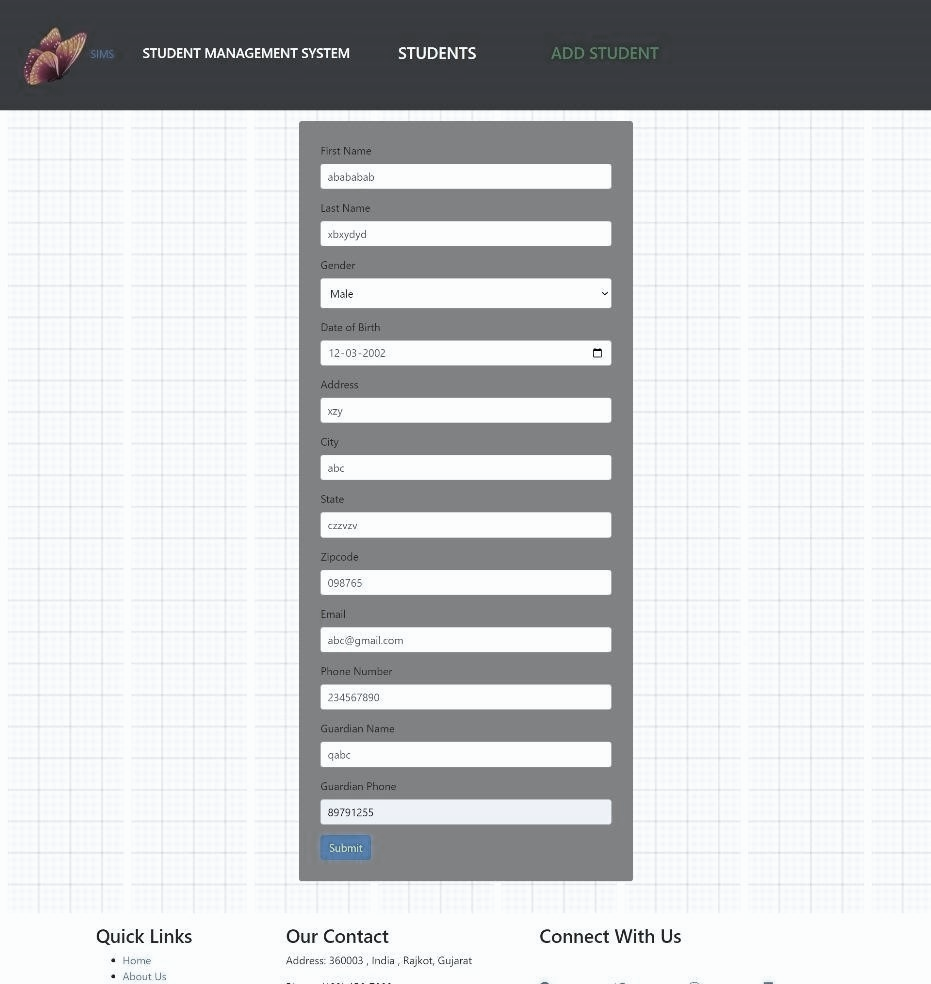
**Lab#14(a) Implement Insert Operation to add new record of student in Student Information System**

**Final Outcome:**



**Code Snippet:**

*// Route handler to create a new student* exports.createStudent = (req, res) => {

*// Extract student data from req.body and insert it into the SQLite database* const {

first\_name, last\_name, gender, date\_of\_birth,

address,

city, state, zipcode,

email, phone\_number, guardian\_name, guardian\_phone,

} = req.body;

const stmt = db.prepare(

"INSERT INTO students (first\_name, last\_name, gender, date\_of\_birth, address, city, state, zipcode, email, phone\_number, guardian\_name, guardian\_phone) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)"

);

stmt.run(

[

first\_name, last\_name, gender, date\_of\_birth, address,

city, state, zipcode, email, phone\_number, guardian\_name, guardian\_phone,

],

(err) => { if (err) {

console.error(err);

res.status(500).send("Internal Server Error");

return;

res.redirect("/sims/all");

}

);

stmt.finalize();

};

//delete student from database exports.deleteStudent = (req, res) => { const id = req.params.id; const stmt = db.prepare("DELETE FROM students WHERE id = ?");

stmt.run([id], (err) => {

if (err) {

console.error(err);

res.status(500).send("Internal Server Error"); return;

}

res.redirect("/sims/all");

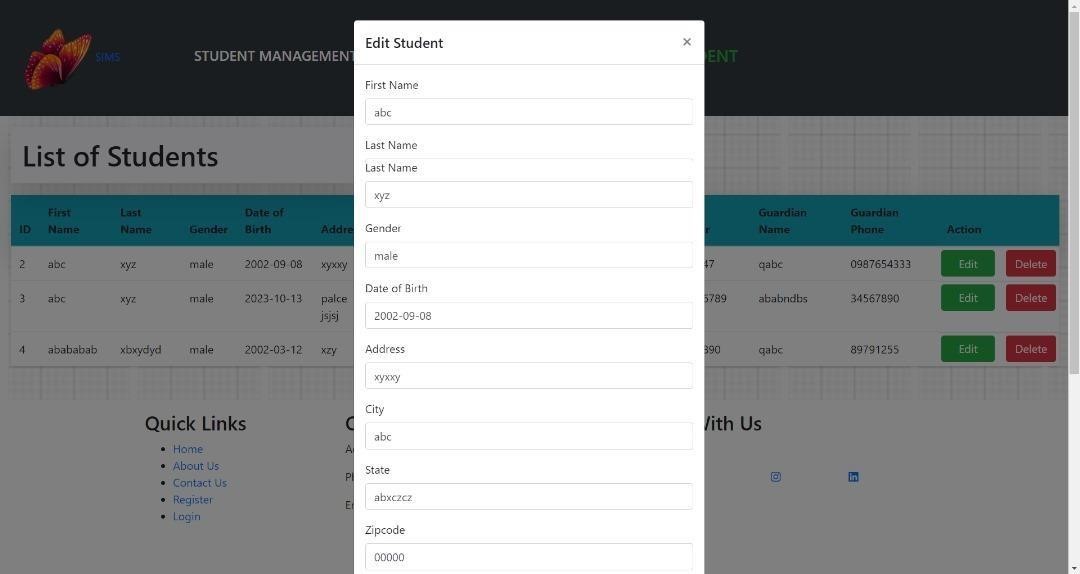
});

stmt.finalize();

};

**Lab#14(b) Implement Update Operation to modify any record of student in Student Information System**

**Final Outcome:**



**Code Snippet:**

*// Route handler to update a student by ID* exports.updateStudent = (req, res) **=>** { **const** studentId = req.params.id;

**const** {

first\_name, last\_name, gender, date\_of\_birth,

address,

city, state, zipcode,

email, phone\_number, guardian\_name, guardian\_phone,

} = req.body;

*// Begin a transaction* db.run("BEGIN TRANSACTION", **function** (beginErr) { if (beginErr) {

console.error(beginErr);

res.status(500).send("Internal Server Error"); return;

}

**const** stmt = db.prepare(

"UPDATE students SET first\_name = ?, last\_name = ?, gender = ?, date\_of\_birth = ?, address = ?, city = ?, state = ?, zipcode = ?, email = ?, phone\_number = ?, guardian\_name = ?, guardian\_phone = ? WHERE id = ?"

);

stmt.run(

first\_name, last\_name, gender, date\_of\_birth,

address,

city, state, zipcode,

email, phone\_number, guardian\_name,

guardian\_phone,

studentId,

(err) => { if (err) {

console.error(err);

// Rollback the transaction in case of an error db.run("ROLLBACK", (rollbackErr) => {

if (rollbackErr) {

console.error(rollbackErr);

}

res.status(500).send("Internal Server Error");

});

} else {

// Commit the transaction db.run("COMMIT", (commitErr) => {

if (commitErr) { console.error(commitErr); res.status(500).send("Internal Server Error");

} else { console.log("Student updated successfully."); res.redirect("/sims/all");

}

});

}

}

);

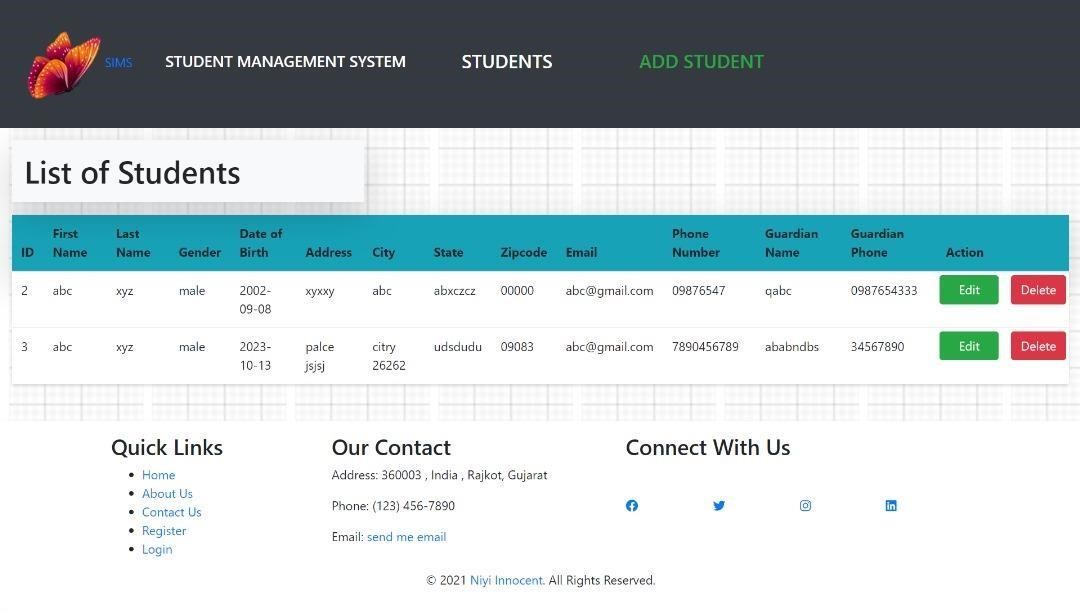
stmt.finalize();

});

};

**Lab#14(c). Implement Delete Operation to remove any student from Student Information System.**

**Final Outcome:**



**Code Snippet:**

*//delete student from database* exports.deleteStudent = (req, res) **=>** { **const** id = req.params.id; **const** stmt = db.prepare("DELETE FROM students WHERE id = ?");

stmt.run([id], (err) **=>** {

if (err) {

console.error(err);

res.status(500).send("Internal Server Error"); return;

}

res.redirect("/sims/all");

});

stmt.finalize();

};