**Lab 3 Extension**

**Validation**

Line 49: [*check*('email', 'Your email is not valid')*.not*()*.isEmpty*()],

Lines 50-55: [*check*('password', 'The password must be 8+ chars long and contain a number')

*.not*()

*.isIn*(['123', 'password', 'god'])

*.withMessage*('Do not use a common word as the password')

*.isLength*({ min: 8 })

*.matches*(/\d/)],

Line 113: [*check*('price', 'Must be an Integer')*.isInt*()],

Lines 149-154: [*check*('password', 'The password must be 8+ chars long and contain a number')

*.not*()

*.isIn*(['123', 'password', 'god'])

*.withMessage*('Do not use a common word as the password')

*.isLength*({ min: 8 })

*.matches*(/\d/)],

* In line 49 I have used validation to check whether the user has entered a valid email & not left the input blank
* In lines 50-55 & 149-154 I have used validation to check that the user enters a password t least 8 characters long, contain a number and not be a common password such as ‘123’ or ‘password’
* In line 113 I have used validation to check whether the user has used an integer for book price

**Sanitisation**

Line 65 : const *plainPassword* = *req.sanitize*(*req.body.password*);

Line 69: letnewrecord=[*req.sanitize*(*req.body.username*), *req.sanitize*(*req.body.firstname*), *req.sanitize*(*req.body.lastname*), *req.sanitize*(*req.body.email*),hashedPassword];

Line 75: varresult='Hello '+ *req.sanitize*(*req.body.firstname*)+' '+ *req.sanitize*(*req.body.lastname*)+' you are now registered! We will send an email to you at '+ *req.sanitize*(*req.body.email*);

Line 76: result += ' Your password is: '+ *req.sanitize*(*req.body.*password) +' and your hashed password is: '+ hashedPassword + ' <a href='+'./'+'>Home</a>';

Line 121: letnewrecord=[*req.sanitize*(*req.body.name*), *req.sanitize*(*req.body.price*)];

Line 127: *res.send*(' This book is added to database, name: '+ *req.sanitize*(*req.body.*name) + ' price '+ *req.sanitize*(*req.body.*price) + '. <a href='+'./'+'>Home</a>');

Line 157: const *username* = *req.sanitize*(*req.body.username*);

Line 158: const *password* = *req.sanitize*(*req.body.password*);

Line 192: *req.session.*userId = *req.sanitize*(*req.body.*username);

Line 222: letdeleteUserSQL="DELETE FROM users WHERE Username = '"+ *req.sanitize*(*req.body.username*)+"'";

Line 231: *res.send*('User: (' + *req.sanitize*(*req.body.*username) + ') was successfully deleted. <a href='+'./'+'>Home</a>');

I have used sanitisation at any input request to ensure the site is well protected from cross-site scripting (XSS) attacks.