

message	hashCode	Author
test: setup RichBodySanitization	c213c96cd01269d0845eac3c808813149325859c	BRYAN
test: sanitize dangerous script tags	42cf3192b04a9732e2b212f77e82f389f2aa6e27	BRYAN
feat: implement allowed tags	37b9847106eab859100edf92cb55244cb15beca6	BRYAN
test: add test for preserving safe html formatting	790a53247097c4554c329e9aca76ba607c142635	BRYAN
feat: add logic for body formating and clean content validation	617865821bc4159071d83958b82bae9637913231	BRYAN
refactor: added some changes to variable names and function docstrings description	f4961e49f3c2e37f4f97bb4e401a6ed92ea3e7a0	BRYAN
refactor: changing function statements for Question model and route	f9e9f97a9f5f71481c71afd7cfc2e7185ca627ad	BRYAN
refactor: modified strip value to remove disallowed tags	9c655559f4a20032d8404f8062c676a052a2ff0b	BRYAN
refactor: modify required functions for tests	593057a47c5bea6c677d8fb4a1ccff3149d06d0a	BRYAN
test: add first fuzzy search test for exact match	aaf6175dc98fc8adce0d2386442ba85552115ed0	BRYAN
feat: implement basic fuzzy search functionality	7f06279fd877edef6f9fc29e459eec948695cdd8	BRYAN
test: add edge cases for fuzzy search	79b9350777417b9a74f2fbc40896534f948826ae	BRYAN
test: add word order independence test	007d09a98ddd11e20a352ad6d9d78ea1cb023e8a	BRYAN
feat: improve fuzzy search with word-based matching	e4d8c2b8e36a0e258026a24a7b3d1b4498d7828b	BRYAN
refactor: improve fuzzy search scoring and filtering	c969dea9fed2401b26d53054f54b1f35040bad9e	BRYAN
test: database general setup file created	6fe311e476c6aac9498ae8e17d78738071572840	BRYAN
test: starting the test for endpoint fuzzy search in question routes	826e22e1c0161c2b7d6d3bcd62fbb227df0716ea	BRYAN
feat: implement basic routing for fuzzy search	c094fb76e2ba0472ae610e72aa42a23ea03484d3	BRYAN

refactor: added some edge cases validations to actual fuzzy search code	a5200639fffa118aaa3c2d67282e4c24ea3437f5	BRYAN
refactor: changed some variables names for better understanding	6d774abb1780a0801b7d01d6521a8a36b392ed75	BRYAN
test: adding test case for basic view counting	23187273ec68a5321eac06f7124308ac55cd850a	BRYAN
feat: implement attribute and routing modifier for question view route	4b35837dce316ffb5a6631914d7ef31f9cd159c9	BRYAN
refactor: fixed missvalidation in test view count	e5e71a641105de54a63bc67907d2e47aceced920	BRYAN
test: added basic endpoint existence test for question-tag route	e7ac3529bd3aca9673e8214c2ec1d31ed66594ea	BRYAN
feat: added minimum implementation to pass routing validation questionTag	6a12afad76f2fb2e5ea810d55e8364c2225d0e59	BRYAN
test: added question with multiple tags, questions empty response, question with no tags test cases	4af1cfec8b8bdac9fcfa03cff2ad7fbd4841f408	BRYAN
feat: added get question per tags logic and added routing to app.py file	f169737923bf7d7078e246d333fec374da588bdd	BRYAN
feat: full tag association logic	84d9e177f789dfbd09a3c2835e550419a304ae90	BRYAN
refactor: optimized query call db	2dcad8b54410979646e7bdd4d67fce265e7ca33c	BRYAN
refactor: changed the url prefix to solve conflict in routing call	01a0da2ddff8e72cd20e3f125801380713b3f38f	BRYAN
refactor: modified votes attribute	3bd471dfaa335ae0d17ae7450f26aed2b54b6a7f	BRYAN
test: implemented validation for get and post votes	1ae4acfbf1a65b31ebbbfa257dc8b3aad53b5496	BRYAN
docs: introduced comments	5e969398f23e6a9073d29687182e59a918333f08	BRYAN
feat: added get and post routes for votes	2d5b1b5b12cb7a15c66a24b5fb420547a40a54b9	BRYAN
test: added setup function for answer, and target vote count tests	ce41fdb51d12ffb37157685f063b0f0021494e56	BRYAN
feat: added route for getting the vote counter	d836b3db85ef8fa20ff90e90d85e9680b0b8d850	BRYAN

refactor: added fields needed for post vote	686d8e3f08bd8f5a729de0eb23b6df0377c5a768	BRYAN
test: added test for switching between upvote and downvote	da5de737305e88dcfb12f83a9d0b5bcbe1a05233	BRYAN
test: created the test case for get all votes per user validation	6ca452d3609f8da55bf2b6303b7a02604c6dca54	BRYAN
feat: added implementation for get route for votes for user	5956b815bb29d788e62cb7af3ff7336e3804466a	BRYAN
refactor: modified the response for votes counting	0c82b62219f15e611153751c83d012e14ea8320a	BRYAN
feat: created comment model	699ca09710780bf2ebc2a98f4eb0d1d31a22908d	BRYAN
test: added basic test for validation GET /answers/<id>/comments endpoint	661d11a53fb4e3d96ca5820572b7c4e6927a25e9	BRYAN
feat: added logic for GET /answer/comments works	afbae64f2f0c6beb16ee67df9b918191d35ce32c	BRYAN
test: added basic testing functions for POST comments endpoint	869370050b9bfca9ef4a1c5a7274d1bd0b9db03e	BRYAN
feat: added logic for POST comment endpoint	2aba2fc604e8ecc7559959b7839d15c9a1374496	BRYAN
test: function to test PATCH req for comments	9e8d246ec7a5645007b5b2543df160982b867f14	BRYAN
feat: added logic for PATCH comment endpoint	a56741a9cbd97586b5291c1176a9bbf28dc7b7b4	BRYAN
feat: added logic for DELETE comment endpoint	2e9c82749df4f6b450c386ae5d82de010222ca0e	BRYAN
test:added validation for GET all comments'	e5892110dfac99c190a34e0d8684192f6dfad165	BRYAN
feat: added route logic for GET all comments	62d4ba9e66e76c5b04f6f4096f24364ece975989	BRYAN
test: added basic test for integration ai route	2dc53023ad6c93b063b8ebc195c1c3bd2dbbee5b	BRYAN
feat: added service AI class an setup	2602e9df9806b05f9467175b77e8fecc5cfae5ff	BRYAN
feat:added routes for ai answers	50219e260d177d71a8742ca90a36caa9ffc412b6	BRYAN
refactor: added edge cases for complex questions	b142cdad14f79904e44ea26f488565d7b2d2752d	BRYAN

refactor: fixed the truncated responses, and added validation	34cb4956a15054e3c794d7b74aa136903b81f89d	BRYAN
test: added test case for answer summarizing	11f2a3686fa261f7ac1775f4b533cc8f0cecb673	BRYAN
feat: added routing and service logic for summarizing	9e8d246ec7a5645007b5b2543df160982b867f14	BRYAN
refactor: modified logic for accepting just one answer	d6422708ce5738fc94c1370ee826e0eac2fe27fd	BRYAN
refactor: tuned up the context for summarizing	76100f3f44357bf490f02044680c761926b1e069	BRYAN
test: Add comprehensive unit tests for gemini_services to improve coverage	ca8bb8514edf3c9408df6efc7aa6d3d9d477cccf	BRYAN
chore: backend auth-middleware redirection for unauthorised access	d4b73f75c839c05c10db507b5c98636b43ce5bfe	DEVANG
test:first test written for user registration failed (class does not exist).	6274f96479dcc5110e4952e38b517fff8ec40fa0	SAAYONEE
feat: added minimal implementation to pass test	685e48c14b28f7a2abf8408bd7af4ee2469a84e8	SAAYONEE
test: added second test that fails because there hasn't been proper implementation of the code	6b678ef6d31ea656884460b92113f96276d64d6f	SAAYONEE
feat: added proper implementation to pass the tests	694ecc2dd1ac2ca90487cb1de915eb8edeb60bf3	SAAYONEE
test: (create_user) added first test which fails (no attribute)	31862f016ef3bd499f0dc203cc6dcbc2d93ac8d3	SAAYONEE
feat: [create_user] added minimal code to pass test	c180efa5f1814d2ef9cb03bf31d9e5cb0cc94090	SAAYONEE
test: [create_user] failed second test which tests opposite functionality	e38d6fbd1cae1d30773b267a413313b01270d8b5	SAAYONEE
feat: [create_user] added code to pass second test	ec18e6e23c948376299de480ce3c5930c7d9813c	SAAYONEE
test: [validate_email] failed first test to validate the email id (no attribute)	badab66b757974d61bd8e1ce91be229c4496adc4	SAAYONEE
feat: [validate_email] added minimal code to pass test	b43148a00c3c094362c73351593ef2a870f4abed	SAAYONEE

test: [validate_email] failed second test to check opposite functionality	18b0dd851973f48f6922fb006a424830f8242ba8	SAAYONEE
feat: [validate_email] added code to pass second test	8327ff13c6438f4ea571256c64ea077bcfc1dfab	SAAYONEE
refactor: [create_user] added email validation before creating user	01f805536cd4560e2c6939fb139e0de9f4670ad6	SAAYONEE
feat: [OTP] added functionality for generating and sending OTP to user for email verification	0db8e54dac6735d590c4ddd595af5c9ad592575d	SAAYONEE
test: [verify_and_create_user] failed first test because no attribute	daf29d93203970b610fe5293096475279b93a89b	SAAYONEE
feat: [verify_and_create_user] minimal implementation	553432e5ae33c3de9113714cab3441ae0cf108ff	SAAYONEE
test: [verify_and_create_user] failed second test with opposite functionality i.e. incorrect OTP	adc330416445f75b6e802aa2f32519ff5374846b	SAAYONEE
feat: [verify_and_create_user] added implementation for passing test cases	39c08e1d98da0cad153f285ad4246f7468d7b82d	SAAYONEE
chore: [verify_and_create_user] fixed test case name	b48c564c72e5fdefaf717d40a12d6dfa19ce312c	SAAYONEE
refactor: [verify_and_create_user] added user details	16ddf44c3e13a1a53768b7813247efd2bd7c294e	SAAYONEE
refactor: [create_user] added pending_email and pending_password to maintain integrity	093d40008c42ba63582dcac09f86dfc64dfc3b31	SAAYONEE
refactor: minor fixes	b7cb4ca0f7dbfd1950c415fe590894248642996b	SAAYONEE
feat: [verify_and_create_user] [frontend: Login + Registratioin page] [routes:...	22e1653ba22992f5dea22ed6fe2c048abce635e9	SAAYONEE
feat: worked on routes (login_routes)	195aac7c63e45dab4de57f26e77f964a46537e4c	SAAYONEE
test: [verify_credentials] failed first test due to no attribute	418648ec7ceeeebd1ad0bdcb60340ac6275a392e	SAAYONEE
feat: [verify_credentials] minimal implementation to pass test	6755cf4b7bb9494d743243dd6ba5b00bc81e91b0	SAAYONEE

feat: [login] frontend and backend implementation of login. Fixed issues and did implementation	6755cf4b7bb9494d743243dd6ba5b00bc81e91b0	SAAYONEE
test: [verify_credentials] second test failed due to minimal implementation which returns true	35a01d7d8694096aec737195802f6272021d361c	SAAYONEE
feat: [verify_credentials] fix implementation to pass second test && fixed filter_by mistake	1e4fdbac05eec93b0b98f915d5200bcdec7c6f49	SAAYONEE
test: [verify_credentials] test to check if user exists or not	f165e0a564b40bd23fb46d428dae572dc4fdcabb	SAAYONEE
feat: [get_user_info] fetch user details, if exists, from the db	ffc69d15bcab949d7ef1083ed191463898016998	SAAYONEE
test: testing if the answer is valid. test failed due to no code	3caaa20192029b9da52161bf9b49466aa06150a2	SAAYONEE
feat: minimal code to pass test	8e8161dae5984e0df4a7d0636142ae498cda23cb	SAAYONEE
feat: actual implementation to test if body is greater than 20 characters	a9268a4f3aec115bd77b5e7313217fdc2bb8b125	SAAYONEE
test: failing test for creating answer with valid data	0861bbc3d28b362595fbeb4675322ff9a4ddbe8b	SAAYONEE
test: minimal code	1e24b29e46fbf18c56477a5690bf5da470166e04	SAAYONEE
test: failing tests opposite to previous test	6e135f358da39e1bf9f5660f849eac994fa5eed4	SAAYONEE
feat: actual code to implement functionality and pass tests [create answer]	88c6a6b22fd68410c7065ae580fd6b3b45f9f6c5	SAAYONEE
test: retrieve answers by question id	42a158ad179dbe3af46007f72f98502c83975a8e	SAAYONEE
feat: retrieve answers by question id	0ddb69b81435a611fa62b33fbd53e27c942d0049	SAAYONEE
test: retrieve answers by user id	cf71c257472d9f0083994dbb5c78c5139e404484	SAAYONEE
feat: retrieve answers by user id	9fcf9c90a46873775a9623f07aa708c46be716c5	SAAYONEE
test: retrieve answers when none exist	14980589021981344a654d7317fb14da074a54fa	SAAYONEE
test: count answers for user failing test	6406669343879bf4ee2219eb446bfee694fcf5f	SAAYONEE
feat: count answers for user passed test	e4356e45185dedd46daf00985ed0867a937c1b8d	SAAYONEE

test: count answers for questions failed test	f084f943718a15d71763253ae6817883e423c1c6	SAAYONEE
feat: count answers for questions passed test	ba75a6c6e663b8d9138180e7f7f216372c5a6995	SAAYONEE
test: integration test for get answers endpoint. Failed	ce871728e36103f2882126ee7ce898a12e88596a	SAAYONEE
feat: integration test for get answers endpoint. Passed Test.	7f7f9778dc198f3409949c4f7bbec1daf49416a9	SAAYONEE
test: integration test to get the endpoint of answer count and if proper...	f4cb70941ec8325b63cffe6ca2302d04f5b0e588	SAAYONEE
feat: integration test to get the endpoint of answer count and if proper...	391a274ef5f5ec7bc6e187a4422081e9bb594b91	SAAYONEE
test: integration test to test that the post endpoint exists. Test failed	e7b717f2cc8ef7401f5d0362468a5f3555d56b51	SAAYONEE
feat: integration test to test that the post endpoint exists. Test passed	ea68666894695f6c372819de6d2d70661a6a38e4	SAAYONEE