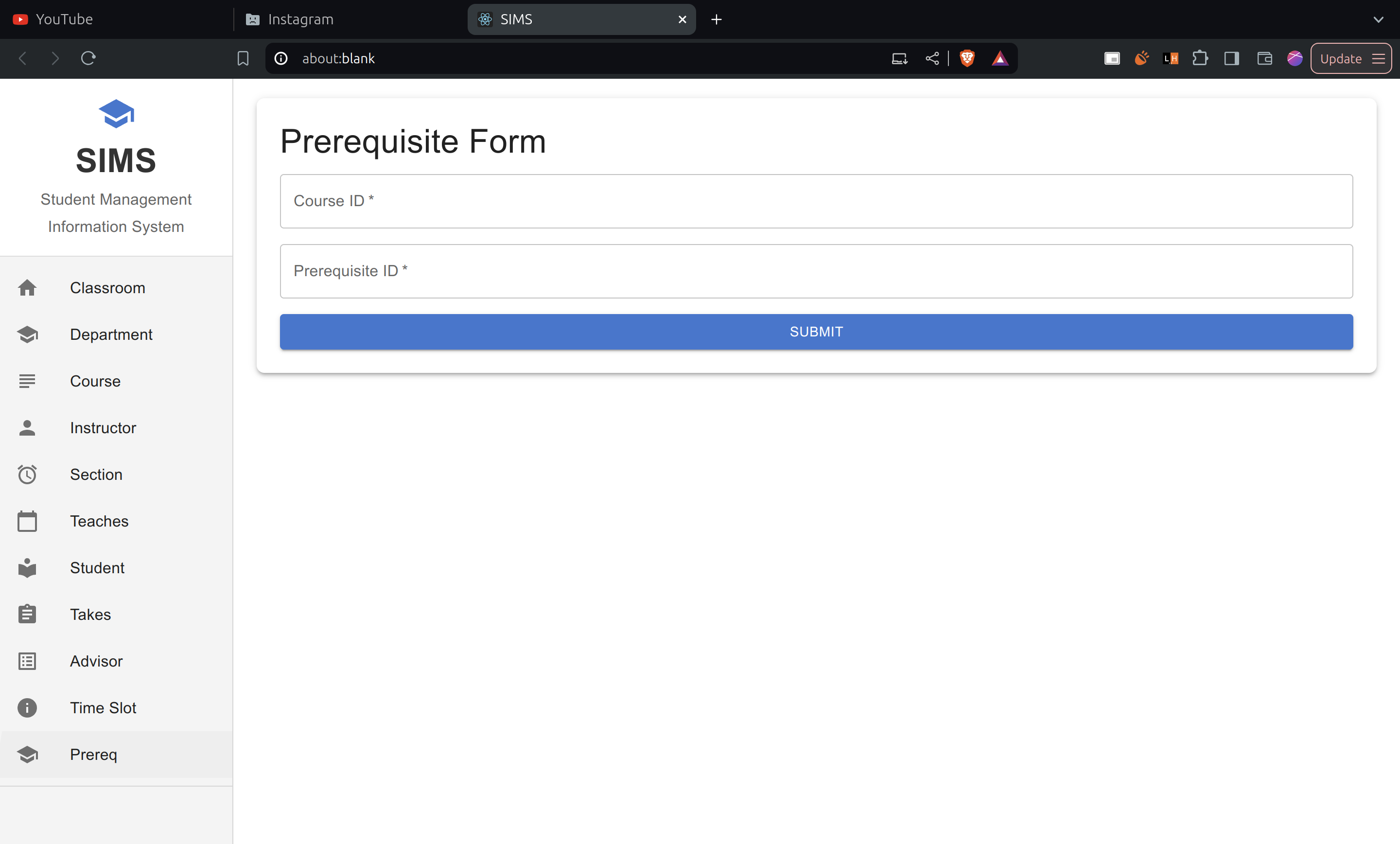
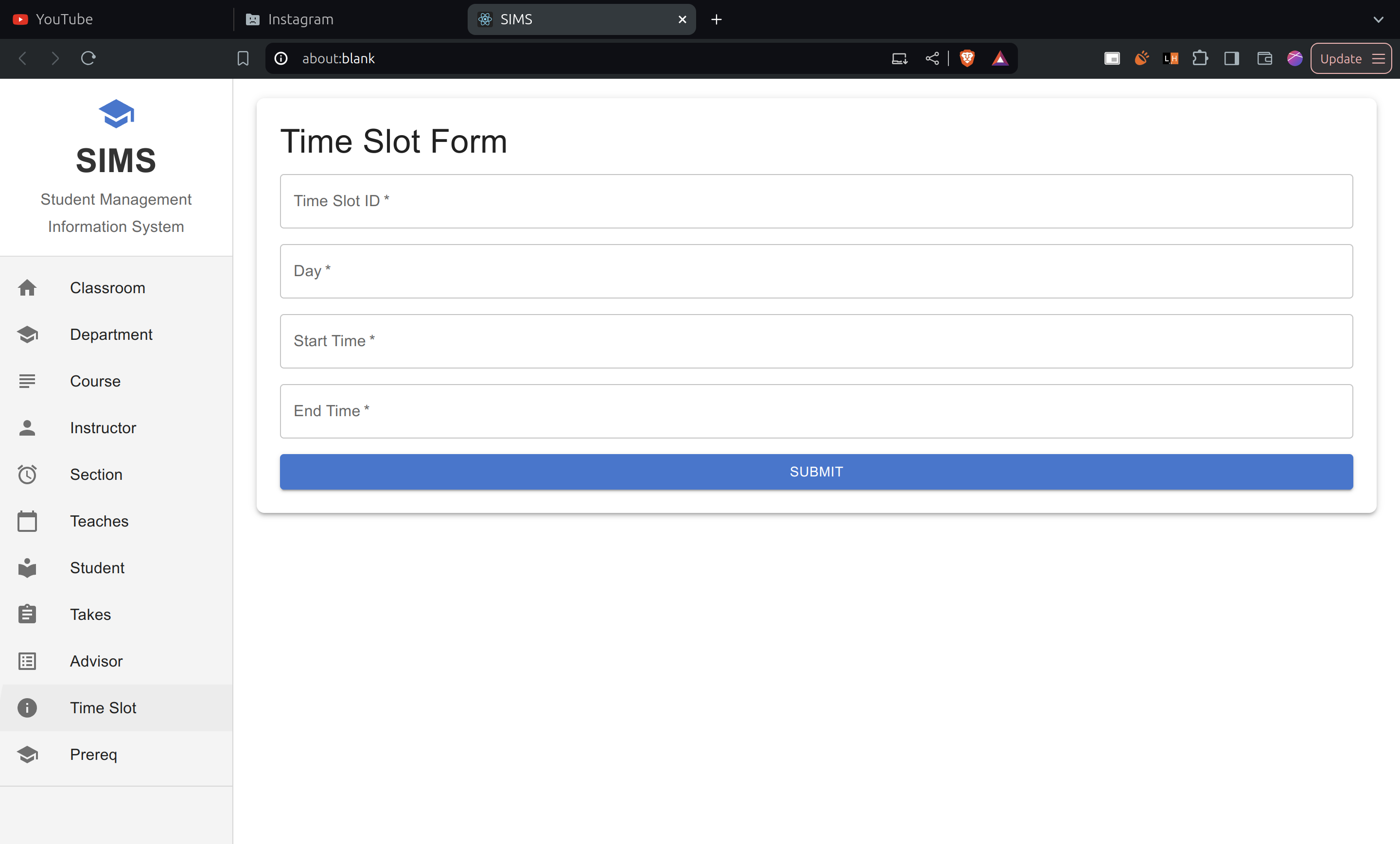
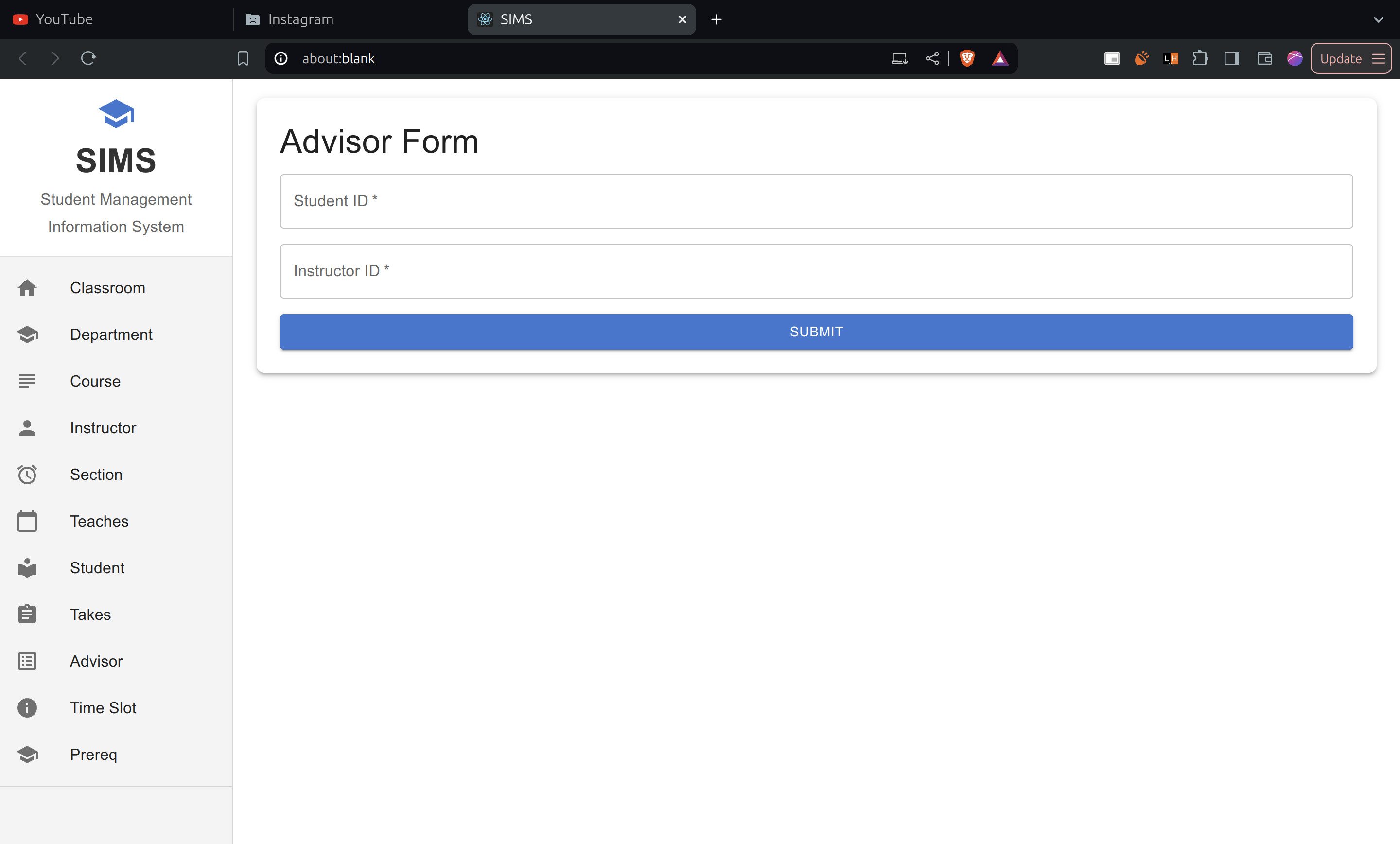
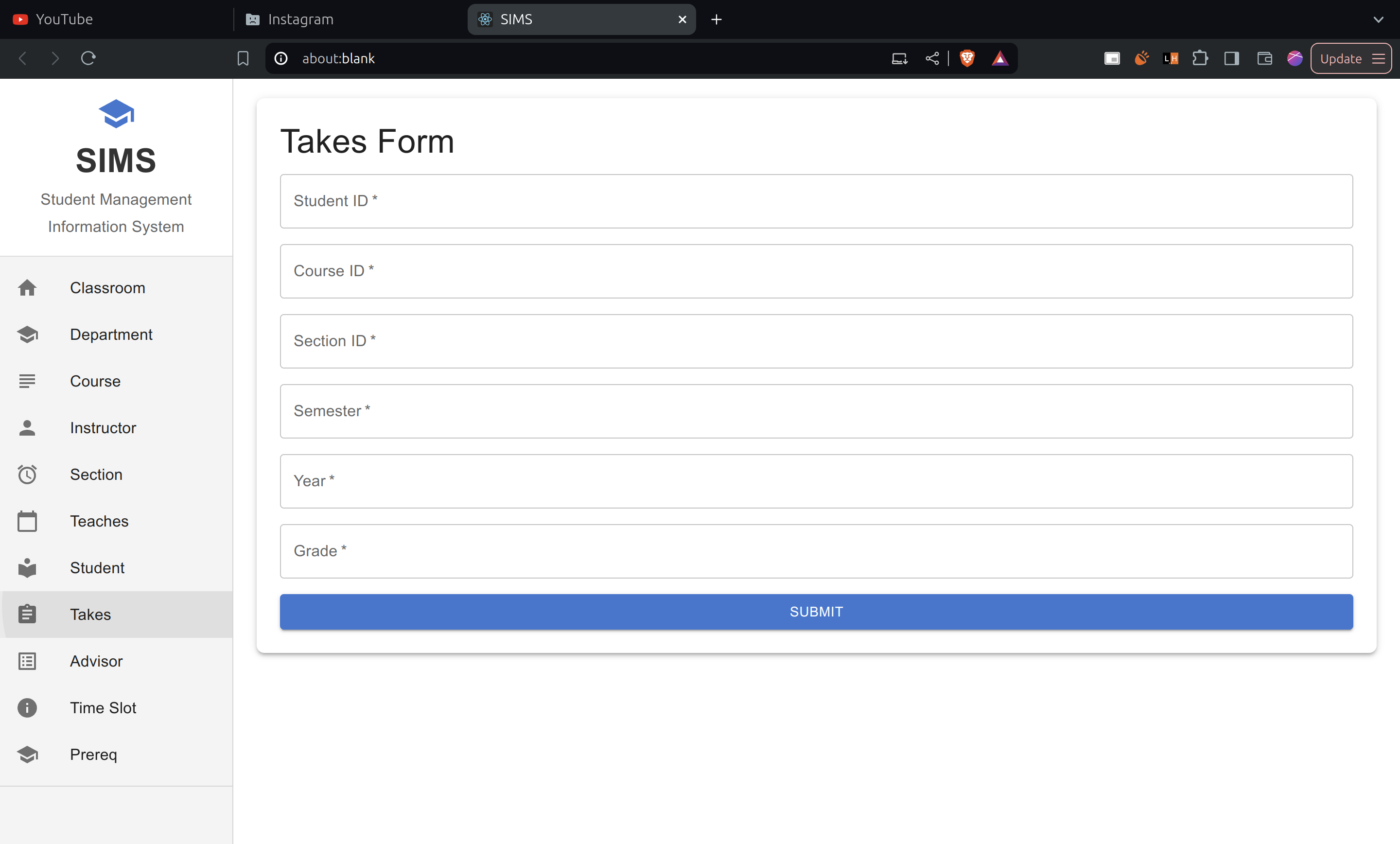
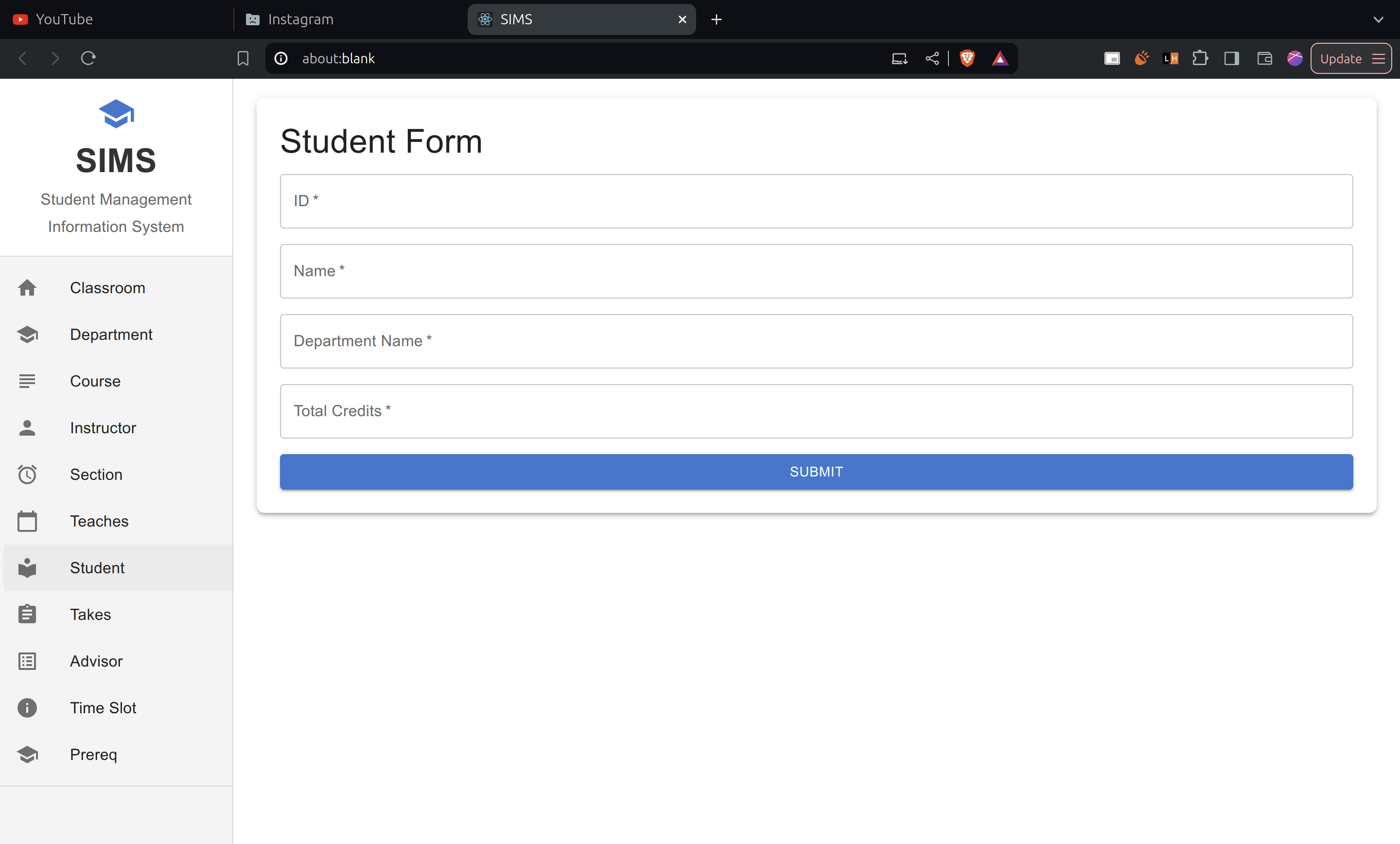
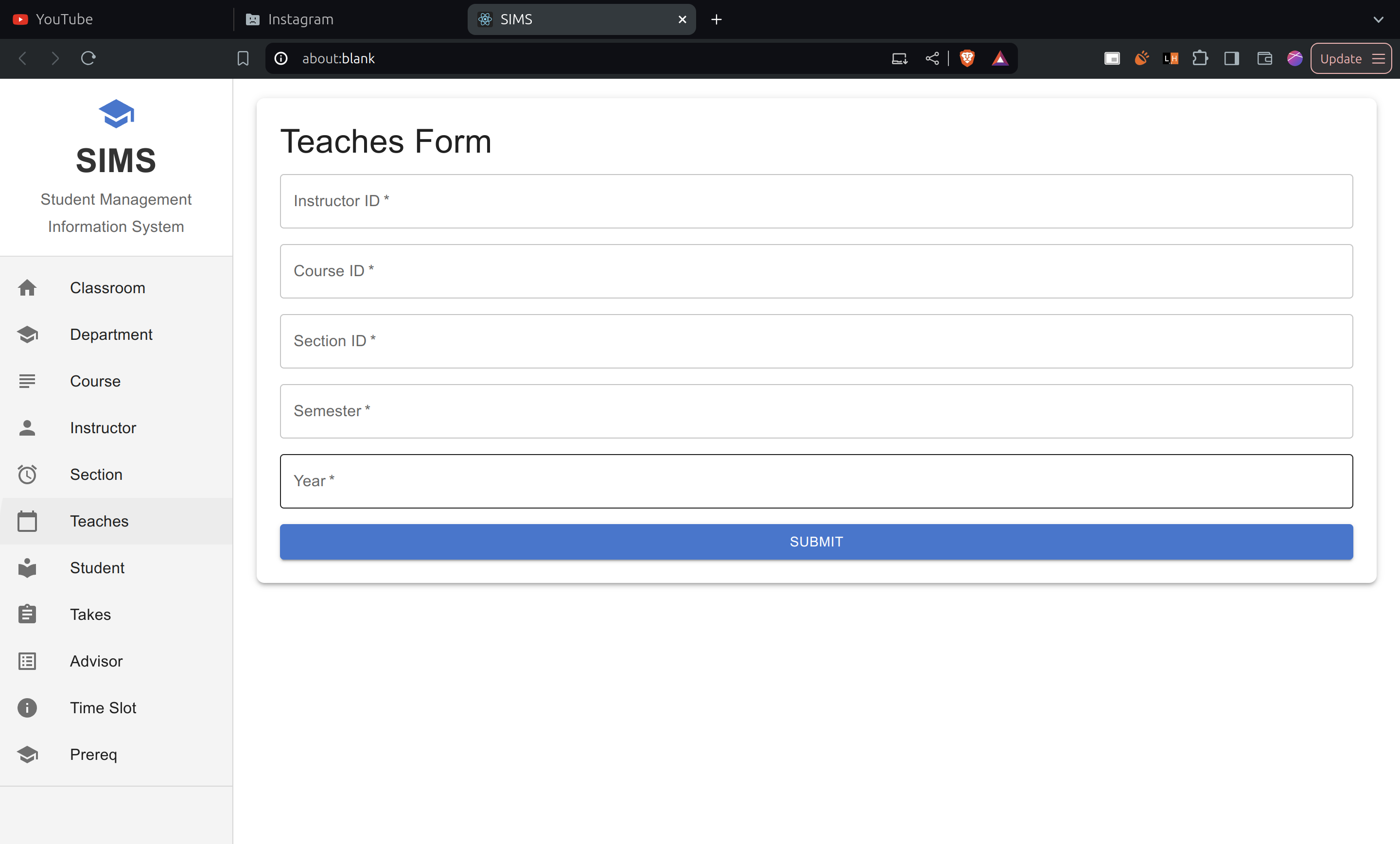
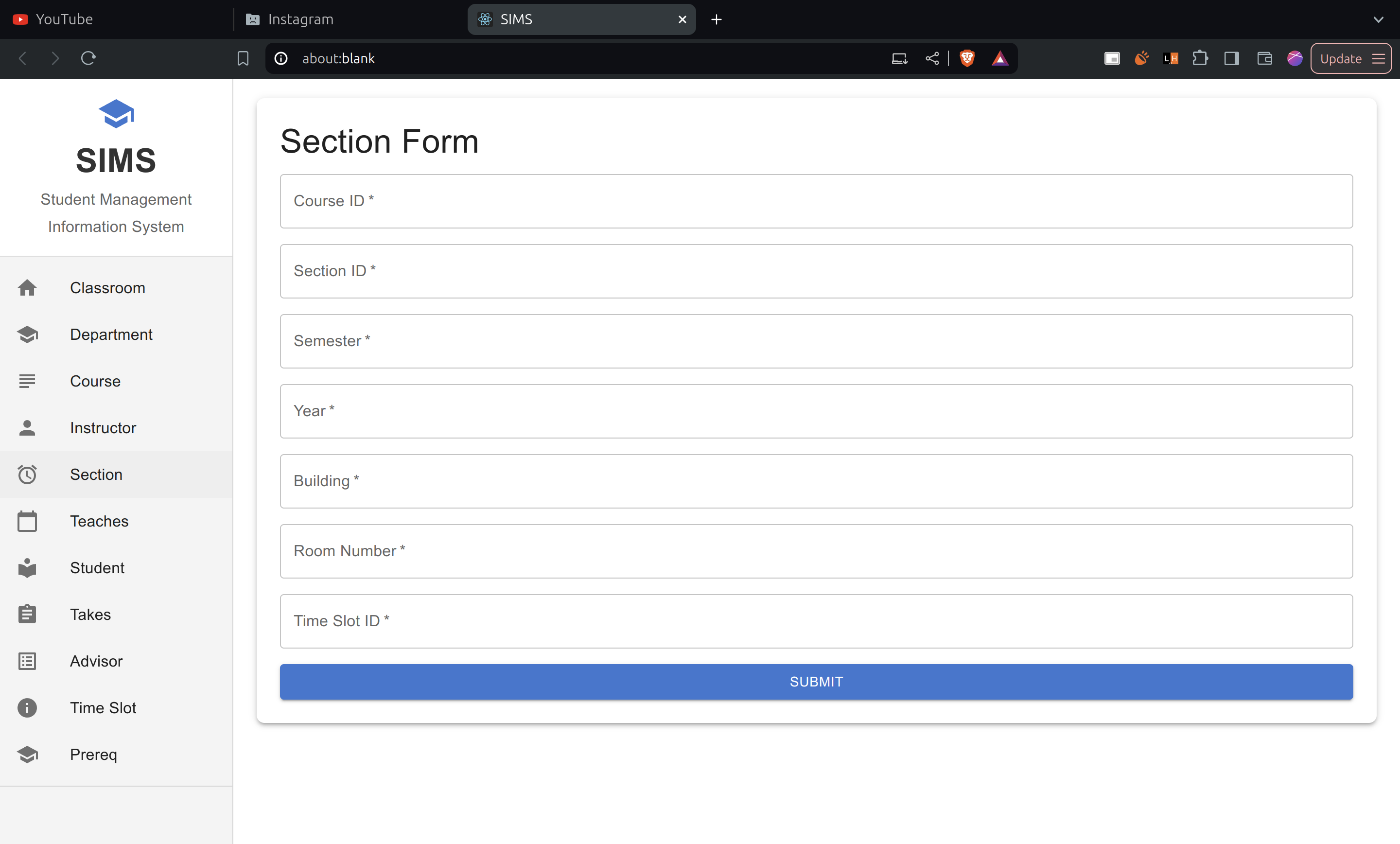
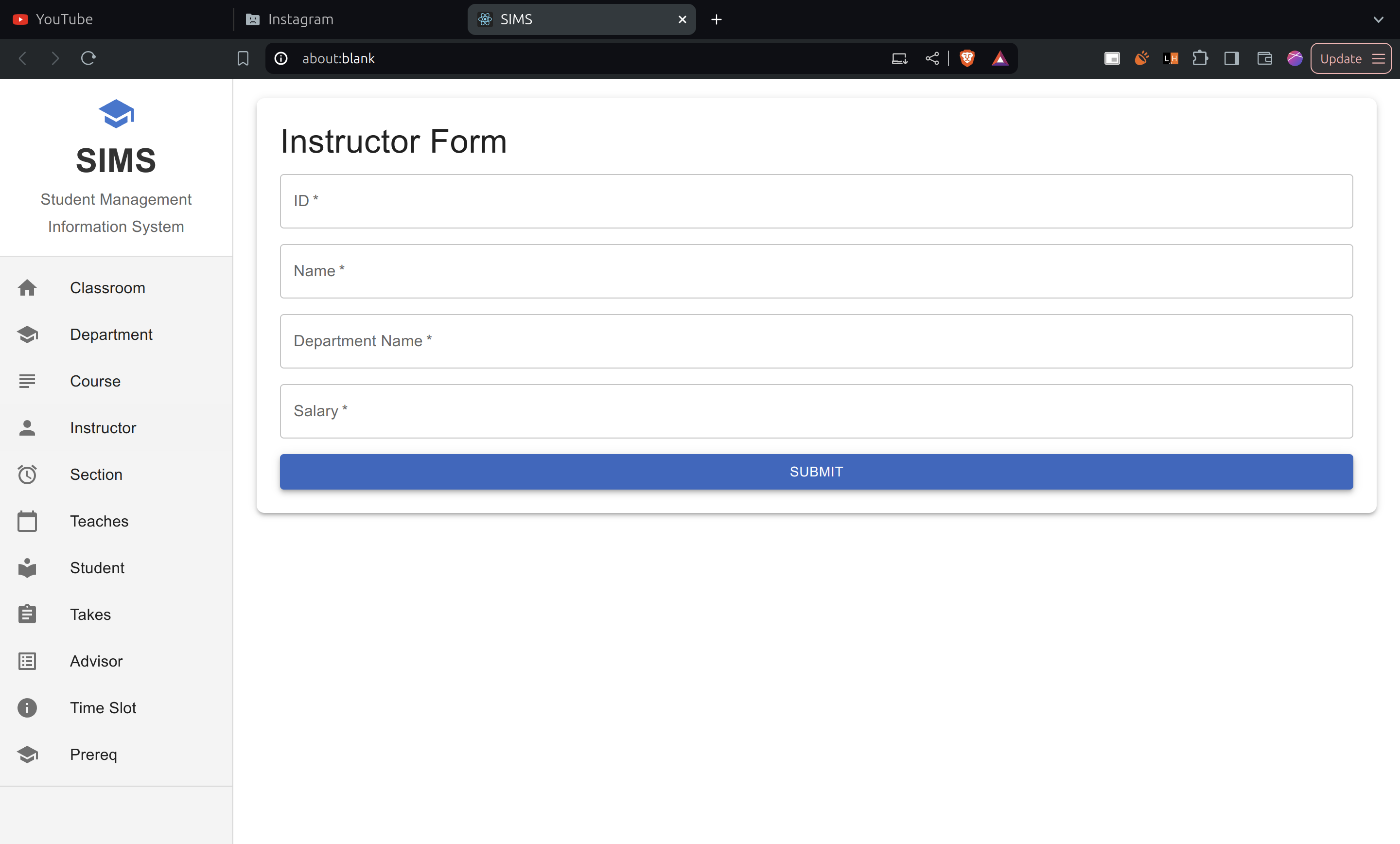
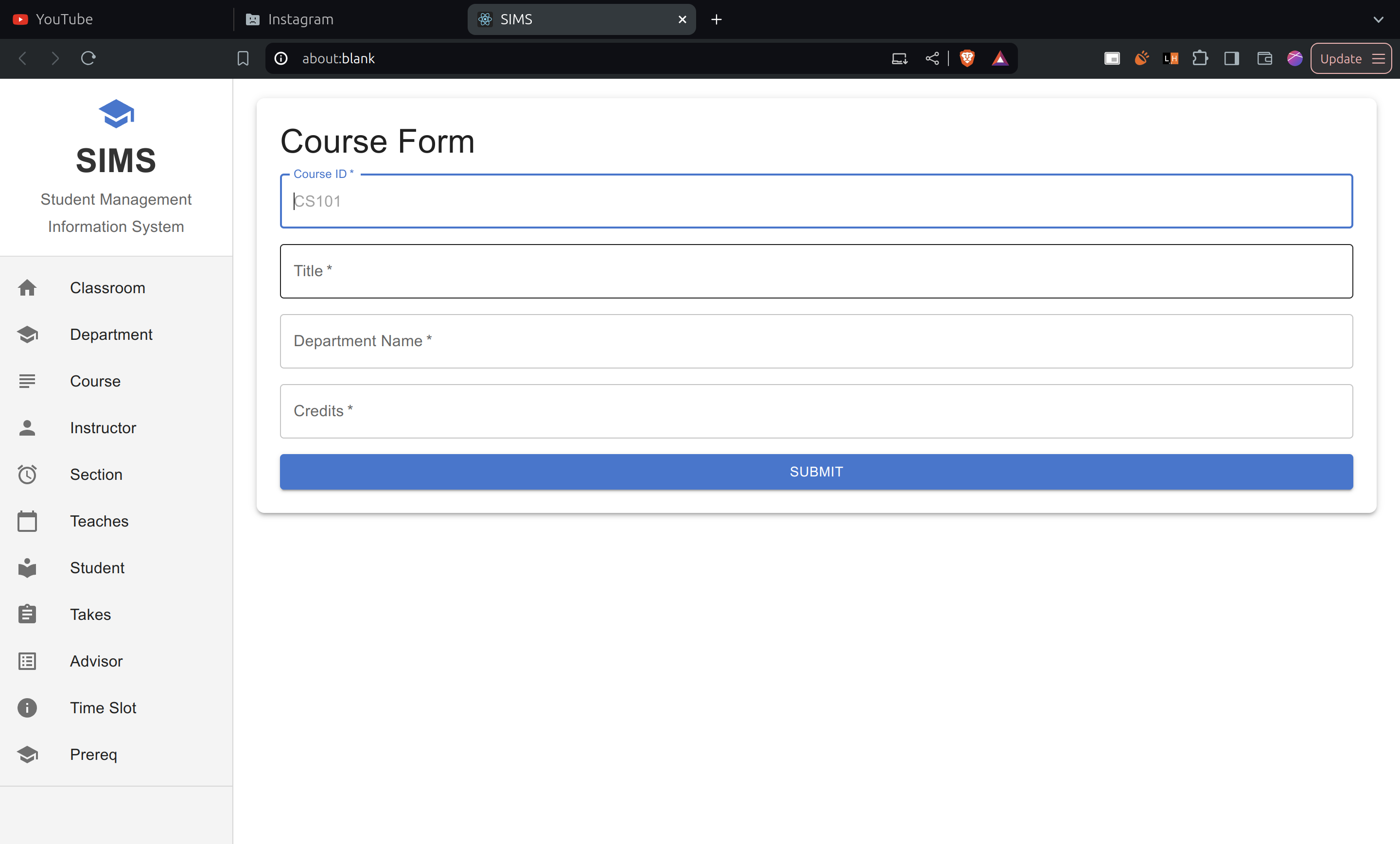
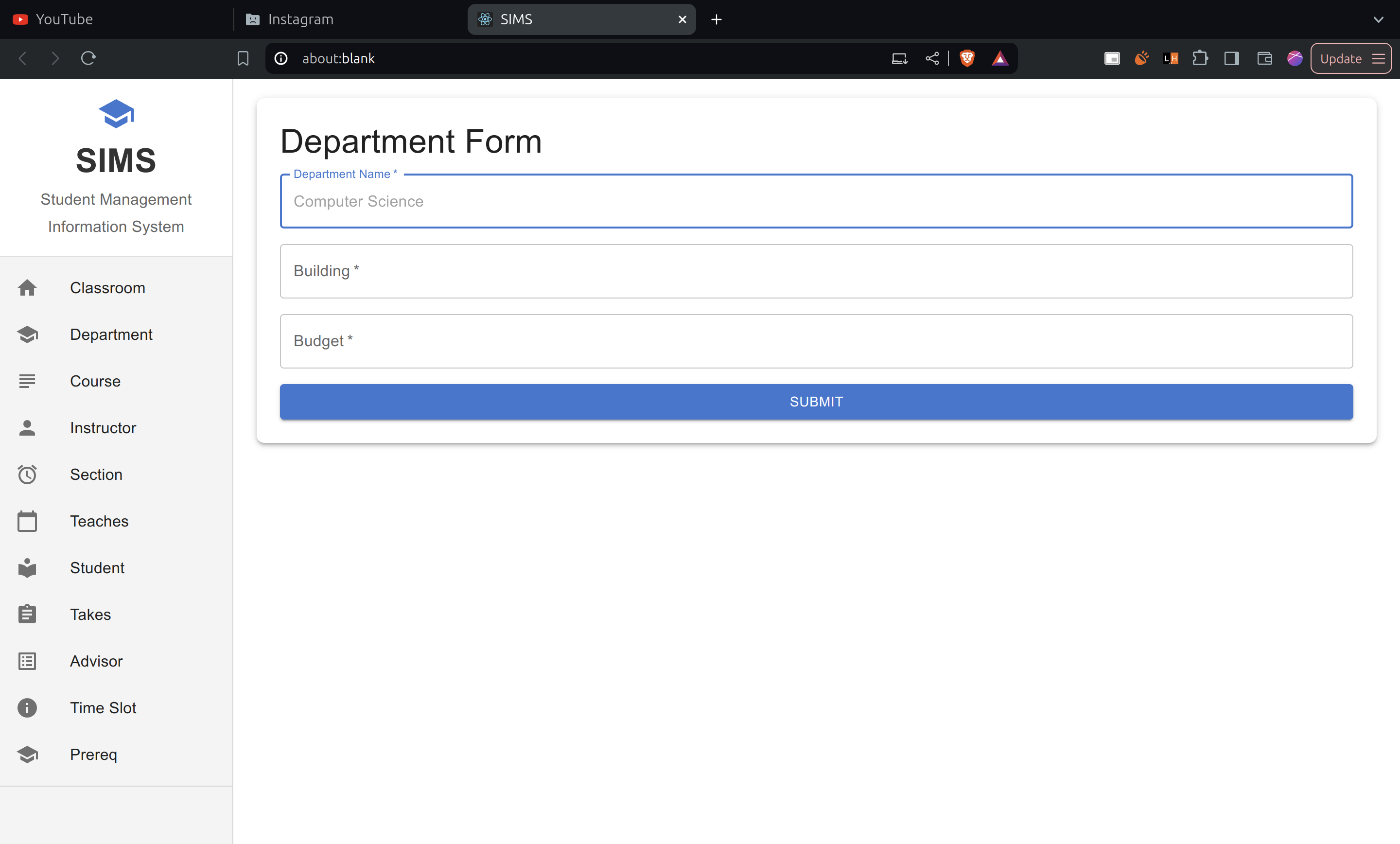
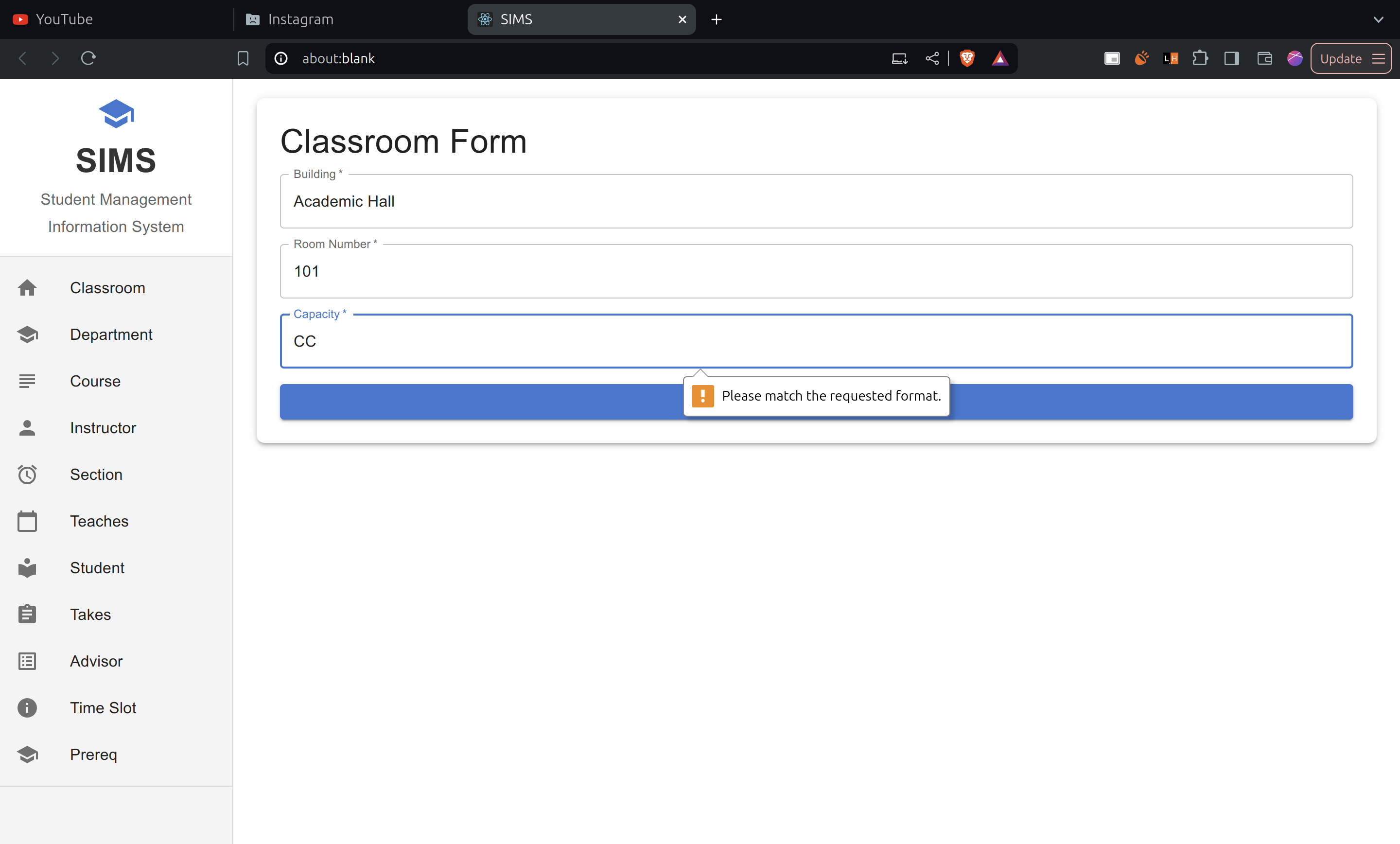
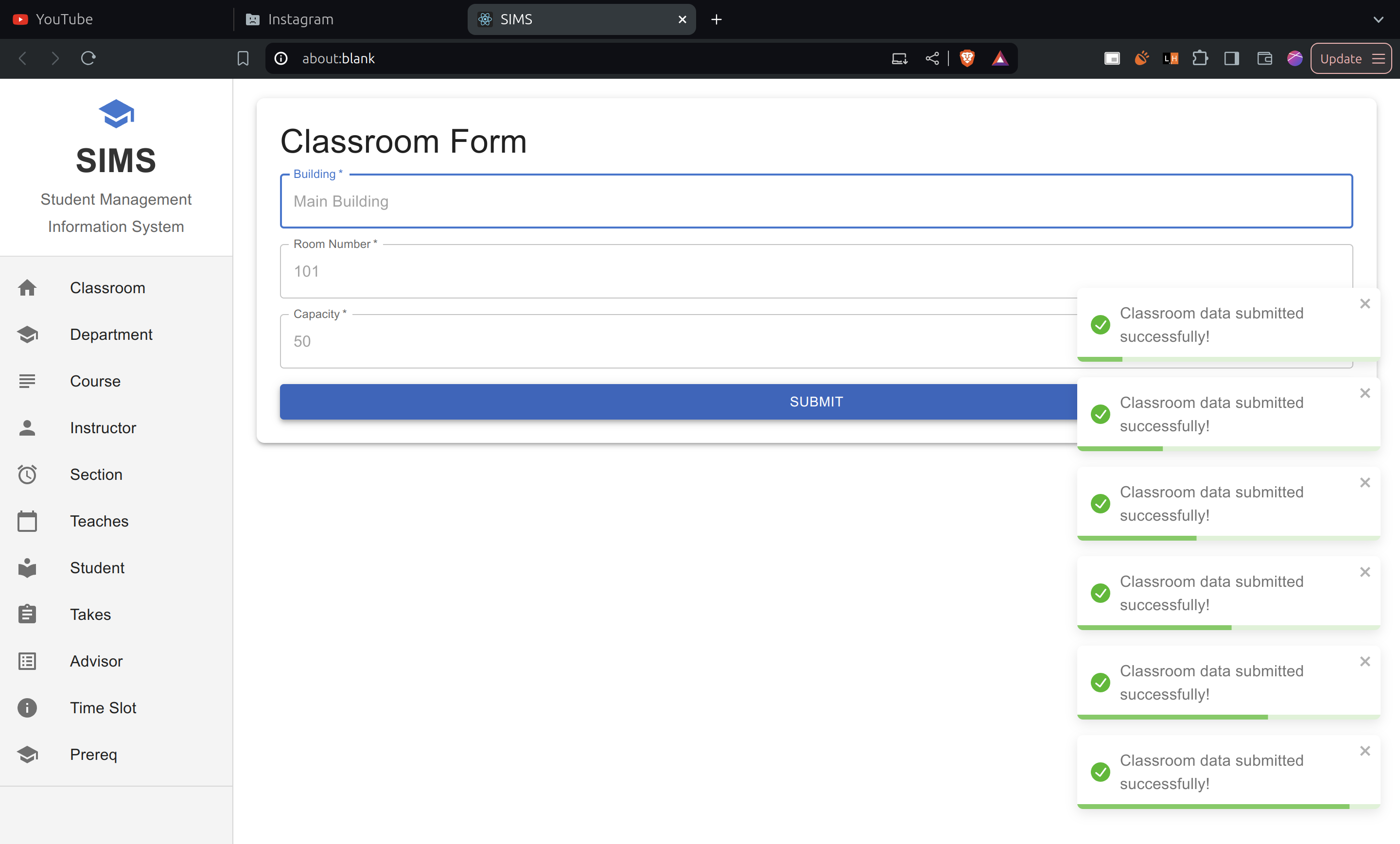
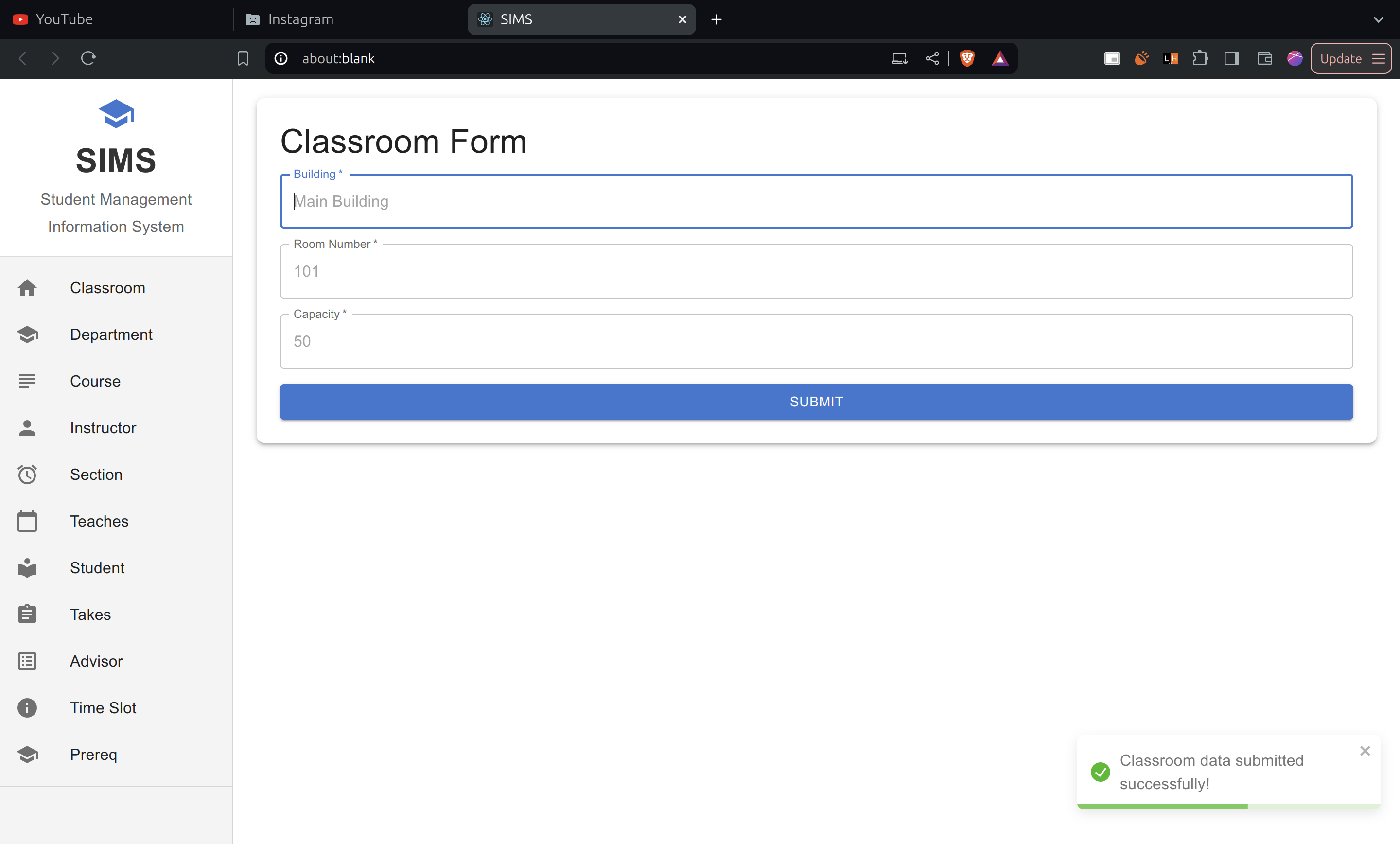
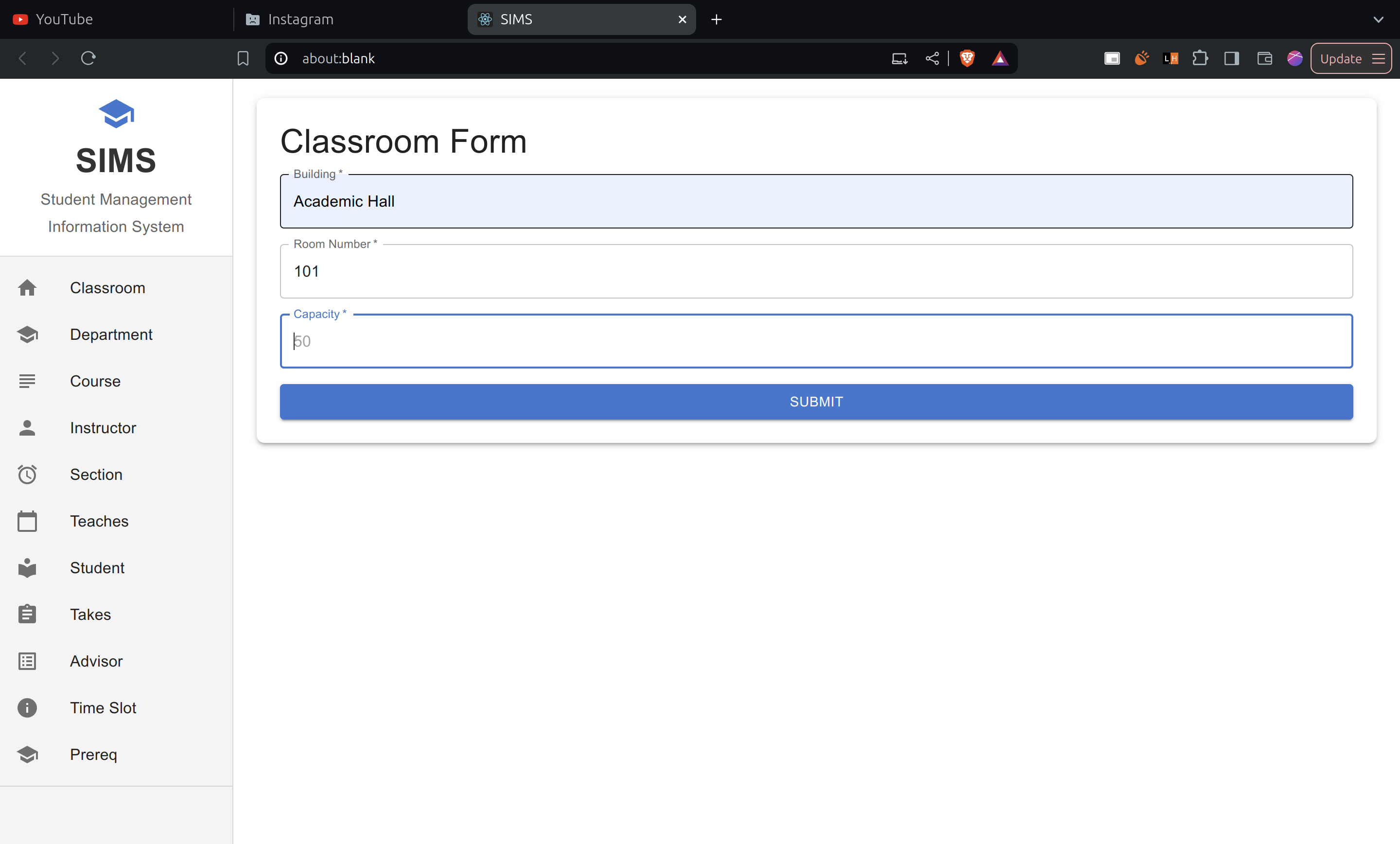
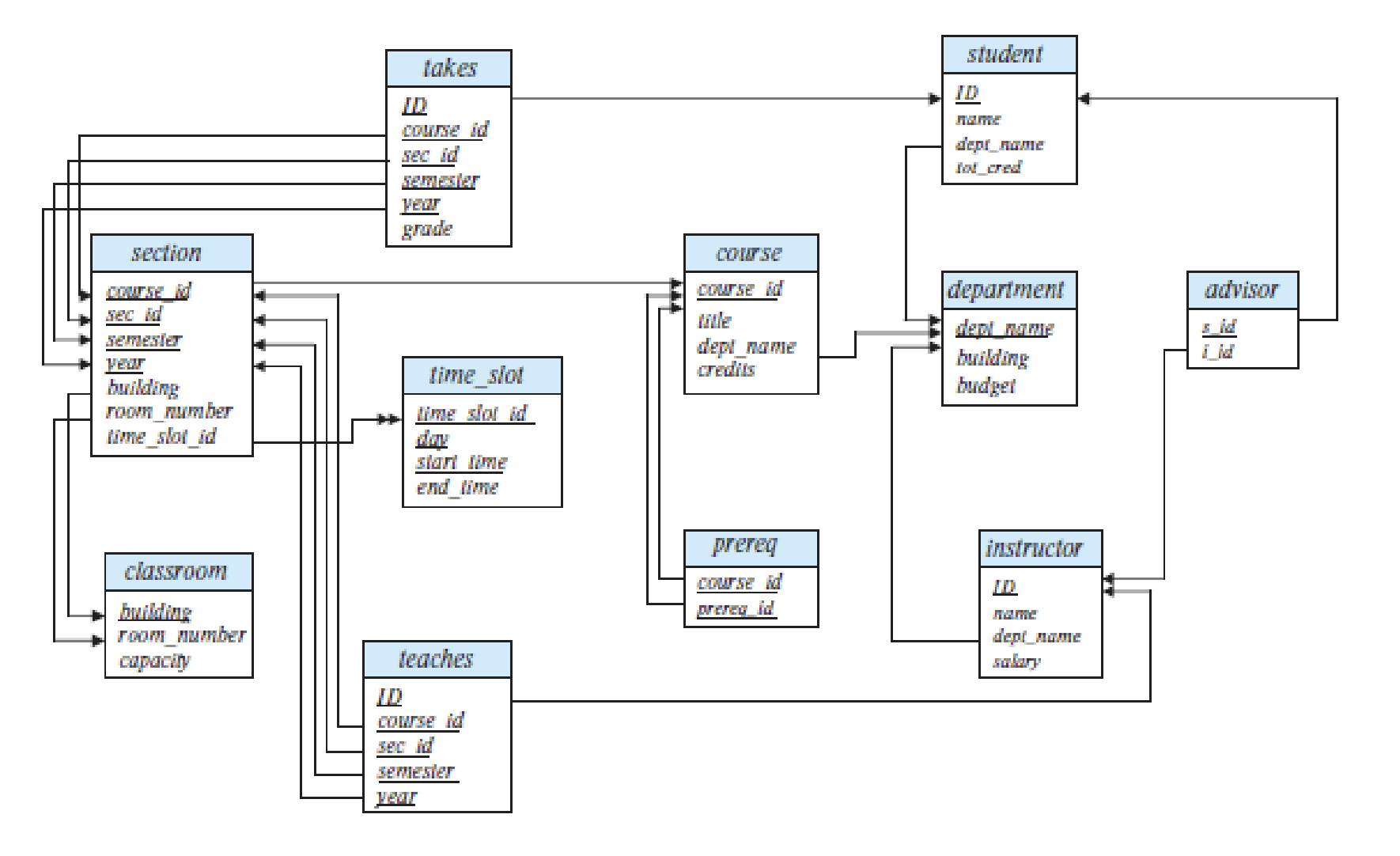
1. Develop the User Interface for the following task by applying HCI Screen design and Layout guidelines : Schneiderman’s eight golden rules.

2. Use React platform with different state of the art UI frameworks. No backend database / No server side coding

**Activity - 2 :** Student Management Information System for University schema

Design the menus and data entry form for each entity of following University Schema

* *Classroom (building, room number, capacity)*
* *Department (dept name, building, budget)*
* *Course (course id, title, dept name, credits)*
* *Instructor (ID, name, dept name, salary)*
* *Section (course id, sec id, semester, year, building, room number, time slot id)*
* *Teaches (ID, course id, sec id, semester, year)*
* *Student (ID, name, dept name, tot cred)*
* *Takes (ID, course id, sec id, semester, year, grade)*
* *Advisor (s ID, i ID)*
* *time slot (time slot id, day, start time, end time)*
* *prereq (course id, prereq id)*



**React UI components using MUI:** [**https://mui.com/**](https://mui.com/)

App.js

// src/App.js

import React, { useState } from 'react';

import Sidebar from './components/Sidebar';

import ClassroomForm from './components/ClassroomForm';

import DepartmentForm from './components/DepartmentForm';

import CourseForm from './components/CourseForm';

import InstructorForm from './components/InstructorForm';

import SectionForm from './components/SectionForm';

import TeachesForm from './components/TeachesForm';

import StudentForm from './components/StudentForm';

import TakesForm from './components/TakesForm';

import AdvisorForm from './components/AdvisorForm';

import TimeSlotForm from './components/TimeSlotForm';

import PrereqForm from './components/PrereqForm';

import { Container, CssBaseline, Paper } from '@mui/material';

function App() {

const [selectedForm, setSelectedForm] = useState('Classroom');

const renderForm = () => {

switch (selectedForm) {

case 'Classroom':

return <ClassroomForm />;

case 'Department':

return <DepartmentForm />;

case 'Course':

return <CourseForm />;

case 'Instructor':

return <InstructorForm />;

case 'Section':

return <SectionForm />;

case 'Teaches':

return <TeachesForm />;

case 'Student':

return <StudentForm />;

case 'Takes':

return <TakesForm />;

case 'Advisor':

return <AdvisorForm />;

case 'Time Slot':

return <TimeSlotForm />;

case 'Prereq':

return <PrereqForm />;

default:

return null;

}

};

return (

<div>

<CssBaseline />

<Sidebar onSelect={setSelectedForm} />

<Container component="main" maxWidth="lg" sx={{ marginLeft: '240px', padding: '20px' }}>

<Paper elevation={3} sx={{ padding: 3, borderRadius: 2 }}>

{renderForm()}

</Paper>

</Container>

</div>

);

}

export default App;

Sidebar.js

import React from 'react';

import { Drawer, List, ListItem, ListItemIcon, ListItemText, Divider, Typography, Box } from '@mui/material';

import { Home, School, Subject, Person, AccessAlarm, CalendarToday, LocalLibrary, Assignment, ListAlt, Info, School as SchoolIcon } from '@mui/icons-material';

const Sidebar = ({ onSelect }) => {

const handleSelect = (formName) => {

onSelect(formName);

};

return (

<Drawer

variant="permanent"

sx={{

width: 240,

flexShrink: 0,

'& .MuiDrawer-paper': {

width: 240,

boxSizing: 'border-box',

backgroundColor: '#f4f4f4',

// paddingTop: 2,

paddingBottom: 2,

},

}}

>

<Box sx={{ padding: 2, display: 'flex', flexDirection: 'column', alignItems: 'center', borderBottom: '1px solid #ddd', backgroundColor: '#fff' }}>

<SchoolIcon sx={{ fontSize: 40, color: '#1976d2', mb: 1 }} />

<Typography variant="h4" component="div" sx={{ fontWeight: 'bold', color: '#333', mb: 0.5 }}>

SIMS

</Typography>

<Typography variant="subtitle1" component="div" sx={{ color: '#666', textAlign: 'center'}}>

Student Management Information System

</Typography>

</Box>

<List>

<ListItem button onClick={() => handleSelect('Classroom')}>

<ListItemIcon><Home /></ListItemIcon>

<ListItemText primary="Classroom" />

</ListItem>

<ListItem button onClick={() => handleSelect('Department')}>

<ListItemIcon><School /></ListItemIcon>

<ListItemText primary="Department" />

</ListItem>

<ListItem button onClick={() => handleSelect('Course')}>

<ListItemIcon><Subject /></ListItemIcon>

<ListItemText primary="Course" />

</ListItem>

<ListItem button onClick={() => handleSelect('Instructor')}>

<ListItemIcon><Person /></ListItemIcon>

<ListItemText primary="Instructor" />

</ListItem>

<ListItem button onClick={() => handleSelect('Section')}>

<ListItemIcon><AccessAlarm /></ListItemIcon>

<ListItemText primary="Section" />

</ListItem>

<ListItem button onClick={() => handleSelect('Teaches')}>

<ListItemIcon><CalendarToday /></ListItemIcon>

<ListItemText primary="Teaches" />

</ListItem>

<ListItem button onClick={() => handleSelect('Student')}>

<ListItemIcon><LocalLibrary /></ListItemIcon>

<ListItemText primary="Student" />

</ListItem>

<ListItem button onClick={() => handleSelect('Takes')}>

<ListItemIcon><Assignment /></ListItemIcon>

<ListItemText primary="Takes" />

</ListItem>

<ListItem button onClick={() => handleSelect('Advisor')}>

<ListItemIcon><ListAlt /></ListItemIcon>

<ListItemText primary="Advisor" />

</ListItem>

<ListItem button onClick={() => handleSelect('Time Slot')}>

<ListItemIcon><Info /></ListItemIcon>

<ListItemText primary="Time Slot" />

</ListItem>

<ListItem button onClick={() => handleSelect('Prereq')}>

<ListItemIcon><School /></ListItemIcon>

<ListItemText primary="Prereq" />

</ListItem>

</List>

<Divider />

</Drawer>

);

};

export default Sidebar;

Sample form:

// src/components/AdvisorForm.js

import React, { useRef } from 'react';

import { TextField, Button, Grid, Typography, Box } from '@mui/material';

import { toast, ToastContainer } from 'react-toastify';

import 'react-toastify/dist/ReactToastify.css';

const AdvisorForm = () => {

const formRef = useRef(null);

const studentIdRef = useRef(null);

const handleSubmit = (e) => {

e.preventDefault();

toast.success('Advisor data submitted successfully!', {

position: 'bottom-right',

});

if (formRef.current) {

formRef.current.reset();

}

if (studentIdRef.current) {

studentIdRef.current.focus();

}

};

return (

<Box>

<Typography variant="h4" gutterBottom>

Advisor Form

</Typography>

<form ref={formRef} onSubmit={handleSubmit}>

<Grid container spacing={2}>

<Grid item xs={12}>

<TextField

label="Student ID"

placeholder="S001"

fullWidth

required

inputRef={studentIdRef}

/>

</Grid>

<Grid item xs={12}>

<TextField

label="Instructor ID"

placeholder="I001"

fullWidth

required

/>

</Grid>

<Grid item xs={12}>

<Button type="submit" variant="contained" color="primary" fullWidth>

Submit

</Button>

</Grid>

</Grid>

</form>

<ToastContainer />

</Box>

);

};

export default AdvisorForm;

### **Schneiderman’s Eight Golden Rules**

1. **Strive for Consistency:**
   1. Followed consistent blue and white color palette.
2. **Enable Frequent Users to Use Shortcuts:**
   1. Pressing enter or tab switches to next input field.
   2. After submit using enter, cursor snaps back to first field of form.
3. **Offer Informative Feedback:**
   1. Displayed confirmation messages when actions are completed (e.g., "Classroom added").
4. **Design Dialogs to Yield Closure:**
   1. As backend was not configured, this rule has not been implemented.
   2. We can ensure that actions like "Add entity" clearly guide the user through the process, providing a final confirmation.
5. **Offer Simple Error Handling:**
   1. Provided simple error messages (e.g., "Please follow required format").
6. **Permit Easy Reversal of Actions:**
   1. As backend was not configured, this rule has not been implemented.
   2. Allow users to undo last action, delete last added item.
7. **Support Internal Locus of Control:**
   1. Predictable actions (e.g., submit should ensure data entry).
   2. Avoid unexpected changes or actions.
8. **Reduce Short-Term Memory Load:**
   1. Simple interface, showing only essential options at any given time.
   2. Used icons with each menu title.