## SQL Schema

***CREATE TABLE users (  
 id INTEGER PRIMARY KEY AUTOINCREMENT,  
 username TEXT NOT NULL UNIQUE,  
 email TEXT NOT NULL UNIQUE,  
 password\_hash TEXT NOT NULL,  
 role TEXT NOT NULL CHECK(role IN ('admin','faculty','student')),  
 created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP  
);  
  
CREATE TABLE students (  
 id INTEGER PRIMARY KEY AUTOINCREMENT,  
 user\_id INTEGER NOT NULL REFERENCES users(id) ON DELETE CASCADE,  
 name TEXT NOT NULL,  
 marks INTEGER DEFAULT 0,  
 grade TEXT DEFAULT 'F'  
);***

## 6.1 server.js (Core routing and middleware)

***const express = require('express');  
const session = require('express-session');  
const bcrypt = require('bcrypt');  
const db = require('./lib/db');  
  
app.post('/login', async (req, res) => {  
 const { username, password } = req.body;  
 const user = db.prepare('SELECT \* FROM users WHERE username = ?').get(username);  
 if (user && await bcrypt.compare(password, user.password\_hash)) {  
 req.session.user = { id: user.id, username: user.username, role: user.role };  
 return res.redirect(`/dashboard/${user.role}`);  
 }  
 res.render('login', { error: 'Invalid username or password' });  
});***

## 6.2 lib/db.js (Database helper)

***const Database = require('better-sqlite3');  
const db = new Database('data.sqlite');  
  
module.exports = db;***

## 6.3 lib/grader.js (Grade calculation helper)

***function calculateGrade(marks) {  
 if (marks >= 90) return 'A+';  
 if (marks >= 80) return 'A';  
 if (marks >= 70) return 'B+';  
 if (marks >= 60) return 'B';  
 if (marks >= 50) return 'C';  
 return 'F';  
}  
  
module.exports = { calculateGrade };***